ANNEXURE

LIST OF SKILL BASED COURSES

(ODD SEMESTER)

S. NO.	SEM	COURSE CODE	COURSE TITLE	L	T	P	C
SEMESTER V (R-2021)							
1.	**V**	SB3001	Project Based Experiential Learning Program	1	0	2	2
2.	**V**	SB3002	Employability Skills: Basics	1	0	2	2
SEMESTER VII (R-2017)							
3.	**VII**	SB8041	Concrete Building System Design	1	0	2	2
4.	**VII**	SB8042	Integrated Building Services	1	0	2	2
5.	**VII**	SB8043	Design of On Off-Grid Solar PV System	1	0	2	2
6.	**VII**	SB8044	Surface Modelling	1	0	2	2
7.	**VII**	SB8045	Information and Communication Technology (ICT) Engineer - 5G Networks	1	0	2	2
8.	**VII**	SB8046	System Architect – 5G Cloud RAN	1	0	2	2
9.	**VII**	SB8047	Linux Administration Fundamentals	1	0	2	2
10.	**VII**	SB8048	Introduction to ChatGPT and Building Applications using ChatGPT	1	0	2	2
11.	**VII**	SB8049	Full Stack Python Programming with Machine Learning	1	0	2	2
12.	**VII**	SB8050	DevOps	1	0	2	2
13.	**VII**	SB8051	Full Stack Development	1	0	2	2
14.	**VII**	SB8052	Artificial Intelligence and Applications	1	0	2	2
15.	**VII**	SB8053	Cyber Security and Applications	1	0	2	2
16.	**VII**	SB8054	Machine Learning	1	0	2	2
17.	**VII**	SB8055	Block Chain Development	1	0	2	2
18.	**VII**	SB8056	Digital Marketing	1	0	2	2
19.	**VII**	SB8057	Data Analytics with Tableau & Python	1	0	2	2
20.	**VII**	SB8058	Manufacturing of Mechanical Components Using CNC Turning / Milling	1	0	2	2
21.	**VII**	SB8059	Programming Logic Circuit Using SIMATIC	1	0	2	2
22.	**VII**	SB8060	Industry 4.0 and its applications in Mechanical Engineering	1	0	2	2
23.	**VII**	SB8061	3D Printing using NX	1	0	2	2
24.	**VII**	SB8062	Electric Vehicle Technology and Manufacturing	1	0	2	2
25.	**VII**	SB8063	Applied ML for Health Care Industry	1	0	2	2
26.	**VII**	SB8064	Industry 4.0- Digital Twin for Process Industry	1	0	2	2
27.	**VII**	SB8065	Drone Basics, Design, Assembly, Test	1	0	2	2
28.	**VII**	SB8066	Architecture Visualization using AR/VR	1	0	2	2

SEMESTER V

SB3001	**PROJECT BASED EXPERIENTIAL LEARNING PROGRAM**	**L T P C**
		1 0 2 2

Program Details

This program is a mandatory, 4 months immersive project-based learning program delivered as a course in the curriculum. 5th Semester students of engineering colleges are the beneficiaries of the program. Project Based Learning is introduced to the students to give them hands-on experience on open-source digital technologies while they learn an end-to-end journey to solve a problem. By the end of this course, the student understands the approach to solve a problem in collaborative learning environment while being guided by mentors from Industry and College

Highlights of the Program:

- Total 80 Hrs. of project-based learning engagement
- 40 Hrs. of virtual live instructor-led, hands-on training on leading technologies such as
- Artificial Intelligence,
- Data Science
- Data Analytics & Visualization
- Internet of Things
- Cloud Application Developer
- 40 Hrs. to develop the solutions for the real-world use cases in various business domains like healthcare, finance, retail, fashion, agriculture, transportation, manufacturing, etc.
- 50 use cases identified to build innovative solutions by implementing leading technologies.
- Team based development activity build their professional skills
- Use case specific mentoring support to handhold the students during the development process.
- Training the students on design thinking, agile methodologies, and project development process.
- 14 hrs. of knowledge sessions for the faculty on design thinking, agile methodologies, and project development process.
- Hands-on with open-source technologies and digital tools
- Master classes for the students and faculty on best industry practices
- Evaluation of project submissions by evaluators
- Dashboard Access to the various stakeholders to track the progress
- Certificate of completion along with assessment report for students

Outcomes of the Program

On completion of the course, the students will be able to

- Understand the leading technologies and apply them for solving real-world problems
- Understand the concepts of design thinking and agile methodologies
- Develop professional skills like teamwork, time management, communication and project management skills
- Understand various digital tools & best practices followed in the industry
- Develop the products from scratch i.e. idea to working prototype
- Industry course completion certificate from IBM

1. Program Milestones

Following are the major milestones of the project. Refer to annexure-1 for the detailed project plan.

- Set-up the platform to onboard various stakeholders

- Organize the program orientation session for principals & faculty
- Onboard the principals, college SPOCs, faculty mentors and faculty evaluators
- Organize the program orientation session for students
- Onboard the students on to the platform
- Use case orientation sessions for the students
- Team formation & project enrollment
- Team approval & faculty mentor, evaluator assignment
- Conduct Agile & Design Thinking sessions for students which can be attended by faculties too
- Training batch selection & team assignment
- Technical training sessions for students
- Knowledge sessions for faculty
- Design thinking & Agile training sessions for students
- Project development by student teams
- Project mentoring sessions
- Project submission by the student teams
- Project approval by the mentors
- Project evaluation by the evaluators & score submission
- Leaderboard generation
- Issuing certificate of completion
- Program success meeting & closure.

Curriculum for Technical Trainings

Technology Track	**Topics covered**	**Training Hours**
Artificial Intelligence	● Introduction to Artificial Intelligence ● Python for Artificial Intelligence ● Data Wrangling Techniques ● Introduction to Neural Networks ● Tensorflow & Keras ● Convolutional Neural Networks ● Natural Language Processing ● Build and Deploy AI Applications	40 Hrs.
Data Science	● Introduction to Data Science ● Python for Data science ● Data Wrangling Techniques ● Data visualization ● Supervised Learning - Regression ● Supervised Learning - Classification ● Unsupervised Learning ● IBM Cloud & Watson Services Build and Deploy Machine Learning Applications	40 Hrs.

Data Analytics & Visualizations	● Introduction to Data Analytics ● Python for Data Analysis ● Extract data from database, txt files, webscrapping ● Data Visualization using Matplotlib, seaborn ● Data Visualization using Plotly ● Data Visualization using Pygal, Bokeh etc. ● Build Visualization Dashboards & Stories	40 Hrs.
Internet of Things (IoT)	● Introduction to Internet of Things (IoT) ● Building Blocks of IoT ● Design an IoT Device ● Getting Started with Arduino UNO & Tinkercad Platform ● Getting started with ESP32 ● Integrate TMP36 Sensor with Arduino UNO ● Integrate Ultrasonic Sensor with Arduino UNO ● Integrate Servo Motor with Arduino UNO ● IoT Communication Technologies ● IoT protocols ● HTTP & MQTT Practical's ● Introduction to Python ● Python - Collection Data Types, Functions ● Modules Files I/O ● Getting started with Raspberry PI ● Raspberry Pi as a Gateway ● Introduction to Cloud ● IBM Cloud & IoT Platform Architecture ● Create and Configure IBM IoT Platform ● IBM Watson IoT Platform Hands-on with online simulator ● Getting Started Node-RED Service ● Web Application Development using Node-RED Service ● Mobile Application Development using MIT App Inventor ● IBM Cloudant DB & its Practical's ● Introduction to Computer Vision with Python ● Introduction to IBM Watson AI & Building Intelligent Devices	40 Hrs.
Cloud Application Developer	● Modern Application Development ● Hands-on with web application development ● Python – Flask Framework ● IBM DB2 ● IBM Object Storage ● IBM Watson Assistant ● Containers & Kubernetes ● IBM Container Registry ● IBM Cloud Foundry	40 Hrs.

Sample Use Cases for Project Development

Technology Track	**Use cases**
Artificial Intelligence	● Fake News Detection Using NLP ● AI-based localization and classification of skin diseases
Data Science	● Assessing the Safety of Municipal Drinking Water ● Comparative Analysis of GDP Growth across Developed and Developing Countries
Data Analytics & Visualizations	● Exploratory Data Analysis of Rainfall in Chennai ● Corporate Employee Attrition Analytics & Visualization
Internet of Things (IoT)	● IoT Based Smart Crop Protection System for Agriculture ● Personal Assistance for Seniors who are Self-Reliant
Cloud Application Developer	● Skill/Job Recommender Application ● Nutrition Assistant Application

TOTAL: 45 PERIODS

SB3002 EMPLOYABILITY SKILLS: BASICS

L	T	P	C
1	0	2	2

Execution Plan

Mode of Delivery: Crash course – Offline Delivery (Classroom)
Timeline: 2 weeks (Monday to Friday) – 4.5 hours/day – 10 days – 45 hours
Evaluation: Daily Practice test - 2 weekly tests – One Final Exam
Zones: 38 districts categorized into 5 groups

CORE EMPLOYABILITY SKILL CATEGORIES (TOTAL – 45 HOURS)

- Aptitude Skills – 20 Hours
- Communication Skills – 10 hours
- Technical & Core Domain Skills on Employability- 8 hours
- Professional Skills – 7 hours

Aptitude Skills-Quantitative & Logical 20

Speed Maths – HCF & LCM – Ratio & proportion – Percentage – Profit & Loss – Simple Interest & Compound Interest – Data Interpretation (Charts, Tables, Pie Charts, Lines, Histogram)– Boats & Streams – Surds & Indices – Time, Speed, Distance and Work - Number Series – Alphabetical Series – Inequalities – Seating Arrangement – Logical order – Logical deductions – Permutations – Combinations – Probability – Arithmetic Progression – Geometry – Clocks – Statistics (Averages)

Communication Skills 10

Basic Grammar – Subject Verb agreement – Tenses – Prepositions – Sentence Structure – Listening – General Gist – Specific Information Reading – Subskills (Skimming, Scanning, Intensive Reading) – Reading Strategies – Speaking – Asking & Answering questions – Self Introduction – Short Presentation – Writing – Email communication – Short Report – Resume Writing - Vocabulary – Word formation – Roots – Prefixes – Suffixes – Synonyms & Antonyms –

Domain specific vocabulary

Technical & Core Domain Skills on Employability: 8

Use of Modern Tools and Technology on domain-specific employability

Professional Skills: 7

Personal attributes (Reliability, Integrity, Dependability – Flexibility - Initiative and Enterprise - Consistency in performance – Time management) - Teamwork, Collaboration, and Co-Operation - Leadership/ Management Skills - Work Ethics - Problem-Solving and Decision-Making Skills - Negotiation Skills and Networking – Interview Preparation & Presentation – Dress Code

TOTAL: 45 PERIODS

SEMESTER VII

SB8041 CONCRETE BUILDING SYSTEM DESIGN

L	T	P	C
1	0	2	2

Course objectives

- ➢ Comprehend site specific aspects of the Multistorey building like geotechnical investigations, project requirements and Compute loads of various elements & services
- ➢ Understand proper selection and design of vertical & lateral load resisting systems (moment frame or shear wall system) for the various loads acting on the building
- ➢ Describe the steps in carrying out a full-fledged analysis for a multi-storeyed reinforced concrete building using analysis software
- ➢ Explain the steps for designing various structural members of a reinforced concrete building,
  including beams, columns, slabs, flat slabs and shear walls in line with practical detailing methods used and calculate the BOQ for the whole structure.

Course Content

Module 1: Introduction and Codes- Design Basis Parameters and Report 9

Indian & International Codes for Reinforced concrete Design, Design loads and detailing of reinforcement, Handbooks for reinforced concrete design, National Building Code 2016, Practical building examples, drawing sizes and scales, Reading Drawings – Architectural & Structural.

Introduction to DBR Parameters - Geometric Parameters, Occupancy Categories, Site location and associated parameters, Design life of structures, Material Specifications - Grade of concrete for vertical and floor elements – Grade of reinforcing steel, Exposure and cover requirements, Fire rating requirements, Load Combinations, Serviceability Requirements, Analysis tools, Design Basis Report, Concept explanation with example buildings

Module 2: Loads & Setting the Structural Scheme 9

Introduction, dead loads, superimposed dead loads, Live loads, Wind loads, Wind pressure coefficients, Determining global wind forces and wind velocity, storey forces and base shears. Earthquake loads, response spectrum to earthquake excitation, seismic design parameters - horizontal acceleration coefficient, Time period, Evaluation and application of seismic base shear, equivalent static method.
Loads from MEP Services and architectural considerations like façade loads

Scheme Design, Concrete floor systems, Sizing and design of various slab systems, Dimensioning & designing of drop panels, Beams, Reinforced Concrete Columns - Location and Shape, Design Axial Load, Sizing, Lateral Load Systems, IS 1893- Requirements, Shear Walls – Location and thickness. Estimating relative stiffness of core walls

Module 3: Structural Models 9

Introduction to Analysis & Modelling, Modelling of Cantilever, Portal Frame, three bay Portal Frame, 3D structural models - Geometry, gravity loads, defining earthquake loads, defining wind loads, Modelling Shear walls, Practical Structural Model of building, Structural models of Floor System, Direct design method for Flat Slabs, Analysis of two-way slabs using moment coefficient method, Application of moment coefficient method, Estimation of deflections

ETABS software demonstration for correct modelling and design of Vertical and Lateral loading systems like Shear Walls

Module 4: Design of Structural Elements 9

Design of structural elements - Design of Beams- flexural reinforcement, shear reinforcement- design of edge beam, Practical examples, Design of flat slabs- Flexural Reinforcement, shear reinforcement- Practical Examples-Design of mesh reinforcement, additional bottom reinforcement, additional top reinforcement, Design of 2-way continuous slabs.
Design of Reinforcements in Columns - Post processing of column forces from analysis, Design and arrangement of vertical reinforcement, Design of horizontal reinforcement, Design of stirrups, Cardinal rules in scheme design of buildings, Coordination with other Engineering disciplines
Design of shear walls – General considerations, Seismic response of RC structures, Vertical and Horizontal Reinforcement, Calculation of design forces, moment capacity of vertical distributed reinforcement, Design of boundary elements and boundary zone. Sizing of elements based on Constructability aspects like formwork, concrete placement and compaction, rebar arrangement to satisfy economy and optimum utilization.

Module 5: Detailing of Structural Elements- Bill of quantities and conclusion 9

Development length of rebars, detailing of various structural elements - flat slabs, two-way continuous slabs, beams, columns and shear wall, detailing and documentation of practical example building.

Bill of quantities - Concrete and steel indices for RC buildings, Reinforcement consumption in RC members, BoQ of practical example building.

TOTAL: 45 PERIODS

Course Outcomes

On completion of the course, the students will be able to:

- ➢ Comprehend site specific aspects of the structure like geotechnical investigations, project requirements
- ➢ Develop design basis requirements like building functionalities, durability and materials

- ➢ Compute loads (Dead, Superimposed, Live, Wind, Seismic) of various elements & services
- ➢ Selection and design of vertical & lateral load resisting systems for the various loads acting on the building
- ➢ Carry out a full-fledged analysis & design for a multi-storeyed reinforced concrete building, using a modelling software

For further reading

REFERENCES:

IS 456:2000, IS 1893, IS 875 (Part 1-5), SP6, NBC (Part 1 & 2) : 2016

Illustrated Design of Reinforced Concrete Buildings - by V. L. Shah, S. R. Karve

Online references:

E-learning content of L&T EduTech

Software requirements:

1. ETABS / STAAD.Pro *

** Software availability to be ensured for carrying out the project work*

Hardware Requirements

1. PC – Minimum requirements

a. 8 GB RAM

b. 2GB - Dedicated GPU

c. 500 GB HDD

Industry Scope:

Design engineer, Site planning engineer, Construction engineer, R&D Engineer, Design Consultant

Project:

S.NO	PROJECT TITLE
1	Design of a Residential Building at Chennai, Tamil Nadu
2	Design of a Hostel Building at Kolkata, West Bengal
3	Design of a College Building at Bengaluru, Karnataka
4	Design of a Government Office at Hyderabad, Telangana
5	Design of a Research establishment at Pune, Maharashtra

S.NO	PROJECT TITLE
6	Design of a Start-up office at Chennai, Tamil Nadu
7	Design of a Residential apartment building at Trivandrum, Kerala
8	Design of a Resort hotel at Guwahati, Assam
9	Design of a Start-up office at Cochin, Kerala
10	Design of a Court building at Kolkata, West Bengal
11	Design of a Residential Building at Mumbai
12	Design of an office building at Bangalore, Karnataka
13	Design of a College Building at Maharashtra
14	Design of a Hostel Building at Bhubaneswar, Orissa
15	Design of an office building at Ahmedabad, Gujarat
16	Design of a Residential Building at Chandigarh, Punjab
17	Design of a Resort hotel at Panaji, Goa
18	Design of a College Building at Lucknow, Uttar Pradesh
19	Design of a Hostel Building at Amaravati, Andhra Pradesh
20	Design of a Government office building at Imphal, Manipur

Task for Projects:

Task 1:

Study of architectural drawings and site-specific aspects to prepare Design Basis Report

Task 2:

Calculation of loads acting on the building

Task 3:

Planning of structural systems and preliminary design of elements

Task 4:

Software aided Modelling of the Reinforced concrete building, analysis and reading results

Task 5:

Design of the Reinforced concrete elements including ductile detailing

Sample Project for Illustration:

An office space is commissioned by a reputed client for their infotech park. The quotation is floated and your structural consultancy firm is also bidding for it.

The infotech park is situated at Noida, Delhi. The client is requesting a G+7 building with total floor area of 12288m^2, with each floor area of 64m x 24m, with equal grid spacing of 8m in both directions. Total height of the building as 28m, with 4m floor to floor height, and a required 3m clear height for serving all intended functional requirements.

Task for Project:

Task 1:

Study of architectural drawings and site-specific aspects to prepare Design Basis Report

Understand the functional requirement of the building by studying the architectural drawings and the design aspects of the building based on the location of the building.

Task 2:

Calculation of loads acting on the building

Calculate the various vertical and lateral loads acting on the building – Dead, Superimposed Dead, Live loads, Wind and Seismic loads, considering the site location, the functional aspects of the building like the MEP services required at different locations of the building. Prepare a Structural Design Basis Report (DBR) based on compiled data.

Task 3:

Planning of structural systems and preliminary design of elements

Prepare the scheme setting of the building – Selection of floor system, Moment frame, Shear wall or Flat slab system. Calculate the dimensions of each components as per the loads calculated in previous task

Task 4:

Software aided Modelling of the Reinforced concrete building, analysis and reading results

Create a structural analysis model of the building, considering, the geometric requirements, the functions of the building, the specifications and properties of the elements, the support conditions between the members and the base connection, the loads acting on different elements along different directions, and the design parameters considered in the preliminary sizing of the elements.

Design of the Reinforced concrete elements including ductile detailing

Perform the design of the building using a excel sheets by designing the elements manually. Prepare reinforcement detailing drawings for each element (one sample of each component)

Prepare a BOQ statement for summarizing the quantity of the materials used in the project.

SB8042 INTEGRATED BUILDING SERVICES L T P C

1 0 2 2

This course is recommended for Civil, Architecture, Mechanical & Electrical Disciplines.

Course Objectives:

The learner will be able to:

- Understand Electrical System for a building infrastructure.
- Learn ELV System and its interface with other allied services.
- Design and implementation of HVAC System.
- Learn and implement Fire Protection and Fire Alarm System
- Understand and implement importance of Public Health Engineering.

Course Content:

Module – I: Advanced Electrical System Design for Buildings 9

Basics of Electrical System, Electrical terminologies, Major Electrical equipment, Building power distribution and its schemes, Fundamentals of Power& distribution transformers, HT, LT, DG Sets, Cables & Wires, UPS and its importance, Introduction of HT, LT switchgears systems, Importance of Lighting design & different Light fixtures used in buildings – Interior, external, street & offices, RMU, HT consumer, Substation Building in Master plan - Space planning for RMU,HT, DG set, HSD yard, Space provision for Electrical Equipment including Substation, Various equipment clearance requirements, HVAC, PHE, FPS service-electrical load input for designing electrical power distribution, Pedestals & ceiling support requirement for all Electrical equipment.

Module – II: Extra Low Voltage System for Infrastructure 9

Introduction & Brief of ELV Systems, Concept of Building Management System (BMS) & Fire Alarm System, Interface with Architecture/ Structure, Access control, CCTV & Public address system - Brief and purpose, BMS - Brief and purpose, BMS interfaces with Electrical, HVAC, Fire & Life Safety and PHE, BMS interfaces with airport systems.

Module – III: Heating, Ventilation & Air conditioning systems 9

Basics of HVAC - Psychrometry and its importance - Major Components of Air conditioning System - Fundamental concepts of Heat transfer, Air-conditioning system, Ventilation system, Pressurization Systems and their importance to Life safety, Chilled water system, Cooling towers and major HVAC equipment, Pumping system in HVAC, Importance of Thermal and Acoustic Insulation, Introduction and basics of Variable Refrigerant Flow (VRF) systems, Radiant cooling, Underfloor distribution, Chilled beams – Space planning - Importance of Static weight /

Operating weights of mechanical equipment - Importance of Floor slab and Terrace roof slab openings / cut-outs

Module – IV: Fire Protection and Life Safety System **9**

Basics of Fire Protection System - Active Fire protection system - Passive Fire protection system - Basics of Smoke Control and Fire Stop Systems - Codes & Standards and Statutory Compliance - Fire and its Classes - Hazard Classification based on building occupancy - Means of Egress and its components - Importance of Life Safety - Refuge Area, Fire Tower and Fire Lift - Occupant Load and Capacity factors - Fire Stopping Materials - Compartmentation in a building - Smoke control & management in Fire Zoning - Components of Fire Compartments.

Module – V Public Health Engineering **9**

Scope of works in Public Health Engineering - Sanitary fixtures and types - Water supply and treatment - Rain water drainage system - Landscape irrigation features – Water demand calculation based on building occupancy – Piping for different plumbing systems in buildings – Pump selection – Plant room sizing - Sewage treatment process - External water supply, storm drainage & sewerage system - Solid waste management - Interfacing PHE system with Architect and Structural engineers.

TOTAL: 45 PERIODS

COURSE OUTCOMES:

On completion of the course, the students will be:

- Design and Selection of Lighting System.
- Space Planning for MEP Electrical & ELV System.
- Design and implementation of HVAC System.
- Calculate pump and sump capacity for Fire Protection System.
- Preparation of Plumbing Schematic Layout.

For Further Reading:

- Handbook on Functional Requirement of Buildings – Published by BIS
- National Building Code of India – 2016

Online References:

1. LMS content of L&T EduTech

Software Requirements: NA

Hardware Requirements: NA

Industry Scope:

The course provides practical knowledge on Building Service Engineering which includes acquiring basic understanding and requirements of electrical, low voltage systems, HVAC system, Fire & Life safety system, Public Health Engineering etc., that will enable to design a holistic, functional entity.

PROJECTS – 20 NUMBERS

S. NO.	PROJECT TITLE

1.	Design of MEP services for IT Park
2.	Design of MEP services for Institution Building
3.	Design of MEP services for Office Buildings
4.	Design of MEP services for Data Centre
5.	Design of MEP services for Mall
6.	Design of MEP services for Multiplex
7.	Design of MEP services for Commercial Complex
8.	Design of MEP services for Clinic
9.	Design of MEP services for Residential Building
10.	Design of MEP services for Apartments
11.	Design of MEP services for Administrative Building
12.	Design of MEP services for Design Centre
13.	Design of MEP services for Boys Hostel
14.	Design of MEP services for Recreation Centre
15.	Design of MEP services for Club House
16.	Design of MEP services for Canteen Building
17.	Design of MEP services for Multi-level Car Parking

S. NO.	PROJECT TITLE
18.	Design of MEP services for Central Library
19.	Design of MEP services for Laboratory
20.	Design of MEP services for Girls Hostel

TASK FOR PROJECTS:

Task 1: Design of Electrical System

Task 2: Design of Extra Low Voltage System

Task 3: Design of HVAC Systems

Task 4: Design of Fire Protection and Fire Alarm System

Task 5: Design of Public Health Engineering

PROJECT - 1 DESCRIPTION:

The following are the building description:

- Height of the Building : Basement + Ground to 13 Floors + Terrace.
- Type of Building : IT Park
- Floor Area : 60 Mtrs (6,50,000 Mtr Sq)
- Toilet to PHE shaft distance : 5-6 meters
- Floor to floor height : 4 meters
- Incoming Water quality : TDS = 200 ppm, Total Hardness = 350 ppm. Iron (as FE) = 0.2 ppm.
- Population : 3800 per floor (1st to 13th, B+G is parking).

TASKS FOR PROJECT:

Task-1: Electrical:

- Selection & design of lighting for office building.
- Mention the Electrical equipment used in the office building.
- Space allocation for Electrical Equipment in the Layout with the statutory clearance.

Task-2: Extra Low Voltage System

- Identify the systems of a Building Management system with its equipment placement.

Task-3: HVAC

- Identifying Low side and High side HVAC equipment.
- Space Provision for HVAC equipment.

Task-4: Fire Protection and Fire Alarm System:

- Identifying different Fire Protection and Fire Alarm Systems for Buildings.
- Understanding Compartmentation in Buildings
- Calculate fire pump and sump capacity.

Task-5: Public Health Engineering:

- Calculate the total sanitary fixtures per floor required considering male female ratio as 70:30.
- Draw a schematic plumbing layout in the buildings.
- Choose an appropriate water and sewage treatment system.

SB8043	DESIGN OF ON / OFF-GRID SOLAR PV SYSTEM	L T P C
		1 0 2 2

COURSE OBJECTIVE:

The main aim of this course is to:

- Understand the details of Solar PV System
- Illustrate the design aspects involved in the Solar Photovoltaic system for On / Off-Grid using manual calculation as well as using software tools.
- Provide hands-on experience in Design of Solar PV System using SAM and PVsyst Software

COURSE CONTENT:

Module 1: Solar Power Generation **9**

Photo Voltaic (PV) Cells – Theory and Principles, Types and Modules, Electrical Performance Characteristics, Photo Voltaic Panel Tilt angles and Optimizing Energy Capture, Photo Voltaic System – Power System Configurations, Charge Controller and Inverters, Cost and Economics, Statistics and Future Outlook.

Module 2: Solar Power Sizing Calculations including On / Off Grid **9**

Solar Panel Sizing Calculation, Sizing of Converter for Solar Panel, PV System Standards and Software, Datasheets of Solar Panel and Inverters, Advantages of Solar Power, Solar Power Challenges and Mitigation

Module 3: Introduction/Familiarization to PVsyst and SAM Software **9**

Demonstration of Features System Advisor Model (SAM) software and PYsyst Software in design of Solar Power Plants; How to create a project, Input parameters on how to import files, metrological data of selected location, Solar Panel database; How to extract results, detailed report, detailed losses.

Module 4: Design of On-Grid Solar Power Plants using Simulation Software **9**

Design and Simulation of Grid Connected PV Solar System and Design Analysis of Grid Connected PV Solar System with Shading effects using PVSYST Software; Simulation of Grid Connected PV Solar System using SAM Software

Module 5: Design of Off-Grid Solar Power Plants using Simulation Software **9**

Design and Simulation of Off grid Connected/Standalone PV Solar System and Design Analysis of PV Solar System with Shading effects using PVSYST Software; Simulation of Off- Grid Connected PV Solar System using SAM Software

TOTAL : 45 PERIODS

COURSE OUTCOMES:

At the end of this course, learners will be able to:

- Design and model the complete photovoltaic system for **off-grid and on-grid system** including the selection of PV panels, inverter sizing, and battery sizing for a S**olar PV System Project** by using the manual method as well as using software tool.

ONLINE REFERENCES:

1. L&T EduTech Course on “Renewable Energy and Power Evacuation”

INDUSTRY SCOPE:

A student going through this course will have employment visibility and opportunity in Renewable Energy Sector as a

1. Design engineer
2. Construction engineer
3. Planning and Estimation engineer.

HARDWARE REQUIREMENT:

Computer/Laptop with minimum i3 Processor/8GB RAM and good broadband connection.

PVSyst / SAM software pre-installed.

INDUSTRY USE CASES

1. Preliminary Evaluation of a Grid-Connected and Standalone Solar PV System

Description:

Preliminary design analysis of Grid connected and Standalone Solar PV system at Mumbai(Maharashtra) location with the following details. Design shall be done by performing manual calculations as well as software tools.

Consider the following data for the preliminary design:

i) 1.5MWp Grid-Connected Solar PV system

ii) A standalone Solar PV system with a Lead Acid Battery with 12V, 15% LOL and 2 days of autonomy

Device	**Number**	**Wattage(W)**	**Hours/day**
Lamps	6	20	10
Fans	2	75	10
Television	1	55	4
Refrigerator	1	250	24
Air Conditioner	1	1000	8
Heater	1	1000	1
Water Pump	1	750	2

Any missing data can be assumed with relevance to the problem and the same to be brought out in the report.

Software Tools: PVSyst

Tasks:

1	Evaluate the annual yield, area for the meteo location
2	Determine the total energy required from the load data. Identify the peak sun hours of the location. (Hint: Use any online calculators like www.footprinthero.com) Total size of battery storage for a number of days of autonomy
3	Evaluate the battery capacity for the required autonomy, LOL, and battery voltage
4	Prepare the layout for both Grid-connected and Standalone system including schematic diagram
5	Prepare the detailed project report with the following

	a) Design methodology and assumptions
	b) Compare the results of manual calculation results with the software results
	c) Prepare the bill of quantity (BoQ) with brief description of key data

2. Design of a 100kWp Grid-connected Solar PV Plant

<u>Description</u>:

Design a 100kWp grid-connected Solar PV Plant and Illustrate for Kanyakumari (Tamil Nadu) location. Design shall be by performing manual as well as simulation tools. Consider the following data for the design:

- a standard solar PV Module datasheet of (Si-Mono): Pmax=660Wp, Vmp=37.8V, Imp=17.5A, Voc=45.7V, Isc=18.5A and efficiency=21.44%

- 80kWac centralized inverter.

Any missing data can be assumed with relevance to the problem and the same to be brought out in the report.

Software Tools: PVSyst

<u>Tasks:</u>

S. No	Description
1	Prepare a worksheet and find the required quantities for that meteo location
2	Estimate the amount of power yielded by constructing a scene without any shading
3	Estimate the amount of power yielded by constructing a scene with shading for a loss factor of 25%
4	Prepare the layout of the solar plant including schematic diagram
5	Prepare the detailed report with the following
	a) Design methodology and assumptions
	b) Compare the results of manual calculation results with the software results
	c) Prepare the bill of quantity (BoQ) with brief description of key data

3. Design of a 10kWp Standalone Solar PV System

<u>Description</u>:

Design a 10kWp Standalone PV Solar System in Dehradun (Uttarakhand) Location.
Design shall be performing manual calculation as well as simulated tools.

Consider the following data for the design:

- a typical Solar Panel of standard manufacturer(Si-Mono): Pmax=400Wp, Vmp=40.3V, Imp=9.9A, Voc=49.0, Isc=10.4A and efficiency=19.89%

- Power Demand is 10kW/day.

Any missing data can be assumed with relevance to the problem and the same to be brought out in the report.
Software tools: PVSyst, SAM

<u>Tasks:</u>

S.No	Description

1	Prepare a worksheet and find the required quantities for that meteo location
2	Using a suitable platform, find the Number of Solar PV modules required including the size of inverter
3	Prepare the total size of battery storage for a number of days of autonomy and battery voltage
4	Prepare the layout of the solar plant including schematic diagram
5	Prepare the detailed report with the following
	a) Design methodology and assumptions
	b) Compare the results of manual calculation results with the software results
	c) Prepare the bill of quantity (BoQ) with brief description of key data

4. Design of a 1MWp Grid-connected Solar PV Plant

Description:

Design a 1MWp grid-connected Solar PV Plant and Illustrate for Anantapur (Andhra Pradesh) location. Design shall be performing manual calculation as well as simulation tools.

Consider the following data for the design:

- a Typical solar PV Module of any standard manufacturer(Si-Mono): Pmax=600Wp Vmp=34.4V, Imp=17.4A, Voc=41.5V Isc=18.5A and efficiency=21.35%
- An 872kWac centralized inverter

Any missing data can be assumed with relevance to the problem and the same to be brought out in the report.

Software Tools: PVSyst, SAM

Tasks:

S. No	Description
1	Prepare a worksheet and find the required quantities for that meteo location
2	Using a suitable platform, find the Number of Solar PV modules
3	Also evaluate for the pnom ratio and annual yield
4	Prepare the layout of the solar plant including schematic diagram
5	Prepare the detailed report with the following
	a) Design methodology and assumptions
	b) Compare the results of manual calculation results with the software results
	c) Prepare the bill of quantity (BoQ) with brief description of key data

5. Design of a 5MWp Grid-connected Solar PV

Description:

Design a 5MWp grid-connected Solar PV Plant and Illustrate for Trichy (Tamil Nadu) location. Design shall be performing manual calculations as well as simulation tools. Consider the following data for the design:

- a Typical solar PV Module of any standard manufacturer(Si-Mono): Pmax= 500Wp Vmp=43.4V, Imp=11.5A, Voc=51.5V Isc=12.1A and efficiency=20.94%
- A 4.65MWac centralized inverter

Any missing data can be assumed with relevance to the problem and the same to be brought out in the report.

Software Tools: PVSyst, SAM

Tasks:

S. No	Description
1	Prepare a worksheet and find the required quantities for that meteo location
2	Using a suitable platform, find the Number of Solar PV modules
3	Evaluate for the pnom ratio and annual energy yielded
4	Prepare the layout of the solar plant including schematic diagram
5	Prepare the detailed report with the following
	a) Design methodology and assumptions
	b) Compare the results of manual calculation results with the software results
	c) Prepare the bill of quantity (BoQ) with brief description of key data

6. Preliminary Evaluation of a Grid-Connected and Standalone Solar PV System

Description:

Preliminary design analysis of Grid connected and Standalone Solar PV system at Mumbai (Maharashtra) location with the following details. Design shall be by performing manual calculations as well as software tools.

Consider the following data for the preliminary design:

iii) 1.5MWp Grid-Connected Solar PV system

iv) A standalone Solar PV system with a Lead Acid Battery with 12V, 15% LOL and 2 days of autonomy

Device	Number	Wattage(W)	Hours/day
Lamps	6	20	10
Fans	2	75	10
Television	1	55	4
Refrigerator	1	250	24
Air Conditioner	1	1000	8
Heater	1	1000	1
Water Pump	1	750	2

Any missing data can be assumed with relevance to the problem and the same to be brought out in the report.

Software Tools: PVSyst

Tasks:

1	Evaluate the annual yield, area for the meteo location
2	Determine the total energy required from the load data. Identify the peak sun hours of the location (Hint: Use any online calculators like www.footprinthero.com) Total size of battery storage for a number of days of autonomy
3	Evaluate the battery capacity for the required autonomy, LOL and battery voltage

4	Prepare the layout for both Grid-connected and Standalone system including schematic diagram
5	Prepare the detailed project report with the following
	d) Design methodology and assumptions
	e) Compare the results of manual calculation results with the software results
	f) Prepare the bill of quantity(BoQ) with brief description of key data

7. Design of a 100kWp Grid-connected Solar PV Plant

Description:

Design a 100kWp grid-connected Solar PV Plant and Illustrate for Kanyakumari (Tamil Nadu) location. Design shall be by performing manual as well as simulation tools.

Consider the following data for the design:

-a standard solar PV Module datasheet of (Si-Mono): Pmax=660Wp, Vmp=37.8V, Imp=17.5A, Voc=45.7V, Isc=18.5A and efficiency=21.44%
- 80kWac centralised inverter.

Any missing data can be assumed with relevance to the problem and the same to be brought out in the report.

Software Tools: PVSyst

Tasks:

S.No	Description
1	Prepare a worksheet and find the required quantities for that meteo location
2	Estimate the amount of power yielded by constructing a scene without any shading
3	Estimate the amount of power yielded by constructing a scene with shading for a loss factor of 25%
4	Prepare the layout of the solar plant including schematic diagram
5	Prepare the detailed report with the following
	d) Design methodology and assumptions
	e) Compare the results of manual calculation results with the software results
	f) Prepare the bill of quantity(BoQ) with brief description of key data

8. Design of a 10kWp Standalone Solar PV System

Description:

Design a 10kWp Standalone PV Solar System in Dehradun (Uttarakhand) Location. Design shall be performing manual calculation as well as simulated tools.

Consider the following data for the design:

- a typical Solar Panel of standard manufacturer(Si-Mono): Pmax=400Wp, Vmp=40.3V, Imp=9.9A, Voc=49.0, Isc=10.4A and efficiency=19.89%
- Power Demand is 10kW/day.

Any missing data can be assumed with relevance to the problem and the same to be brought out in the report.

Software tools: PVSyst, SAM

Tasks:

S.No	Description
1	Prepare a worksheet and find the required quantities for that meteo location
2	Using a suitable platform, find the Number of Solar PV modules required including the size of inverter
3	Prepare the total size of battery storage for a number of days of autonomy and battery voltage
4	Prepare the layout of the solar plant including schematic diagram
5	Prepare the detailed report with the following
	d) Design methodology and assumptions
	e) Compare the results of manual calculation results with the software results
	f) Prepare the bill of quantity(BoQ) with brief description of key data

9. Design of a 1MWp Grid-connected Solar PV Plant

Description:

Design a 1MWp grid-connected Solar PV Plant and Illustrate for Anantapur (Andhra Pradesh) location. Design shall be performing manual calculation as well as simulation tools.

Consider the following data for the design:

- a Typical solar PV Module of any standard manufacturer(Si-Mono): Pmax=600Wp Vmp=34.4V, Imp=17.4A, Voc=41.5V Isc=18.5A and efficiency=21.35%

- A 872kWac centralized inverter

Any missing data can be assumed with relevance to the problem and the same to be brought out in the report.

Software Tools: PVSyst, SAM

Tasks:

S.No	Description
1	Prepare a worksheet and find the required quantities for that meteo location
2	Using a suitable platform, find the Number of Solar PV modules
3	Also evaluate for the pnom ratio and annual yield
4	Prepare the layout of the solar plant including schematic diagram
5	Prepare the detailed report with the following
	d) Design methodology and assumptions
	e) Compare the results of manual calculation results with the software results
	f) Prepare the bill of quantity(BoQ) with brief description of key data

10. Design of a 5MWp Grid-connected Solar PV Plant

Description:

Design a 5MWp grid-connected Solar PV Plant and Illustrate for Trichy (Tamil Nadu) location. Design shall be performing manual calculations as well as simulation tools.

Consider the following data for the design:

- a Typical solar PV Module of any standard manufacturer(Si-Mono): Pmax= 500Wp Vmp=43.4V, Imp=11.5A, Voc=51.5V Isc=12.1A and efficiency=20.94%

- A 4.65MWac centralised inverter

Any missing data can be assumed with relevance to the problem and the same to be brought out in the report.

Software Tools: PVSyst, SAM

Tasks:

S.No	Description
1	Prepare a worksheet and find the required quantities for that meteo location
2	Using a suitable platform, find the Number of Solar PV modules
3	Evaluate for the pnom ratio and annual energy yielded
4	Prepare the layout of the solar plant including schematic diagram

5	Prepare the detailed report with the following
	d) Design methodology and assumptions
	e) Compare the results of manual calculation results with the software results
	f) Prepare the bill of quantity(BoQ) with brief description of key data

11. Design of a 2.25GWp Grid-connected Solar PV Plant

Description:

Design a 2.25GWp grid-connected Solar PV Plant and Illustrate for Jodhpur (Rajasthan) location. Design shall be by performing manual as well as simulation tools.

Consider the following data for the design:

- a standard solar PV Module datasheet of (Si-Mono): Pmax=480Wp, Vmp=42V, Imp=11.39A, Voc=50.8V, Isc=18.5A and efficiency=21.76%

- 250kWac String inverter.

Any missing data can be assumed with relevance to the problem and the same to be brought out in the report.

Software Tools: PVSyst

Tasks:

S.No	Description
1	Prepare a worksheet and find the required quantities for that meteo location
2	Using a suitable platform, find the Number of Solar PV modules
3	Also evaluate for the pnom ratio and annual yield
4	Prepare the layout of the solar plant including schematic diagram
5	Prepare the detailed report with the following
	i. Design methodology and assumptions
	ii. Compare the results of manual calculation results with the software results
	iii. Prepare a worksheet and find the required quantities for that meteo location

12. Design of a 1GWp Grid-connected Solar PV Plant

Description:

Design a 1GWp grid-connected Solar PV Plant and Illustrate for Kurnool(Andhra Pradesh) location. Design shall be by performing manual as well as simulation tools. Consider the following data for the design:

-a standard solar PV Module datasheet of (Si-Mono): Pmax=555Wp, Vmp=31.8V, Imp=17.45A, Voc=38.10V, Isc=18.56A and efficiency=22.89%

- 250kWac String inverter.

Any missing data can be assumed with relevance to the problem and the same to be brought out in the report.

Software Tools: PVSyst

Tasks:

S.No	Description
1	Prepare a worksheet and find the required quantities for that meteo location
2	Using a suitable platform, find the Number of Solar PV modules
3	Also evaluate for the pnom ratio and annual yield
4	Prepare the layout of the solar plant including schematic diagram
5	Prepare the detailed report with the following
	i. Design methodology and assumptions
	ii. Compare the results of manual calculation results with the software results
	iii. Prepare a worksheet and find the required quantities for that meteo location

Description:

13. Design of a 60MWp Grid-connected Solar PV Plant

Design a 60MWp grid-connected Solar PV Plant and Illustrate for Indore (Madhya Pradesh) location. Design shall be by performing manual as well as simulation tools. Consider the following data for the design:

- a standard solar PV Module datasheet of (Si-Mono): Pmax=390Wp, Vmp=40.5V, Imp=9.64, Voc=49.7V, Isc=10.08A and efficiency=22.12%

- 3437kWac string inverter.

Any missing data can be assumed with relevance to the problem and the same to be brought out in the report.

Software Tools: PVSyst

Tasks:

S.No	Description
1	Prepare a worksheet and find the required quantities for that meteo location
2	Using a suitable platform, find the Number of Solar PV modules
3	Also evaluate for the pnom ratio and annual yield
4	Prepare the layout of the solar plant including schematic diagram
5	Prepare the detailed report with the following
	i. Design methodology and assumptions
	ii. Compare the results of manual calculation results with the software results
	iii. Prepare a worksheet and find the required quantities for that meteo location

14. Design of a 25MWp Grid-connected Solar PV Plant

Description:

Design a 25MWp grid-connected Solar PV Plant and Illustrate for Surat (Gujarat) location. Design shall be by performing manual as well as simulation tools.

Consider the following data for the design:

- a standard solar PV Module datasheet of (Si-Mono): Pmax=500Wp, Vmp=42.80V, Imp=11.69A, Voc=51.7V, Isc=12.28A and efficiency=22.69%
- 1000kWac string inverter.

Any missing data can be assumed with relevance to the problem and the same to be brought out in the report.

Software Tools: PVSyst

Tasks:

S.No	Description
1	Prepare a worksheet and find the required quantities for that meteo location
2	Using a suitable platform, find the Number of Solar PV modules
3	Also evaluate for the pnom ratio and annual yield
4	Prepare the layout of the solar plant including schematic diagram
5	Prepare the detailed report with the following
	i. Design methodology and assumptions
	ii. Compare the results of manual calculation results with the software results
	iii. Prepare a worksheet and find the required quantities for that meteo location

15. Design of a 8MWp Grid-connected Solar PV Plant

Description:

Design a 8MWp grid-connected Solar PV Plant and Illustrate for Hubli (Karnataka) location. Design shall be by performing manual as well as simulation tools.
Consider the following data for the design:

- a standard solar PV Module datasheet of (Si-Mono): Pmax=485Wp, Vmp=42.2V, Imp=11.49A, Voc=51.10V, Isc=12.07A and efficiency=22.00%

- 80kWac centralised inverter.

Any missing data can be assumed with relevance to the problem and the same to be brought out in the report.

Software Tools: PVSyst

Tasks:

S.No	Description
1	Prepare a worksheet and find the required quantities for that meteo location
2	Estimate the amount of power yielded by constructing a scene without any shading
3	Estimate the amount of power yielded by constructing a scene with shading for a loss factor of 3%
4	Prepare the layout of the solar plant including schematic diagram
5	Prepare the detailed report with the following
	g) Design methodology and assumptions
	h) Compare the results of manual calculation results with the software results
	i) Prepare the bill of quantity(BoQ) with brief description of key data

16. Design of a 20MWp Grid-connected Solar PV Plant

Description:

Design a 20MWp grid-connected Solar PV Plant and Illustrate for Andaman Nicobar Islands (UT) location. Design shall be by performing manual as well as simulation tools.

Consider the following data for the design:

- a standard solar PV Module datasheet of (Si-Mono): Pmax=435Wp, Vmp=42.5V, Imp=10.24A, Voc=50.4V, Isc=10.67A and efficiency=24.10%

- 4000kWac centralised inverter.

Any missing data can be assumed with relevance to the problem and the same to be brought out in the report.

Software Tools: PVSyst

Tasks:

S.No	Description
1	Prepare a worksheet and find the required quantities for that meteo location
2	Using a suitable platform, find the Number of Solar PV modules
3	Also evaluate for the pnom ratio and annual yield
4	Prepare the layout of the solar plant including schematic diagram
5	i. Prepare the detailed report with the following
	ii. Design methodology and assumptions
	iii. Compare the results of manual calculation results with the software results

17. Design of a 35kWp Grid-connected Solar PV Plant

Description:

Design a 35kWp grid-connected Solar PV Plant and Illustrate for Shimla (Himachal Pradesh) location. Design shall be by performing manual as well as simulation tools.

Consider the following data for the design:

- a standard solar PV Module datasheet of (Si-Mono): Pmax=435Wp, Vmp=42.5V, Imp=10.24A, Voc=50.4V, Isc=10.67A and efficiency=24.10%

- 30kWac centralised inverter.

Any missing data can be assumed with relevance to the problem and the same to be brought out in the report.

Software Tools: PVSyst

Tasks:

S.No	Description
1	Prepare a worksheet and find the required quantities for that meteo location
2	Estimate the amount of power yielded by constructing a scene without any shading
3	Estimate the amount of power yielded by constructing a scene with shading for a loss factor of 25%
4	Prepare the layout of the solar plant including schematic diagram
5	Prepare the detailed report with the following
	j) Design methodology and assumptions
	k) Compare the results of manual calculation results with the software results
	l) Prepare the bill of quantity(BoQ) with brief description of key data

18. Design of a 20MWp Grid-connected Solar PV Plant

Description:

Design a 20MWp grid-connected Solar PV Plant and Illustrate for Durgapur (West Bengal) location. Design shall be performing manual calculation as well as simulation tools.

Consider the following data for the design:

- a standard solar PV Module datasheet of (Si-Mono): Pmax=485Wp, Vmp=42.2V, Imp=11.49A, Voc=51.10V, Isc=12.07A and efficiency=22.00%
- A 250kWac centralised inverter

Any missing data can be assumed with relevance to the problem and the same to be brought out in the report.

Software Tools: PVSyst, SAM

Tasks:

S.No	Description
1	Prepare a worksheet and find the required quantities for that meteo location
2	Using a suitable platform, find the Number of Solar PV modules
3	Also evaluate for the pnom ratio and annual yield
4	Prepare the layout of the solar plant including schematic diagram
5	Prepare the detailed report with the following
	g) Design methodology and assumptions
	h) Compare the results of manual calculation results with the software results
	i) Prepare the bill of quantity(BoQ) with brief description of key data

19. Design of a 30MWp Grid-connected Solar PV Plant

Description:

Design a 30MWp grid-connected Solar PV Plant and Illustrate for Rourkela (Odhisa) location. Design shall be performing manual calculation as well as simulation tools.

Consider the following data for the design:

- a standard solar PV Module datasheet of (Si-Mono): Pmax=390Wp, Vmp=40.5V, Imp=9.64, Voc=49.7V, Isc=10.08A and efficiency=22.12
- A 1200kWac string inverter

Any missing data can be assumed with relevance to the problem and the same to be brought out in the report.

Software Tools: PVSyst, SAM

Tasks:

S.No	Description
1	Prepare a worksheet and find the required quantities for that meteo location
2	Using a suitable platform, find the Number of Solar PV modules
3	Also evaluate for the pnom ratio and annual yield
4	Prepare the layout of the solar plant including schematic diagram
5	Prepare the detailed report with the following
	j) Design methodology and assumptions

	k) Compare the results of manual calculation results with the software results
	l) Prepare the bill of quantity(BoQ) with brief description of key data

20. Design of a 100MWp Grid-connected Solar PV Plant

Description:

Design a 100MWp grid-connected Solar PV Plant and Illustrate for Ranchi (Jharkhand) location. Design shall be performing manual calculation as well as simulation tools.

Consider the following data for the design:

- a typical Solar Panel of standard manufacturer(Si-Mono): Pmax=400Wp, Vmp=40.3V, Imp=9.9A, Voc=49.0, Isc=10.4A and efficiency=19.89%
- A 1200kWac centralised inverter

Any missing data can be assumed with relevance to the problem and the same to be brought out in the report.

Software Tools: PVSyst

Tasks:

S.No	Description
1	Prepare a worksheet and find the required quantities for that meteo location
2	Using a suitable platform, find the Number of Solar PV modules
3	Also evaluate for the pnom ratio and annual yield
4	Prepare the layout of the solar plant including schematic diagram
5	Prepare the detailed report with the following
	m) Design methodology and assumptions
	n) Compare the results of manual calculation results with the software results
	o) Prepare the bill of quantity(BoQ) with brief description of key data

SB8044 SURFACE MODELLING

L	T	P	C
1	0	2	2

COURSE OBJECTIVES:

Understand the fundamentals of Generative Shape Design and its application in creating wireframe geometry.

- Develop proficiency in the extruding technique to transform wireframe geometries into solid models.
- Gain a comprehensive understanding of surface modeling principles and techniques.
- Explore the impact of surface relimitations on aesthetics, functionality, and manufacturability.
- Foster a quality-driven mindset by incorporating surface check tools into the design validation process.

UNIT 1: GENERATIVE SHAPE DESIGN 9

Introduction to Surface Design- Advantages of using generative shape design- Designing a surface-based feature- Managing features- Introduction to ordered geometrical set- Geometrical set versus ordered geometrical set- About hybrid design-Creating a Reference Geometry-Introduction to Reference Geometry- Creating Reference Geometries- Introduction to local geometries- Creating Multiple axis systems.

UNIT 2: **CREATING A WIREFRAME GEOMETRY AND EXTRUDING 9**

Creating 3D curves - About the curve continuity- About the impact of tension- creating curves from scratch- Creating a 3D Corner- Creating a Boundary- Projecting elements- Creating parallel curve **-** Extruding and revolving a profile and sweeping a profile - Selecting profiles for extrude and revolve- Extruding a profile - Revolving a profile- Extruding/revolving a profile using the context toolbar-Creating a surface offset from a Reference-Creating an offset surface- Sweeping a profile**-** Importance of the Spine- Calculating a spine using a plane or guide curve- About explicit Sweep- About conical sweep

UNIT 3: CREATING SURFACES 9

Creating a Multi-Section and an Adaptive Sweep Surface - Creating Multi-section surface- Create an Adaptive Sweep Surface-Calculating an Adaptive Sweep- Adaptive Sweep and simple Sweep- Constrained Sketch- Common errors when computing the Sweep- Create an adaptive Sweeping using Existing Surface

UNIT 4: SURFACE RELIMITATIONS 9

Surface Re-Limitation and Connection - About the Re-limitation of Surfaces- Commands to Re- limit Surfaces and Curves- About the split Command- Splitting Elements- About the Trim Command- Trimming an Element- About Extrapolation of Elements- Create a Car Door handle - **Connecting Surfaces Smoothly** - Understanding the need to connect the Surfaces smoothly- Types of fillets- Managing Shape fillets extremities- Creating an Edge Fillet Using the Context Toolbar- Criteria for Selecting the Blends and the Fillets- Exercise: Create the Fillets for Existing Surface

UNIT 5: SURFACE CHECK TOOLS 9

Checking the Surface Continuity - Understanding the need of Surface Continuity- Exercise: Analyze the surface Continuity- Types of Flaws- Detecting the Geometric Connection in the Surfaces- Connection Analysis- Analyzing Surface - Curve Connections- Healing Operation

TOTAL: 45 PERIODS

OUTCOMES:

At the end of this course the student will be able to:

- The fundamentals of Generative Shape Design and its application in creating wireframe geometry.
- Develop proficiency in the extruding technique to transform wireframe geometries into solid models.
- Gain a comprehensive understanding of surface modeling principles and techniques.
- Explore the impact of surface relimitations on aesthetics, functionality, and manufacturability.
- a quality-driven mindset by incorporating surface check tools into the design validation process.

REFERNCES:

https://www.3ds.com/

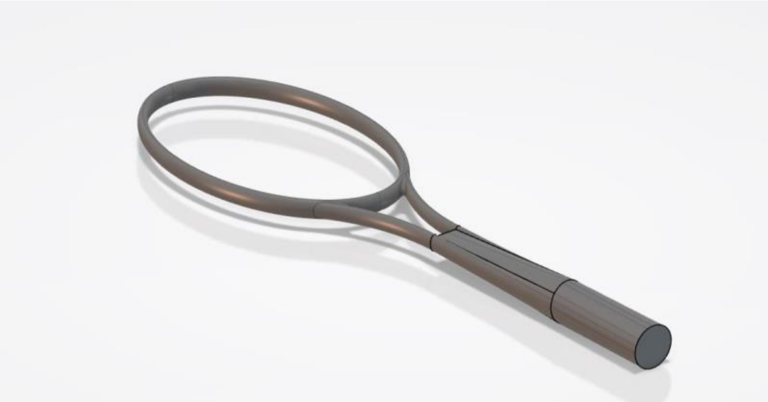
https://eduspace.3ds.com/

LIST OF EQUIPMENTS FOR BATCH OF 5O STUDENTS			
S NO	**Type**	**Description**	**Qty**
1	Workstation	32 GB RAM, Intel i7 Processor, 4 GB Graphics Card, Operating System 250GB, Storage 250GB hard disk, Mouse, Keyboard.	50
2	Projector	LCD Projector or Smart Board	1
3	Internet	50 to 100 Mbps High Speed Internet Connection with Wi-Fi facility	1
4	Computer Lab	Computer lab with 50 Students Capacity	1
5	Software	Dassault Systemes 3DEXPERIENCE R2022x Hotfix6.28	50

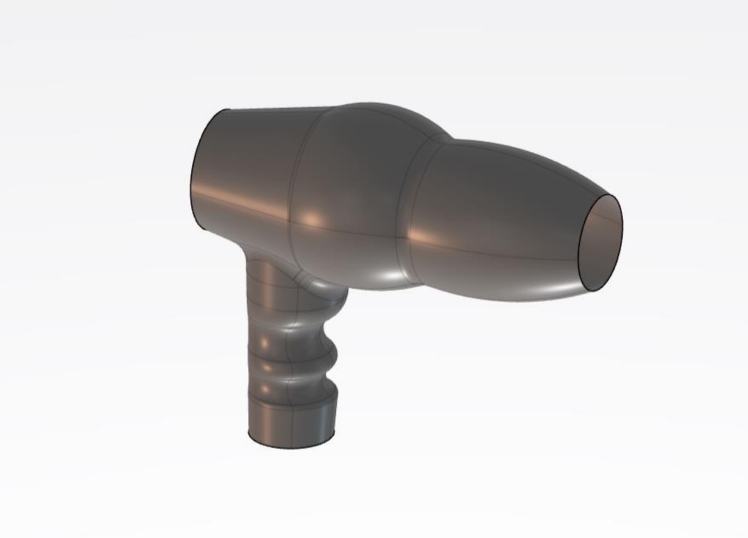
INDSUTRY SCOPE:

Automotive, Industrial Equipment, Aerospace, Agriculture, Electric Vehicle, Manufacturing, Production, Heavy Machinery industry.

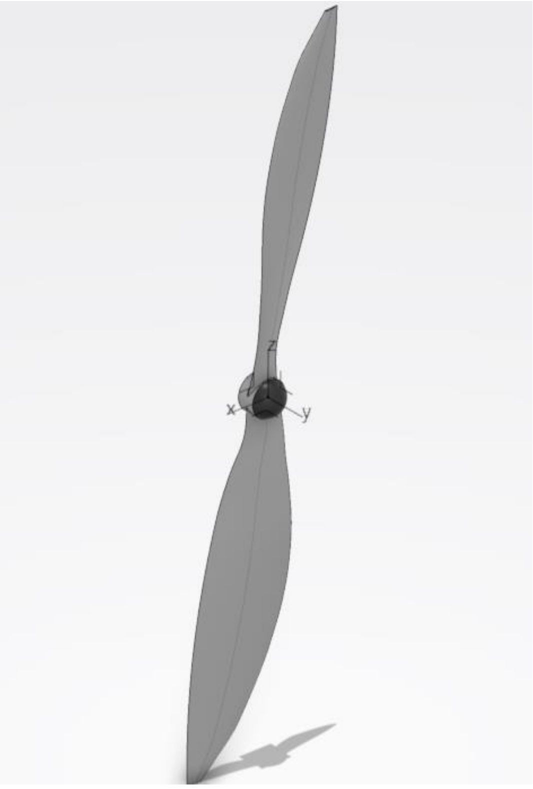
TASK 1: Design of Badminton bat frame.



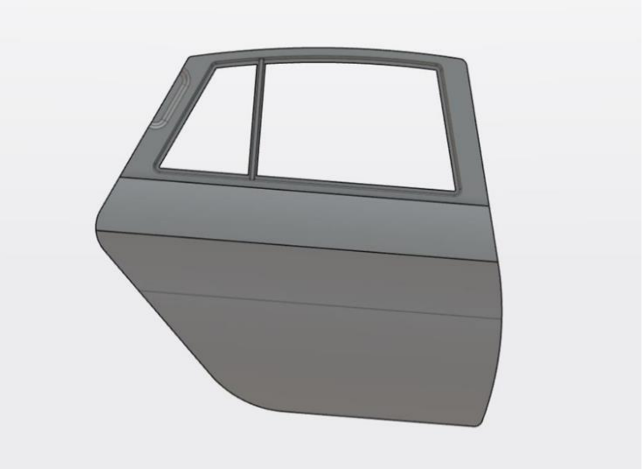
TASK 2: Design of Dryer



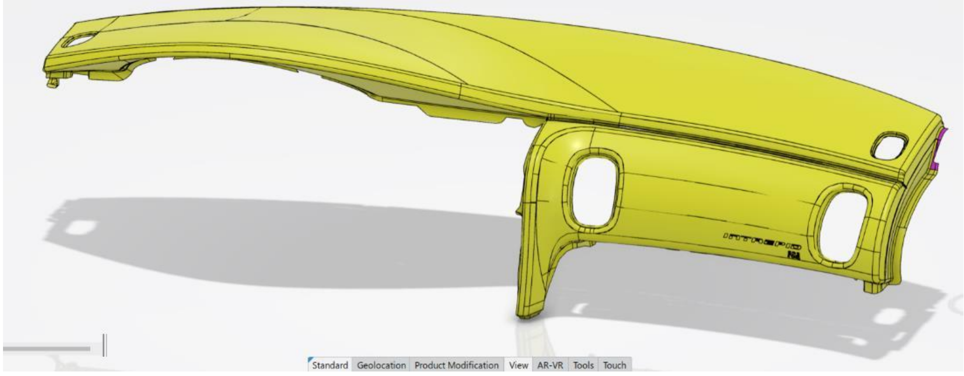
TASK 3: Design of Aircraft Prope TASK 4: Design of Car door panel.



TASK 4: Design of Car door panel.



TASK 5: Design of Car Dashboard



INDUSTRY USE CASES

1. Automotive industry: Creating and refining complex car body designs, including exterior surfaces and aerodynamic components.
2. Aerospace industry: Designing aircraft fuselages, wings, and engine nacelles with precise surface curvature and aerodynamic characteristics.
3. Product design and consumer goods: Developing consumer products such as electronics, appliances, and furniture with sleek and ergonomic surface designs.
4. Industrial machinery: Designing complex machinery and equipment with smooth and functional surface profiles for efficient operation.
5. Architecture and construction: Creating 3D models of buildings, structures, and interiors with accurate surface representation for visualization and design evaluation.
6. Medical devices: Designing prosthetics, orthopedic implants, and medical equipment with anatomically accurate and patient-specific surface shapes.
7. Jewelry and accessories: Designing intricate and detailed surfaces for jewelry, watches, and fashion accessories.
8. Packaging industry: Designing product packaging with attractive and functional surface textures, patterns, and branding elements.
9. Consumer electronics: Designing smartphones, tablets, laptops, and other electronic devices with visually appealing and ergonomic surface forms.
10. Furniture design: Creating stylish and comfortable furniture designs with visually pleasing surface contours and textures.
11. Sports equipment: Designing sports equipment like bicycles, helmets, and tennis rackets with optimized aerodynamics and performance-enhancing surfaces.
12. Film and entertainment industry: Creating visual effects and character models with detailed and realistic surface textures for movies, games, and animations.
13. Industrial design: Designing industrial equipment, tools, and machinery with optimized surface profiles for improved functionality and ergonomics.
14. Packaging design: Creating innovative and visually appealing packaging designs for food, beverages, cosmetics, and other consumer products.
15. Art and sculpture: Designing and visualizing artistic sculptures and installations with intricate and unique surface forms.
16. Footwear design: Designing comfortable and stylish footwear with optimized surface shapes and contours for proper fit and aesthetics.

17. Marine and shipbuilding industry: Designing ship hulls and marine structures with hydrodynamic surface shapes for optimal performance and fuel efficiency.
18. Interior design: Creating 3D models of interior spaces with accurately represented surface materials, textures, and finishes.
19. Consumer appliances: Designing appliances such as refrigerators, washing machines, and kitchen gadgets with user-friendly and visually appealing surface designs.
20. Sustainable design: Designing environmentally friendly products with optimized surface forms to minimize material waste and improve resource efficiency.

SB8045 INFORMATION AND COMMUNICATION TECHNOLOGY (ICT) ENGINEER – 5G NETWORKS

L	T	P	C
1	0	2	2

COURSE OBJECTIVE

Students will be able to:

- Apply their knowledge of information and communication technology to design, deploy, and optimize 5G networks.
- Gain expertise in various aspects of 5G networks, including network architecture, protocols, security, and performance optimization.
- Develop skills in network planning, troubleshooting, and the integration of new technologies in 5G networks.

COURSE CONTENT:

Semester 1:

UNIT I: Introduction to 5G Networks (3 + 6)

- Overview of 5G network architecture, components, and services.
- Understanding the key features and requirements of 5G networks.
- Introduction to 5G protocols and interfaces.

UNIT II: 5G Network Planning and Deployment (3 + 6)

- Network planning methodologies for 5G networks.
- Site selection, coverage analysis, and capacity planning.
- Deployment considerations and challenges in 5G networks.

UNIT III: 5G Network Protocols and Security (3 + 6)

- Understanding the protocols used in 5G networks.
- Security mechanisms and protocols in 5G networks.
- Testing and validating the security of 5G networks.

UNIT IV: 5G Network Performance Optimization (3 + 6)

- Performance metrics and key performance indicators in 5G networks.
- Optimization techniques for coverage, capacity, and quality of service.
- Analyzing and improving the performance of 5G networks.

UNIT V: Emerging Technologies in 5G Networks (3 + 6)

- Introduction to emerging technologies such as network slicing, edge computing, and Internet of Things (IoT) in 5G networks.
- Integration and optimization of emerging technologies in 5G networks.
- Evaluating the impact of emerging technologies on network performance.

Semester 2:

UNIT VI: Advanced 5G Network Planning and Optimization (3 + 6)

- Advanced network planning techniques for 5G networks.
- Optimization of network parameters and configurations.
- Troubleshooting and resolving performance issues in 5G networks.

UNIT VII: 5G Network Automation and Management (3 + 6)

- Automation frameworks and tools for managing 5G networks.
- Network management protocols and procedures.
- Implementing network automation for efficient network operations.

UNIT VIII: 5G Network Virtualization and Cloud Computing (3 + 6)

- Virtualization concepts and technologies in 5G networks.
- Cloud computing platforms and their integration in 5G networks.
- Deploying and managing virtualized network functions in 5G networks.

UNIT IX: 5G Network Testing and Validation (3 + 6)

- Testing methodologies and tools for 5G networks.
- Performance testing, load testing, and interoperability testing.
- Validation of 5G network functionalities and performance.

UNIT X: Research and Innovation in 5G Networks (3 + 6)

- Exploring emerging trends and research topics in 5G networks.
- Conducting research projects and experiments in 5G networks.
- Presenting and documenting research findings.

TOTAL: 45 PERIODS

COURSE OUTCOME:

- Students will be able to design, deploy, and optimize 5G networks using information and communication technology.
- Gain expertise in network planning, troubleshooting, and performance optimization of 5G networks.
- Understand the security aspects and protocols used in 5G networks.
- Explore emerging technologies and their integration in 5G networks.

REFERENCES:

1. "5G Networks: Powering Digitalization" by Afif Osseiran, Jose F. Monserrat, and Patrick Marsch.
2. "Wireless Communications and Networks: 5G and Beyond" by Cheng-Xiang Wang, Guangjie Han, and Zhiqin Wang.

SOFTWARE REQUIREMENT:

- Simulation tools for 5G network planning, optimization, and testing.
- Network management and automation platforms.

HARDWARE REQUIREMENT:

- 5G network equipment or simulation environments for practical implementation and testing.

INDUSTRY SCOPE:

Upon completion of this course, students can pursue careers as ICT engineers in the telecommunication industry, focusing on 5G networks. They can work in network planning and optimization, network operations and management, or research and development roles in organizations that are involved in 5G network deployments and innovations.

Semester 1: Use Cases for Information and Communication Technology (ICT) Engineer - 5G Networks

1. Use Case: Network Capacity Planning

1. Test Case 1: Validate the accuracy of traffic prediction models.
2. Test Case 2: Test the scalability of the network.
3. Test Case 3: Evaluate the effectiveness of capacity expansion strategies.
4. Test Case 4: Assess the impact of network congestion on latency and throughput.
5. Test Case 5: Test the performance of load balancing algorithms.

2. Use Case: Network Functionality Testing

1. Test Case 1: Validate the functionality of network protocols.
2. Test Case 2: Test the handover procedures between base stations.

3. Test Case 3: Evaluate the performance of connection establishment and release procedures.
4. Test Case 4: Assess the impact of network handovers on user experience.
5. Test Case 5: Test the reliability and performance of network signaling protocols.

3. Use Case: Network Energy Efficiency

1. Test Case 1: Validate the effectiveness of power-saving mechanisms.
2. Test Case 2: Test the impact of energy-saving features on network performance.
3. Test Case 3: Evaluate the accuracy of energy consumption models.
4. Test Case 4: Assess the effectiveness of power optimization techniques.
5. Test Case 5: Test the resilience of energy-saving mechanisms.

4. Use Case: Network Synchronization

1. Test Case 1: Verify the accuracy and stability of time synchronization mechanisms.
2. Test Case 2: Test the performance of synchronization protocols.
3. Test Case 3: Evaluate the resilience of synchronization mechanisms against network delays.
4. Test Case 4: Assess the synchronization accuracy in supporting time-sensitive applications.
5. Test Case 5: Test the synchronization recovery mechanisms after network failures.

5. Use Case: Network Security and Privacy

1. Test Case 1: Validate the implementation of authentication and encryption mechanisms.
2. Test Case 2: Test the effectiveness of access control mechanisms.
3. Test Case 3: Evaluate the privacy protection measures for user data.
4. Test Case 4: Assess the resilience of network security measures against attacks.
5. Test Case 5: Test the performance of security protocols.

6. Use Case: Network Slicing and Service Isolation

1. Test Case 1: Validate the creation and management of network slices.
2. Test Case 2: Test the isolation mechanisms for network slices.
3. Test Case 3: Evaluate the performance of network slicing in supporting diverse services.
4. Test Case 4: Assess the scalability of network slicing for differentiated services.
5. Test Case 5: Test the resource optimization algorithms for network slices.

7. Use Case: Network Automation and Orchestration

1. Test Case 1: Verify the automation of network provisioning and configuration processes.
2. Test Case 2: Test the orchestration of network services and resources.
3. Test Case 3: Evaluate the performance of network automation platforms.
4. Test Case 4: Assess the scalability and reliability of network automation frameworks.
5. Test Case 5: Validate the integration of artificial intelligence for intelligent network management.

8. Use Case: Network Virtualization and Edge Computing

1. Test Case 1: Validate the deployment and management of virtualized network functions.
2. Test Case 2: Test the performance of virtual network services in terms of latency and throughput.
3. Test Case 3: Evaluate the scalability and elasticity of virtualized network resources.
4. Test Case 4: Assess the effectiveness of edge computing in reducing network latency.
5. Test Case 5: Test the fault tolerance mechanisms in virtualized network environments.

9. Use Case: Network Performance Optimization

1. Test Case 1: Evaluate the performance of optimization algorithms for resource allocation.
2. Test Case 2: Test the effectiveness of interference mitigation techniques.
3. Test Case 3: Assess the impact of mobility management strategies on network performance.
4. Test Case 4: Validate the effectiveness of Quality of Service (QoS) mechanisms.
5. Test Case 5: Test the performance of handover optimization algorithms.

10. Use Case: Network Resilience and Disaster Recovery

1. Test Case 1: Validate the resilience mechanisms for fault detection and recovery.
2. Test Case 2: Test the effectiveness of backup and redundancy mechanisms.
3. Test Case 3: Evaluate the performance of network restoration procedures.
4. Test Case 4: Assess the impact of disaster scenarios on network availability.
5. Test Case 5: Validate the disaster recovery mechanisms for network services.

Semester 2: Advanced Use Cases for Information and Communication Technology (ICT) Engineer - 5G Networks:

1. Use Case: Network Slicing for Ultra-Reliable Low-Latency Communication (URLLC)

1. Test Case 1: Validate the end-to-end latency requirement for URLLC network slices.
2. Test Case 2: Test the reliability and availability of URLLC network slices under high network load.
3. Test Case 3: Evaluate the impact of network congestion on URLLC service delivery.
4. Test Case 4: Assess the resilience of URLLC network slices against multiple simultaneous failures.
5. Test Case 5: Validate the synchronization and coordination mechanisms for URLLC services.

2. Use Case: Multi-Access Edge Computing (MEC) Integration

1. Test Case 1: Test the scalability and elasticity of MEC resources for demanding applications.
2. Test Case 2: Validate the performance of edge computing applications under high user load.
3. Test Case 3: Assess the fault tolerance and disaster recovery mechanisms of MEC environments.
4. Test Case 4: Evaluate the security and privacy measures in MEC deployments.
5. Test Case 5: Test the interoperability and integration of MEC platforms with 5G networks.

3. Use Case: Network Intelligence and Machine Learning

1. Test Case 1: Validate the accuracy and efficiency of machine learning algorithms for network optimization.
2. Test Case 2: Test the performance of anomaly detection mechanisms in complex network scenarios.
3. Test Case 3: Evaluate the impact of intelligent network decisions on overall network performance.
4. Test Case 4: Assess the scalability and adaptability of network intelligence algorithms.
5. Test Case 5: Test the robustness and resilience of network intelligence systems against adversarial attacks.

4. Use Case: Network Slicing for IoT Applications

1. Test Case 1: Validate the scalability and performance of IoT network slices for a massive number of devices.
2. Test Case 2: Test the reliability and availability of IoT network slices under varying traffic patterns.
3. Test Case 3: Evaluate the security and privacy measures for IoT network slices.
4. Test Case 4: Assess the resource allocation and dynamic adaptation of IoT network slices.
5. Test Case 5: Validate the integration and interoperability of IoT devices with network slices.

5. Use Case: Network Automation and Cognitive Networks

1. Test Case 1: Verify the automation and self-configuration of network elements in complex network topologies.
2. Test Case 2: Test the cognitive decision-making capabilities of network management systems in real-time scenarios.
3. Test Case 3: Evaluate the performance of self-healing mechanisms in response to multiple simultaneous failures.
4. Test Case 4: Assess the impact of cognitive networks on overall network efficiency and reliability under high load.
5. Test Case 5: Test the resilience and adaptability of cognitive network algorithms in dynamic network environments.

6. Use Case: Massive Machine-Type Communication (mMTC) Optimization

1. Test Case 1: Validate the scalability of mMTC networks for a massive number of machine-type devices.

2. Test Case 2: Test the performance of resource allocation algorithms for efficient mMTC communication.
3. Test Case 3: Evaluate the impact of interference management techniques on mMTC network capacity.
4. Test Case 4: Assess the energy efficiency and power consumption of mMTC networks.
5. Test Case 5: Validate the reliability and latency requirements of mMTC services.

7. Use Case: Network Security and Privacy

1. Test Case 1: Test the effectiveness of encryption and authentication mechanisms in securing 5G networks.
2. Test Case 2: Evaluate the resilience of network security measures against sophisticated cyber-attacks.
3. Test Case 3: Assess the privacy preservation mechanisms in user data handling within the network.
4. Test Case 4: Validate the performance of intrusion detection and prevention systems in real-time scenarios.
5. Test Case 5: Test the compliance of network security measures with industry standards and regulations.

8. Use Case: Network Function Virtualization (NFV) Orchestration

1. Test Case 1: Test the scalability and performance of virtual network functions (VNFs) in NFV deployments.
2. Test Case 2: Validate the orchestration and management of VNFs across multiple NFV infrastructure nodes.
3. Test Case 3: Assess the fault tolerance and disaster recovery mechanisms in NFV environments.
4. Test Case 4: Evaluate the resource optimization and elasticity of VNFs in dynamic network scenarios.
5. Test Case 5: Test the interoperability and compatibility of VNFs from different vendors.

9. Use Case: Cloud-RAN and Virtualized Radio Access Networks

1. Test Case 1: Validate the performance and efficiency of Cloud-RAN deployments in providing high-quality mobile services.
2. Test Case 2: Test the scalability and elasticity of virtualized radio access networks under varying user density.
3. Test Case 3: Assess the synchronization and timing accuracy of virtualized RAN environments.
4. Test Case 4: Evaluate the impact of network congestion on the performance of virtualized RAN.
5. Test Case 5: Validate the integration and compatibility of virtualized RAN with 5G core networks.

10. Use Case: Network Slicing for Entertainment and Media Services

1. Test Case 1: Test the quality of service (QoS) and quality of experience (QoE) for media streaming services in network slices.
2. Test Case 2: Validate the scalability and performance of network slices for live event broadcasting.
3. Test Case 3: Evaluate the reliability and availability of network slices for real-time gaming applications.
4. Test Case 4: Assess the resource allocation and optimization for high-bandwidth media services in network slices.
5. Test Case 5: Test the seamless handover and mobility support for entertainment services across network slices.

These use cases and test cases for the second semester are designed to provide advanced challenges and reflect the progress made by the students in understanding and applying the concepts of 5G networks.

SB8046 SYSTEM ARCHITECT – 5G CLOUD RAN **L T P C**
1 0 2 2

COURSE OBJECTIVE

Students will be able to:

- Apply their knowledge to design and architect 5G networks with a focus on Cloud RAN.
- Utilize advanced techniques and technologies in 5G network synchronization, security, network slicing, edge computing, and automation.
- Develop expertise in the deployment and orchestration of virtualized network functions in Cloud RAN environments.

COURSE CONTENT

Semester 1:

UNIT I: Introduction to 5G Networks and Cloud RAN (3 + 6)

- Overview of 5G network architecture and components.
- Introduction to Cloud RAN and its advantages.
- Understanding the concept of network virtualization.

UNIT II: Cloud RAN Architecture and Configuration (3 + 6)

- Cloud RAN architecture and functional components.
- Configuration and deployment of Cloud RAN platforms.
- Integration of Cloud RAN with existing network infrastructure.

UNIT III: Network Synchronization in 5G Cloud RAN (3 + 6)

- Synchronization requirements and challenges in Cloud RAN environments.
- Deployment and configuration of synchronization mechanisms.
- Testing and validation of synchronization accuracy and performance.

UNIT IV: Security in 5G Cloud RAN (3 + 6)

- Security threats and vulnerabilities in Cloud RAN.
- Implementing security measures and protocols.
- Testing and evaluating the effectiveness of security mechanisms.

UNIT V: Network Slicing and Edge Computing in 5G Cloud RAN (3 + 6)

- Understanding network slicing and its applications in Cloud RAN.
- Integration of edge computing platforms in Cloud RAN.
- Testing and optimizing network slicing and edge computing performance.

Semester 2:

UNIT VI: Virtualized Network Functions (VNF) in 5G Cloud RAN (3 + 6)

- Virtualization technologies and concepts.
- Deployment and orchestration of VNFs in Cloud RAN.
- Performance testing and optimization of virtualized network functions.

UNIT VII: Advanced Network Automation and Orchestration (3 + 6)

- Automation frameworks and tools for Cloud RAN.
- Orchestration of network operations and management tasks.

- Advanced testing and validation of automated Cloud RAN networks.

UNIT VIII: Artificial Intelligence (AI) in 5G Cloud RAN (3 + 6)

- AI techniques and algorithms for network optimization in Cloud RAN.
- Integration of AI-driven decision-making in Cloud RAN operations.
- Testing and evaluating the impact of AI on network performance.

UNIT IX: Advanced Analytics and Big Data in 5G Cloud RAN (3 + 6)

- Utilizing big data analytics for network optimization in Cloud RAN.
- Processing and analysis of large-scale data in Cloud RAN.
- Evaluating the effectiveness of analytics-driven decision-making.

UNIT X: Research and Innovation in 5G Cloud RAN (3 + 6)

- Exploring emerging trends and technologies in 5G Cloud RAN.
- Conducting research and innovation projects in Cloud RAN.
- Presenting and documenting research findings.

TOTAL: 45 PERIODS

COURSE OUTCOME

- Students will be able to design and architect 5G networks using Cloud RAN technologies.
- Develop expertise in advanced topics such as network slicing, edge computing, and virtualized network functions.
- Gain hands-on experience in testing, validation, and optimization of 5G Cloud RAN deployments.
- Understand the industry scope in embedded & IoT-based products, location tracking, safety, public health, and the automotive industry.

REFERENCES

- "5G System Design: An End-to-End Perspective" by Wan Lei and Zhang Hua.
- "Cloud-Radio Access Networks (Cloud-RAN) - Principles, Technologies, and Applications" by Tony Q. S. Quek, Yong Ngee, and Ben G. H. Gwee.

SOFTWARE REQUIREMENT:

- Simulation tools for 5G network architecture and Cloud RAN platforms.
- Network configuration and orchestration tools.

HARDWARE REQUIREMENT:

- Cloud RAN platforms or simulation environments.
- 5G network equipment for practical implementation and testing.

INDUSTRY SCOPE: Upon completion of this course, students can explore career opportunities in the telecommunication industry, particularly in the design, implementation, and optimization of 5G networks. They can work as 5G network architects, system engineers, or researchers in organizations focusing on 5G technologies and Cloud RAN deployments.

Semester 1: Use Cases for System Architect - 5G Cloud RAN:

1. Use Case: Cloud RAN Deployment Planning

1. Test Case 1: Validate the selection of optimal Cloud RAN deployment locations based on coverage and capacity requirements.
2. Test Case 2: Test the scalability and resource allocation for different user density scenarios in Cloud RAN.
3. Test Case 3: Evaluate the impact of network interference and latency on Cloud RAN performance.
4. Test Case 4: Assess the synchronization and timing accuracy of Cloud RAN nodes.
5. Test Case 5: Validate the integration and interoperability of Cloud RAN with the existing network infrastructure.

2. **Use Case: Virtualization of RAN Functions**

1. Test Case 1: Test the performance and scalability of virtualized RAN functions under varying user loads.
2. Test Case 2: Validate the efficient utilization of hardware resources in virtualized RAN environments.
3. Test Case 3: Evaluate the fault tolerance and resilience of virtualized RAN functions.
4. Test Case 4: Assess the impact of network congestion on virtualized RAN performance.
5. Test Case 5: Test the interoperability and compatibility of virtualized RAN functions with different vendors.

3. **Use Case: Network Slicing for Cloud RAN**

1. Test Case 1: Validate the end-to-end latency and throughput requirements for different network slices in Cloud RAN.
2. Test Case 2: Test the resource allocation and isolation mechanisms for network slices in Cloud RAN.
3. Test Case 3: Evaluate the performance and reliability of network slicing for different service types.
4. Test Case 4: Assess the scalability and elasticity of network slices in Cloud RAN.
5. Test Case 5: Validate the seamless handover and mobility support across network slices in Cloud RAN.

4. **Use Case: Centralized Baseband Unit (BBU) Pooling**

1. Test Case 1: Test the pooling and allocation of BBU resources in centralized architectures.
2. Test Case 2: Validate the load balancing and dynamic resource optimization in BBU pooling scenarios.
3. Test Case 3: Evaluate the impact of network congestion on BBU pooling performance.
4. Test Case 4: Assess the fault tolerance and resilience of BBU pooling architectures.
5. Test Case 5: Test the interoperability and compatibility of BBUs from different vendors in pooled environments.

5. **Use Case: Network Synchronization in Cloud RAN**

1. Test Case 1: Validate the synchronization mechanisms for distributed radio units in Cloud RAN architectures.
2. Test Case 2: Test the accuracy and stability of time synchronization in Cloud RAN networks.
3. Test Case 3: Evaluate the impact of synchronization errors on network performance and coverage.
4. Test Case 4: Assess the synchronization resilience in the presence of network disruptions or failures.
5. Test Case 5: Validate the interoperability of synchronization protocols across different network elements.

6. **Use Case: Dynamic Spectrum Sharing (DSS) in Cloud RAN**

1. Test Case 1: Test the efficiency and fairness of spectrum allocation in DSS-enabled Cloud RAN.
2. Test Case 2: Validate the interference management and coexistence mechanisms in DSS scenarios.
3. Test Case 3: Evaluate the impact of dynamic spectrum sharing on network capacity and performance.
4. Test Case 4: Assess the scalability and resource optimization in DSS-enabled Cloud RAN networks.
5. Test Case 5: Test the interoperability of DSS implementations with legacy network equipment.

7. **Use Case: Network Resilience and Disaster Recovery in Cloud RAN**

1. Test Case 1: Validate the failover and recovery mechanisms in case of Cloud RAN node failures.
2. Test Case 2: Test the robustness and reliability of backup and redundancy solutions in Cloud RAN.
3. Test Case 3: Evaluate the impact of network disruptions on service continuity and restoration time.
4. Test Case 4: Assess the fault detection and isolation capabilities in Cloud RAN architectures.
5. Test Case 5: Validate the interoperability and compatibility of disaster recovery mechanisms with different vendors.

8. Use Case: Energy Efficiency in Cloud RAN

1. Test Case 1: Test the energy consumption and optimization strategies in Cloud RAN deployments.
2. Test Case 2: Validate the performance trade-offs between energy efficiency and network capacity in Cloud RAN.
3. Test Case 3: Evaluate the impact of traffic load and user density on energy consumption in Cloud RAN.
4. Test Case 4: Assess the effectiveness of power-saving mechanisms in virtualized RAN environments.
5. Test Case 5: Test the interoperability of energy-saving features across different network elements.

9. Use Case: Security and Privacy in Cloud RAN

1. Test Case 1: Validate the authentication and access control mechanisms in Cloud RAN architectures.
2. Test Case 2: Test the encryption and privacy preservation mechanisms for user data in Cloud RAN.
3. Test Case 3: Evaluate the resilience of Cloud RAN against cyber-attacks and intrusions.
4. Test Case 4: Assess the security of communication channels and interfaces in Cloud RAN.
5. Test Case 5: Validate the compliance of Cloud RAN security measures with industry standards and regulations.

10. Use Case: Performance Monitoring and Optimization in Cloud RAN

1. Test Case 1: Test the monitoring and analysis of key performance indicators (KPIs) in Cloud RAN.
2. Test Case 2: Validate the optimization algorithms and techniques for improving network performance in Cloud RAN.
3. Test Case 3: Evaluate the impact of traffic load and user behavior on Cloud RAN performance.
4. Test Case 4: Assess the effectiveness of self-optimizing network (SON) mechanisms in Cloud RAN.
5. Test Case 5: Test the interoperability and compatibility of performance monitoring tools with different vendors.

These use cases and test cases for the first semester of the System Architect - 5G Cloud RAN course are designed to provide a solid foundation in the fundamental concepts and practices of Cloud RAN architecture.

Semester 2: Use Cases for System Architect - 5G Cloud RAN

1. Use Case: Network Slicing Orchestration in Cloud RAN

1. Test Case 1: Validate the dynamic allocation and management of resources for network slices in Cloud RAN.
2. Test Case 2: Test the scalability and performance of network slicing orchestration algorithms in complex scenarios.

3. Test Case 3: Evaluate the end-to-end latency and throughput guarantees for network slices in Cloud RAN.
4. Test Case 4: Assess the resilience and fault tolerance of network slicing orchestration mechanisms.
5. Test Case 5: Test the interoperability and compatibility of network slicing orchestration solutions with multiple vendors.

2. Use Case: Edge Computing Integration in Cloud RAN

1. Test Case 1: Validate the deployment and orchestration of edge computing resources in Cloud RAN architectures.
2. Test Case 2: Test the performance and latency improvements achieved through edge computing integration.
3. Test Case 3: Evaluate the scalability and resource allocation for edge computing tasks in Cloud RAN.
4. Test Case 4: Assess the interoperability and compatibility of edge computing platforms with Cloud RAN components.
5. Test Case 5: Test the resilience and fault recovery mechanisms in edge computing-enabled Cloud RAN.

3. Use Case: Network Function Virtualization (NFV) in Cloud RAN

1. Test Case 1: Validate the virtualization and orchestration of network functions in Cloud RAN environments.
2. Test Case 2: Test the performance and scalability of virtualized network functions in high-demand scenarios.
3. Test Case 3: Evaluate the security and isolation mechanisms for virtualized network functions in Cloud RAN.
4. Test Case 4: Assess the interoperability and compatibility of NFV solutions with Cloud RAN components.
5. Test Case 5: Test the resilience and fault recovery mechanisms in NFV-enabled Cloud RAN.

4. Use Case: Multi-Access Edge Computing (MEC) in Cloud RAN

1. Test Case 1: Validate the deployment and integration of MEC platforms with Cloud RAN architectures.
2. Test Case 2: Test the performance improvements and latency reduction achieved through MEC integration.
3. Test Case 3: Evaluate the scalability and resource allocation for MEC tasks in Cloud RAN.
4. Test Case 4: Assess the interoperability and compatibility of MEC platforms with Cloud RAN components.
5. Test Case 5: Test the resilience and fault recovery mechanisms in MEC-enabled Cloud RAN.

5. Use Case: Network Automation and Orchestration in Cloud RAN

1. Test Case 1: Validate the automation and orchestration of network operations and management tasks in Cloud RAN.
2. Test Case 2: Test the efficiency and accuracy of automated network provisioning and configuration in Cloud RAN.
3. Test Case 3: Evaluate the impact of network automation on operational costs and resource utilization in Cloud RAN.
4. Test Case 4: Assess the resilience and fault recovery mechanisms in automated Cloud RAN networks.
5. Test Case 5: Test the interoperability and compatibility of network automation solutions with Cloud RAN components.

6. Use Case: Artificial Intelligence (AI) in Cloud RAN Optimization

1. Test Case 1: Validate the use of AI algorithms for network performance optimization in Cloud RAN.
2. Test Case 2: Test the accuracy and efficiency of AI-based resource allocation and optimization in Cloud RAN.
3. Test Case 3: Evaluate the impact of AI-driven decision-making on network capacity and user experience in Cloud RAN.
4. Test Case 4: Assess the resilience and fault tolerance of AI-enabled Cloud RAN architectures.
5. Test Case 5: Test the interoperability and compatibility of AI solutions with Cloud RAN components.

7. Use Case: Advanced Security Mechanisms in Cloud RAN

1. Test Case 1: Validate the effectiveness of advanced encryption and privacy preservation techniques in Cloud RAN.
2. Test Case 2: Test the resilience of Cloud RAN against advanced cyber threats and attacks.
3. Test Case 3: Evaluate the performance impact of advanced security mechanisms on Cloud RAN operations.
4. Test Case 4: Assess the interoperability and compatibility of advanced security solutions with Cloud RAN components.
5. Test Case 5: Validate the compliance of advanced security measures with industry standards and regulations.

8. Use Case: 5G Network Synchronization in Cloud RAN

1. Test Case 1: Validate the synchronization mechanisms for distributed time and frequency references in Cloud RAN.
2. Test Case 2: Test the accuracy and reliability of synchronization protocols in dynamic Cloud RAN deployments.
3. Test Case 3: Evaluate the impact of synchronization errors on network performance and user experience in Cloud RAN.
4. Test Case 4: Assess the interoperability and compatibility of synchronization solutions with Cloud RAN components.
5. Test Case 5: Test the resilience and fault tolerance of synchronization mechanisms in Cloud RAN.

9. Use Case: Cloud-native Application Development in Cloud RAN

1. Test Case 1: Validate the deployment and management of cloud-native applications in Cloud RAN environments.
2. Test Case 2: Test the performance and scalability of cloud-native applications in Cloud RAN architectures.
3. Test Case 3: Evaluate the interoperability and compatibility of cloud-native application frameworks with Cloud RAN components.
4. Test Case 4: Assess the resilience and fault recovery mechanisms in cloud-native Cloud RAN applications.
5. Test Case 5: Test the security and isolation mechanisms for cloud-native applications in Cloud RAN.

10. Use Case: Advanced Analytics and Big Data in Cloud RAN

1. Test Case 1: Validate the use of advanced analytics and big data techniques for network optimization in Cloud RAN.
2. Test Case 2: Test the accuracy and efficiency of data-driven decision-making in Cloud RAN operations.
3. Test Case 3: Evaluate the scalability and performance of big data processing frameworks in Cloud RAN.

4. Test Case 4: Assess the interoperability and compatibility of analytics and big data solutions with Cloud RAN components.
5. Test Case 5: Test the privacy preservation mechanisms for sensitive data in Cloud RAN analytics.

These use cases and test cases for the second semester of the System Architect - 5G Cloud RAN course are designed to provide advanced knowledge and skills in the application and optimization of Cloud RAN architectures. They reflect the industry's evolving requirements and the need for students to demonstrate their progress in understanding complex concepts and solving advanced challenges in Cloud RAN deployments.

SB8047 LINUX ADMINISTRATION FUNDAMENTALS L T P C
1 0 2 2

Unit No.	Name	Theory Hours	Practical Hours	Total Hours
1	Introduction to Linux	2	2	4
2	Manage Users and File Access	3	3	6
3	Tuning System Performance and Software	5	5	10
4	Manage Storage Stacks	5	5	10
5	Manage Network and Containers	5	5	10
	Total Hours	20	20	**40**

Pre-Requisites:

- Basic Knowledge of Hardware & Networking
- Basic System OS Knowledge

Requirements

Hardware:

- 16GB RAM
- i3/i5 processor
- Network Access

Software:

- Red Hat version 9.0 ISO Image

Skill Objectives:

- Package management with new repository structure and Appstream modules
- Create storage devices, volumes, and file systems, including Stratis storage management
- Configure network services and security
- Manage processes, scheduling, and tuning
- Manage users, groups, and authentication
- Perform server management with the Cockpit web management utility
- Troubleshoot and obtain support
- Run containers

CONTENT

UNIT I INTRODUCTION TO LINUX 04 Hours
Theory 02 Hours

Edit Text Files from the Shell Prompt - Configure SSH Key-based Authentication – Describe Linux File System Hierarchy Concepts - Make Links Between Files - Match File Names with Shell Expansions

Skill Training **02 Hours**

Guided Exercise: Edit Text Files from the Shell Prompt
Guided Exercise: Configure SSH Key-based Authentication
Guided Exercise: Make Links Between Files
Guided Exercise: Match File Names with Shell Expansions

UNIT II MANAGE USERS AND FILE ACCESS **06 Hours**
Theory **03 Hours**

Describe User and Group Concepts - Gain Superuser Access - Manage Local User Accounts – Manage Local Group Accounts - Manage User Passwords- Manage File System Permissions from the Command Line - Manage Default Permissions and File Access

Skill Training **03 Hours**

Guided Exercise: Gain Superuser Access
Guided Exercise: Manage Local User Accounts
Guided Exercise: Manage Local Group Accounts Guided Exercise: Manage User Password
Guided Exercise: Manage File System Permissions from the Command Line
Guided Exercise: Manage Default Permissions and File Access

UNIT III TUNING SYSTEM PERFORMANCE AND SOFTWARE MANAGEMENT **10 Hours**

Theory **05 Hours**

Kill Processes - Monitor Process Activity - Monitor Process Activity - Schedule Recurring User Jobs -Register Systems for Red Hat Support - Install and Update Software Packages with DNF - Enable DNF Software Repositories - - Control System Services - Select the Boot Target – Reset the Root Password

Skill Training **05 Hours**

Guided Exercise: Kill Processes
Guided Exercise: Monitor Process Activity
Guided Exercise: Schedule Recurring User Jobs
Guided Exercise: Install and Update Software Packages with DNF Guided Exercise: Enable DNF Software Repositories
Guided Exercise: Control System Services Guided Exercise: Select the Boot Target Guided Exercise: Reset the Root Password

UNIT IV MANAGE SECURITY & STORAGE **10 Hours**

Theory **05 Hours**

Change the SELinux Enforcement Mode - Control SELinux File Contexts - Adjust SELinux Policy with Booleans - Investigate and Resolve SELinux Issues - Mount and Unmount File Systems - Add Partitions, File Systems, and Persistent Mounts - Manage Swap Space - Create and Extend Logical Volumes - Describe System Log Architecture - Review Syslog & System Journal Entries - Preserve the System Journal - Maintain Accurate Time

Skill Training **05 Hours**

Guided Exercise: Change the SELinux Enforcement Mode
Guided Exercise: Control SELinux File Contexts
Guided Exercise: Adjust SELinux Policy with Booleans Guided Exercise: Investigate
Guided Exercise: Control SELinux File Contexts
Guided Exercise: Control SELinux File Contexts
Guided Exercise: Adjust SELinux Policy with Booleans Guided Exercise:
Investigate and Resolve SELinux Issues Guided Exercise: Mount and Unmount File Systems Guided Exercise: Add Partitions, File Systems
Guided Exercise: Manage Swap Space
Guided Exercise: Create and Extend Logical Volumes
Guided Exercise: Review Syslog & System Journal Entries Guided Exercise: Preserve the System Journal
Guided Exercise: Maintain Accurate Time

UNIT V MANAGE NETWORK AND CONTAINERS 10 Hours

Theory 05 Hours

Validate Network Configuration - Configure Networking from the Command Line - Edit Network Configuration Files - Configure Hostnames and Name Resolution - Manage Network-Attached Storage with NFS - Manage Server Firewalls - Run Containers - Deploy Containers - Manage Container Storage and Network Resources - Manage Containers as System Services

Skill Training 05 Hours

Guided Exercise: Validate Network Configuration
Guided Exercise: Configure Networking from the Command Line
Guided Exercise: Edit Network Configuration Files
Guided Exercise: Configure Hostnames and Name Resolution
Guided Exercise: Manage Server Firewalls
Guided Exercise: Manage Network-Attached Storage with NFS Guided Exercise: Deploy Containers
Guided Exercise: Manage Container Storage and Network Resources
Guided Exercise: Manage Containers as System Services

TOTAL: 45 Hours

Skill Outcomes:

1. Understand and use essential tools for handling files, directories, command-line environments, and documentation
2. Create simple shell scripts
3. Operate running systems, including booting into different run levels, identifying processes, starting and stopping virtual machines, and controlling services
4. Configure local storage using partitions and logical volumes
5. Create and configure file systems and file system attributes, such as permissions, encryption, access control lists, and network file systems
6. Deploy, configure, and maintain systems, including software installation, update, and core services
7. Manage users and groups
8. Manage security, including basic firewall and SELinux configuration
9. Perform basic container management

REFERENCES

1. Ted Jordan - CompTIA Linux+ Certification All-in-One Exam Guide-McGraw-Hill Education (2020)
2. Christine Bresnahan_ Richard Blum - LPIC-1 _ Linux Professional Institute Certification study guide (2021)
3. Christopher Negus - Linux Bible, 10th Edition
4. Sander van Vugt - Red Hat RHCSA 8 EX200 Certification Guide, 2nd edition

Test Project 1:

Title:

Configuring the Secure Webserver

Description:

Web services, in the general meaning of the term, are services offered via the Web. In a typical Web services scenario, a user application sends a request to a service at a given URL using the protocol of HTTP. The service receives the request, processes it, and returns a response.

Test Project 2:

Title:

Create a MySQL database as a Container

Description:

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings

Test Project 3:

Title:

Deploy the Kickstart server

Description:

deploy the kickstart server for installing the RHEL 9 operating system in a client machine as un-attendant installation.

Industry Scope:

Network Engineer, System Administrator, Software Developer, Linux Administrator, Cloud Operator

SB8048 INTRODUCTION TO CHATGPT AND BUILDING APPLICATIONS USING CHATGPT

L T P C
1 0 2 2

Module 1: Python Basics and ChatGPT Fundamentals 10

Introduction to Python programming language: history, features, and applications- Setting up the Python development environment: installing Python, IDEs, and code editors- Variables, data types, and basic operations: numbers, strings, lists, tuples, dictionaries- Control flow statements: if-else, loops (for and while), and conditional expressions- Functions and modules: defining and using functions, creating and importing modules- File handling in Python: reading from and writing to files, file modes and operations- Exception handling: handling errors and exceptions using try-except blocks, handling multiple exceptions- Introduction to ChatGPT and its architecture: transformer models, attention mechanism - Exploring the core components of ChatGPT: input encoding, attention layers, output decoding

Module 2: Ethical Considerations and Bias Detection 8

Applications and benefits of ChatGPT: virtual assistants, customer support, content generation- Potential challenges and risks associated with ChatGPT: biases, misinformation, harmful content- Strategies for mitigating risks and ensuring ethical use of ChatGPT: dataset curation, monitoring, user feedback- Best practices for building an ethical ChatGPT system: transparency, explainability, user consent- Data Quality Strategies for ChatGPT: data preprocessing, validation, and augmentation- Principles for Using ChatGPT Responsibly: promoting inclusive and unbiased conversations, user privacy, and security- Creating effective prompts for ChatGPT: formulating clear and specific questions, instructions, and context- Analyzing GPT response quality: evaluating coherence, relevance, and factual accuracy- Detecting plagiarism in ChatGPT's responses: techniques for identifying copied or unoriginal content- Measuring bias and discrimination in ChatGPT's responses: identifying and addressing bias, fairness considerations

Module 3: Natural Language Processing and Content Creation 8

Overview of NLP and its relevance to ChatGPT: text preprocessing, tokenization, and language modeling- Using ChatGPT for content creation: generating text, creative writing, blog post generation- Leveraging ChatGPT for scriptwriting and code troubleshooting: generating code snippets, debugging assistance- Prompt design and code completion: structuring prompts for code-related queries, leveraging autocomplete features- Image generation with OpenAI DALLE2: understanding the principles and applications of DALLE for image synthesis

Module 4: Whisper for Speech Recognition and DALLE for Image Generation 9

Understanding the Whisper API for speech recognition: features, benefits, and use cases- Exploring the architecture and capabilities of Whisper models: deep learning models for speech-to-text conversion- Integrating Whisper for real-time transcription services: handling audio inputs, transcribing and processing speech data- Overview of the DALLE model for image generation: generative models for creating unique and creative images- Understanding the principles and applications of DALLE: image synthesis based on textual prompts and concepts- Integrating DALLE for generating images based on user prompts: leveraging pretrained models, controlling image attributes

Module 5: Streamlit and API Integration 10

Introduction to Streamlit and its features: building web-based applications with Python- Designing user interfaces with Streamlit components: layout design, widgets, buttons, and input fields- Integrating ChatGPT, Whisper, and DALLE with Streamlit for interactive AI applications: connecting backend APIs, handling user inputs and model outputs- Deploying ChatGPT, Whisper, and DALLE applications to cloud platforms: deployment strategies, cloud services, scalability considerations- Introduction to Application Programming Interfaces (APIs): understanding APIs and their role in software development

TOTAL : 45 PERIODS

Books:

1. The Power of ChatGPT: How to Put ChatGPT to Work for You by Willis Turner
2. The Art of Prompt Engineering with ChatGPT A Hands on Guide by Nathan Hunter

Online References:

1. **Chat GPT: https://openai.com/product/chatgpt**
2. **DALL-E: https://openai.com/product/dall-e-2**
3. **Whisper: https://openai.com/research/whisper**

6. Sample 20 Use Cases

Use Case 1: Document Summarizer-ChatGPT API

Description:

The project aims to develop a document summarizer using the ChatGPT API. The application will allow users to input a document or a text passage, and the ChatGPT API will generate a concise summary of the input. The application will provide a user-friendly interface for text input, processing, and displaying the generated summary.

Tasks:

1. Streamlit Development: Design and implement the user interface for the document summarizer application using Streamlit. Create input fields for text input, such as a textarea or file upload widget, and display the generated summary in a readable format.

2. Backend Development: Develop the server-side logic using Python. Handle user requests and integrate with the ChatGPT API.

3. ChatGPT API Integration: Understand and integrate the ChatGPT API into the application for document summarization. Handle API authentication, input format, and response processing.

4. Text Processing: Pre-process and clean user input text to improve summarization results. Handle text formatting, punctuation, and special characters.

5. Model Output Processing: Handle the output of the ChatGPT API, which may include multiple response segments. Extract the relevant summary segment and format it for display.

6. Deployment: Deploy the application to a web server or hosting platform to make it accessible to users over the internet.

Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. ChatGPT API Integration: Understanding the ChatGPT API and integrating it into the application for document summarization. Exploring the API's capabilities and limitations.

2. Streamlit Development: Designing and implementing a user interface for the document summarizer application using Streamlit, a Python library for building web applications. Creating input fields for text input and displaying the generated summary.

3. Backend Development: Building the server-side logic using Python. Handling user requests, integrating with the ChatGPT API, and processing the generated summary.

4. Text Processing: Pre-processing and cleaning user input text to remove unnecessary elements and ensure better summarization results. Handling text formatting, punctuation, and special characters.

5. Model Output Processing: Handling the output of the ChatGPT API, which may include multiple response segments. Extracting the relevant summary segment and formatting it for display.

6. Deployment: Deploying the application to a web server or hosting platform to make it accessible to users over the internet.

Evaluation:

The evaluation will consist of the following components:

1. MCQ Questions: A set of 30 multiple-choice questions covering the technologies and concepts used in the project, such as API integration, Streamlit development, backend development, text processing, and deployment.

2. Live Evaluation: Industrial experts/faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts/faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 2: AI Job Interview Simulation with evaluation-ChatGPT API
Description:

The project aims to develop an AI job interview simulation with evaluation using the ChatGPT API. The application will simulate a job interview scenario where users can interact with an AI interviewer and answer a series of interview questions. The ChatGPT API will generate interview questions and provide responses based on the user's answers. Additionally, the application will evaluate the user's performance and provide feedback based on their answers and overall performance in the interview.

Tasks:

1. Streamlit Development: Design and implement the user interface for the job interview simulation application using Streamlit. Create interactive components for answering interview questions, displaying feedback, and evaluating the user's performance.

2. Backend Development: Develop the server-side logic using Python. Handle user interactions, integrate with the ChatGPT API, process interview questions and responses, and evaluate the user's performance.

3. ChatGPT API Integration: Understand and integrate the ChatGPT API into the application for simulating an AI job interview. Handle API authentication, question generation, response processing, and feedback generation.

4. Interview Question Generation: Utilize the ChatGPT API to generate interview questions based on predefined criteria. Implement strategies to ensure diversity and relevance in the generated questions.

5. User Response Processing: Handle user responses to interview questions and process them to generate appropriate feedback. Analyze user answers, provide constructive feedback, and evaluate performance based on predefined evaluation criteria.

6. Deployment: Deploy the application to a web server or hosting platform to make it accessible to users over the internet.

Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. ChatGPT API Integration: Understanding the ChatGPT API and integrating it into the application for simulating an AI job interview. Exploring the API's capabilities and limitations in generating interview questions and responses.

2. Streamlit Development: Designing and implementing a user interface for the job interview simulation application using Streamlit, a Python library for building web applications. Creating interactive components for answering interview questions, displaying feedback, and evaluating the user's performance.

3. Backend Development: Building the server-side logic using Python. Handling user interactions, integrating with the ChatGPT API, processing interview questions and responses, and evaluating the user's performance.

4. Interview Question Generation: Utilizing the ChatGPT API to generate interview questions based on predefined criteria. Implementing strategies to ensure diversity and relevance in the generated questions.
5. User Response Processing: Handling user responses to interview questions and processing them to generate appropriate feedback. Analyzing user answers, providing constructive feedback, and evaluating performance based on predefined evaluation criteria.

6. Deployment: Deploying the application to a web server or hosting platform to make it accessible to users over the internet.

Evaluation:

The evaluation will consist of the following components:

1. MCQ Questions: A set of 30 multiple-choice questions covering the technologies and concepts used in the project, such as API integration, Streamlit development, backend development, question generation, user response processing, and evaluation criteria.

2. Live Evaluation: Industrial experts/faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts/faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 3: Customer Support Chatbot using ChatGPT API

Description:

The project aims to develop a customer support chatbot using the ChatGPT API. The chatbot will provide automated assistance and support to customers by responding to their queries and providing relevant information. The ChatGPT API will generate responses based on the user's inputs and predefined knowledge base or conversation flows. The application will have a user-friendly interface for customers to interact with the chatbot and receive timely assistance.

Tasks:

1. Streamlit Development: Design and implement the user interface for the customer support chatbot application using Streamlit. Create an interactive chat interface for users to communicate with the chatbot.

2. Backend Development: Develop the server-side logic using Python. Handle user interactions, integrate with the ChatGPT API, and process user queries.

3. ChatGPT API Integration: Understand and integrate the ChatGPT API into the application for building a customer support chatbot. Handle API authentication, query processing, and response generation.

4. Natural Language Processing: Implement techniques for natural language processing to understand user queries and generate appropriate responses. Apply techniques such as tokenization, intent recognition, and entity extraction.

5. Knowledge Base Integration: Integrate a predefined knowledge base or conversation flows into the chatbot to provide accurate and relevant information to customers. Implement mechanisms to retrieve information from a database or external APIs.

6. Testing and Improvements: Conduct comprehensive testing of the chatbot's functionality and performance. Identify areas for improvement and implement enhancements to ensure better user experience and accurate responses.

Learning Outcome:
Through this project, you will gain experience and understanding in the following areas:
1. ChatGPT API Integration: Understanding the ChatGPT API and integrating it into the application for building a customer support chatbot. Exploring the API's capabilities and limitations in generating responses for user queries.

2. Streamlit Development: Designing and implementing a user interface for the customer support chatbot application using Streamlit, a Python library for building web applications. Creating an interactive chat interface for users to communicate with the chatbot.

3. Backend Development: Building the server-side logic using Python. Handling user interactions, integrating with the ChatGPT API, and processing user queries.

4. Natural Language Processing: Implementing techniques for natural language processing, including tokenization, intent recognition, and entity extraction, to understand user queries and generate appropriate responses.
5. Knowledge Base Integration: Integrating a predefined knowledge base or conversation flows into the chatbot to provide accurate and relevant information to customers. This may involve database integration or API calls to retrieve information.

6. Testing and Improvements: Conducting thorough testing of the chatbot's functionality and performance. Identifying areas for improvement and implementing enhancements to ensure better user experience and accurate responses.

Evaluation:
The evaluation will consist of the following components:
1. MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as API integration, Streamlit development, backend development, natural language processing, knowledge base integration, and testing.

2. Live Evaluation: Industrial experts/faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts/faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 4: Insight Report Generation from Data using ChatGPT
Description:
The project aims to develop an Insight Report Generation application that utilizes ChatGPT to generate insightful reports from a given dataset. The application will provide a user-friendly interface for users to input their data and generate detailed reports based on the analyzed data. The reports will include key findings, trends, patterns, correlations, and other relevant insights derived from the data.

Tasks:
1. Streamlit Development: Design and implement the user interface for the Insight Report Generation application using Streamlit. Create data input forms and result display components.

2. Backend Development: Develop the server-side logic using Python. Handle user requests, data processing, and integrate the data analysis and ChatGPT model into the application.

3. Data Pre-processing: Clean, transform, and pre-process the dataset, handling missing values, outliers, and data inconsistencies. Apply techniques like imputation, normalization, and

feature scaling.
4. Data Analysis: Apply data analysis techniques, such as descriptive statistics, data visualization, and exploratory data analysis, to extract insights from the dataset.

5. Insight Report Generation: Utilize the results of data analysis to generate insightful reports using ChatGPT. Summarize key findings, trends, and patterns in a written format.

6. Application Integration: Connect the data analysis, ChatGPT model, and report generation components to the Streamlit frontend of the web application. Incorporate the analysis results and generated reports into the application's logic for processing user input and providing insights.
Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. Data Analysis: Understanding various data analysis techniques such as descriptive statistics, data visualization, exploratory data analysis, and correlation analysis. Applying these techniques to extract meaningful insights from the data.

2. Natural Language Processing (NLP): Exploring NLP concepts and techniques, including text generation, summarization, and language understanding. Understanding how to leverage NLP models like ChatGPT for generating insightful reports.

3. Data Pre-processing: Cleaning and pre-processing the dataset to handle missing values, outliers, and data inconsistencies. Applying techniques such as data imputation, **normalization, and feature scaling to ensure the data's quality and suitability for analysis.**

4. Insight Generation: Utilizing the analyzed data to generate meaningful insights and summarizing them in a concise and informative manner. Understanding how to int erpret the findings and translate them into written reports.

5. Application Development: Designing and implementing a user-friendly interface for the Insight Report Generation application using Streamlit, a Python library for building web applications. Building the server-side logic using Python to handle user input, data analysis, and report generation.

Evaluation:
The evaluation will consist of the following components:

1. MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as data analysis techniques, data pre-processing, NLP, and application development.

2. Live Evaluation: Industrial experts/faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts/faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 5: AI-Based Language Proficiency Assessment using ChatGPT

Description:

The project aims to develop an AI-based Language Proficiency Assessment application using ChatGPT. The application will provide an automated assessment of an individual's language proficiency in a specific language, such as English, French, or Spanish. Users will interact with the application by answering questions or engaging in conversations, and the AI model will analyze their responses to assess their language skills.

Tasks:

1. Streamlit Development: Design and implement the user interface for the Language Proficiency Assessment application using Streamlit. Create input forms for user responses and result display components.

2. Backend Development: Develop the server-side logic using Python. Handle user requests, process user responses, and integrate the ChatGPT model for language assessment.

3. Dataset Preparation: Curate or collect a dataset of language assessment questions or prompts, covering various aspects of language proficiency. Ensure diversity in question types and difficulty levels.

4. Model Training and Fine-tuning: Train or fine-tune the ChatGPT model using the curated dataset to align its responses with accurate language assessment. Optimize the model's performance for language proficiency evaluation.

5. Language Proficiency Assessment: Utilize the trained ChatGPT model to assess users' language proficiency based on their responses. Evaluate grammar, vocabulary, comprehension, and fluency levels.

6. Application Integration: Connect the language assessment components to the Streamlit frontend of the web application. Incorporate the assessment results into the application's logic for processing user input and providing language proficiency scores.

Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. Natural Language Processing (NLP): Understanding the fundamentals of NLP, including text processing, language understanding, and language generation. Familiarizing yourself with NLP models like ChatGPT and their capabilities for language assessment.

2. Language Proficiency Assessment: Learning about different techniques and metrics used in language proficiency assessment, such as grammar, vocabulary, comprehension, and fluency. Understanding how to evaluate language skills based on user responses.

3. Dataset Preparation: Curating or collecting a dataset of language assessment questions or prompts suitable for evaluating language proficiency. Ensuring diversity in question types and difficulty levels.

4. Model Training and Fine-tuning: Training or fine-tuning the ChatGPT model using the curated dataset to align its responses with accurate language assessment. Optimizing the model's performance for language proficiency evaluation.

5. Application Development: Designing and implementing a user-friendly interface for the Language Proficiency Assessment application using Streamlit, a Python library for building web applications. Building the server-side logic using Python to handle user input, language assessment, and result generation.

Evaluation:

The evaluation will consist of the following components:

1. MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as NLP, language proficiency assessment, dataset preparation, model training, and application development.

2. Live Evaluation: Industrial experts/faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts/faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 6: Virtual Travel Expert with ChatGPT Intelligence

Description:

The project aims to develop a Travel Assistant application that utilizes ChatGPT to provide assistance and information to travelers. The application will serve as a virtual travel guide, offering recommendations, answering questions, and providing relevant information about destinations, flights, accommodations, local attractions, transportation, and more.

Tasks:

1. Streamlit Development: Design and implement the user interface for the Travel Assistant application using Streamlit. Create a conversational interface where users can interact with the virtual travel guide.

2. Backend Development: Develop the server-side logic using Python. Handle user requests, process queries, integrate travel-related data sources, and utilize the ChatGPT model for generating responses.

3. Data Integration: Integrate and utilize travel-related data sources, such as flight APIs, hotel APIs, tourism websites, and travel blogs, to gather information and provide accurate recommendations.

4. Conversational User Interface: Design and implement a conversational flow that allows users to ask questions, seek recommendations, and receive relevant information in a natural and interactive manner.

5. Contextual Understanding: Enhance the ChatGPT model's contextual understanding capabilities to maintain context and provide coherent responses during conversations. Handle follow-up questions and maintain a consistent user experience.

6. Application Integration: Connect the data sources, ChatGPT model, and conversational components to the Streamlit frontend of the Travel Assistant application. Incorporate the model's responses and data sources into the application's logic for processing user input and providing travel-related assistance.

Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. Natural Language Processing (NLP): Understanding the fundamentals of NLP and how it can be applied to develop conversational agents. Familiarizing yourself with NLP models like ChatGPT for generating responses and understanding user queries.

2. Travel Domain Knowledge: Acquiring knowledge about popular travel destinations, flights, accommodations, local attractions, transportation options, travel itineraries, and travel-related information.

3. Data Integration: Integrating and utilizing travel-related data sources, such as flight APIs, hotel APIs, tourism websites, and travel blogs, to gather information and provide accurate recommendations.

4. Conversational User Interface: Designing and implementing a user-friendly conversational interface for the Travel Assistant application using Streamlit, a Python library for building web applications. Building a conversational flow that allows users to ask questions, seek recommendations, and receive relevant information in a natural and interactive manner.

5. Contextual Understanding: Enhancing the ChatGPT model's contextual understanding capabilities to maintain context and provide coherent responses during conversations. Handling follow-up questions and maintaining a consistent user experience.

6. Application Development: Developing the server-side logic using Python to handle user input, process queries, integrate data sources, and generate appropriate responses using the ChatGPT model.

Evaluation:

The evaluation will consist of the following components:

1. MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as NLP, travel domain knowledge, data integration, conversational interface design, and application development.

2. Live Evaluation: Industrial experts/faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts/faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 7: Virtual Study Buddy Using ChatGPT

Description:

The project aims to develop a Virtual Study Buddy application that utilizes ChatGPT to assist students in their learning journey. The application will provide a virtual study companion that can answer questions, provide explanations, offer study tips, and engage in interactive learning conversations with students.

Tasks:

1. Streamlit Development: Design and implement the user interface for the Virtual Study Buddy application using Streamlit. Create an interactive and user-friendly environment for students to engage with the study companion.

2. Backend Development: Develop the server-side logic using Python. Handle user requests, process study-related queries, integrate educational resources, and utilize the ChatGPT model for generating responses.

3. Educational Content Integration: Integrate and utilize educational resources, including textbooks, lecture notes, online references, and study materials, to gather information and provide accurate explanations and study assistance.

4. Conversational User Interface: Design and implement a conversational flow that encourages students to ask questions, seek explanations, and engage in interactive learning conversations with the Virtual Study Buddy.

5. Contextual Understanding: Enhance the ChatGPT model's contextual understanding capabilities to maintain context and provide coherent responses during study sessions. Handle follow-up questions and maintain a consistent learning experience.

6. Application Integration: Connect the educational resources, ChatGPT model, and conversational components to the Streamlit frontend of the Virtual Study Buddy application. Incorporate the model's responses and educational resources into the application's logic for processing student input and providing study assistance.

Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. Natural Language Processing (NLP): Understanding the fundamentals of NLP and its applications in educational contexts. Familiarizing yourself with ChatGPT and its usage for generating responses and understanding student queries.

2. Educational Content Knowledge: Acquiring knowledge in various subject areas, including mathematics, science, literature, history, and more. Understanding common student difficulties and misconceptions in different subjects.

3. Learning Companion Design: Designing and implementing an interactive and engaging user interface for the Virtual Study Buddy application using Streamlit, a Python library for building web applications. Creating a conversational flow that encourages students to ask questions, seek explanations, and receive study assistance.

4. Contextual Understanding: Enhancing ChatGPT's contextual understanding capabilities to provide coherent responses and maintain context during study sessions. Handling follow- up questions and maintaining a consistent learning experience.

5. Adaptive Learning: Incorporating adaptive learning techniques into the Virtual Study Buddy application. Personalizing the study experience based on individual student needs, progress, and preferences.

6. Application Development: Developing the server-side logic using Python to handle user input, process study-related queries, integrate educational resources, and generate appropriate responses using the ChatGPT model.

Evaluation:

The evaluation will consist of the following components:

1. MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as NLP, educational content knowledge, conversational interface design, adaptive learning, and application development.

2. Live Evaluation: Industrial experts/faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts/faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 8: Coding Assistance using ChatGPT

Description:

The project aims to develop a Coding Assistance application that utilizes ChatGPT to provide support and guidance to developers during their coding process. The application will serve as a virtual coding companion, helping developers with code suggestions, error explanations, debugging tips, and general programming assistance.

Tasks:

1. Frontend Development: Design and implement the user interface for the Coding Assistance application using Streamlit, creating an intuitive and interactive environment for developers to interact with the virtual coding companion.

2. Backend Development: Develop the server-side logic using Python and a backend framework like Flask or FastAPI. Handle user requests, process code-related queries, integrate code analysis tools, and utilize the ChatGPT model for generating coding assistance.

3. Code Analysis Integration: Integrate and utilize code analysis tools and libraries to analyze code structures, identify errors, and provide coding suggestions. Incorporate code analysis results into the application's logic for generating relevant responses.

4. Conversational User Interface: Design and implement a conversational flow using

Streamlit that encourages developers to ask coding-related questions, seek suggestions, and engage in interactive coding assistance conversations with the virtual companion.

5. Contextual Understanding: Enhance the ChatGPT model's contextual understanding capabilities to maintain code context, understand code-specific queries, and generate accurate coding assistance. Handle follow-up questions and maintain a consistent coding experience.

6. Application Integration: Connect the code analysis tools, ChatGPT model, and conversational components to the Streamlit frontend of the Coding Assistance application. Incorporate the model's responses and code analysis results into the application's logic for processing developer input and providing coding assistance.

Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. Natural Language Processing (NLP): Understanding the fundamentals of NLP and its applications in coding assistance. Familiarizing yourself with ChatGPT and its usage for generating code suggestions and understanding developer queries.

2. Programming Languages: Acquiring knowledge and proficiency in one or more programming languages, such as Python, Java, JavaScript, or C++. Understanding common coding patterns, best practices, and debugging techniques.

3. Code Analysis: Learning techniques for analyzing and understanding code structures, identifying errors, and providing suggestions for improvement. Familiarizing yourself with code analysis tools and libraries.

4. Contextual Understanding: Enhancing ChatGPT's contextual understanding capabilities to provide coherent and relevant coding assistance. Handling code-specific queries, understanding code context, and generating accurate suggestions.

5. Application Development: Developing the server-side logic using Python and a backend framework like Flask or FastAPI to handle user input, process code-related queries, integrate code analysis tools, and generate appropriate responses using the ChatGPT model.

Evaluation:

The evaluation will consist of the following components:

1. MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as NLP, programming languages, code analysis, conversational interface design, and application development.

2. Live Evaluation: Industrial experts/Faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts/Faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 9: Documentation Generator for Code using ChatGPT

Description:
The project aims to develop a Documentation Generator application that utilizes ChatGPT to automate the process of generating documentation for code. The application will assist developers in creating comprehensive and user-friendly documentation by analyzing code structures, extracting relevant information, and generating corresponding documentation sections.

Tasks:
1. Frontend Development: Design and implement the user interface for the Documentation Generator application using Streamlit, creating an intuitive and user-friendly environment for developers to interact with the code documentation generation tool.
2. Backend Development: Develop the server-side logic using Python and a backend framework like Flask or FastAPI. Handle user requests, analyze code structures, extract relevant information, and utilize the ChatGPT model for generating documentation sections.

3. Code Analysis Integration: Integrate and utilize code analysis tools and libraries to parse code, extract relevant information, and understand code context. Incorporate code analysis results into the application's logic for generating accurate documentation sections.

4. Documentation Generation: Utilize the ChatGPT model to generate natural language descriptions based on code analysis results. Generate documentation sections for functions, classes, variables, and code usage examples.

5. Documentation Formatting: Implement formatting and structuring techniques to ensure the generated documentation follows standard practices. Add appropriate headings, sections, and code snippets for better readability and usability.

6. Application Integration: Connect the code analysis tools, ChatGPT model, and documentation generation components to the frontend of the Documentation Generator application. Incorporate the model's responses and code analysis results into the application's logic for processing code input and generating documentation.

Learning Outcome:
Through this project, you will gain experience and understanding in the following areas:

1. Natural Language Processing (NLP): Understanding the fundamentals of NLP and its applications in code documentation generation. Familiarizing yourself with ChatGPT and its usage for generating natural language descriptions based on code analysis.

2. Programming Languages: Acquiring knowledge and proficiency in one or more programming languages, such as Python, Java, JavaScript, or C++. Understanding code structures, syntax, and relevant documentation practices.

3. Code Analysis: Learning techniques for analyzing code structures, extracting relevant information, and understanding code context. Familiarizing yourself with code parsing, abstract syntax trees (AST), and code analysis libraries.
4. Documentation Standards: Understanding common documentation standards and best practices, such as documenting functions, classes, variables, and code usage examples. Learning how to structure and format documentation for readability and usability.

5. Application Development: Developing the server-side logic using Python and a backend framework like Flask or FastAPI to handle code analysis, process code-related queries, integrate code parsing tools, and generate documentation using the ChatGPT model.

Evaluation:
The evaluation will consist of the following components:

1. MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as NLP, programming languages, code analysis, documentation standards, and application development.

2. Live Evaluation: Industrial experts/Faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.
3. Feedback: The industrial experts/Faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 10: News Summarizer using ChatGPT

Description:
The project aims to develop a News Summarizer application using ChatGPT. The application will analyze news articles from various sources and generate concise summaries that capture the key information and main points of the articles. Users will be able to input the URL or text of a news article, and the ChatGPT model will generate a summary based on its understanding of the content.

Tasks:

1. Frontend Development: Design and implement the user interface for the News Summarizer application using Streamlit. Create input forms for user-provided URLs or text and a display component for the generated summaries.

2. Backend Development: Develop the backend logic using Python and ChatGPT. Handle user requests, process the provided URLs or text, and utilize the ChatGPT model for generating summaries.

3. Dataset Preparation: Curate or collect a dataset of news articles suitable for training and evaluating the ChatGPT model for summarization. Ensure diversity in topics and sources to cover a wide range of news content.
4. Model Training and Fine-tuning: Train or fine-tune the ChatGPT model using the curated dataset to align its responses with accurate summarization. Optimize the model's performance for generating concise and informative summaries.

5. News Summarization: Utilize the trained ChatGPT model to generate summaries of news articles based on user-provided URLs or text. Extract the key information and main points from the articles to create concise summaries.

6. Application Integration: Connect the summarization components to the Streamlit frontend of the News Summarizer application. Incorporate the generated summaries into the application's logic for processing user input and displaying the results.

Learning Outcome:
Through this project, you will gain experience and understanding in the following areas:

1. Natural Language Processing (NLP): Understanding the fundamentals of NLP, including text processing, language understanding, and language generation. Familiarizing yourself with NLP models like ChatGPT and their capabilities for text summarization.

2. Text Summarization Techniques: Learning about different text summarization techniques, such as extractive and abstractive summarization. Understanding their strengths and limitations and selecting an appropriate approach for the News Summarizer application.

3. Dataset Preparation: Curating or collecting a dataset of news articles suitable for training and evaluating the ChatGPT model for summarization. Ensuring diversity in topics and sources to provide a comprehensive coverage of news content.
4. Model Training and Fine-tuning: Training or fine-tuning the ChatGPT model using the curated dataset to align its responses with accurate summarization. Optimizing the model's performance for generating concise and informative summaries.

5. Application Development: Designing and implementing a user-friendly interface for the News Summarizer application using Streamlit. Building the backend logic using Python to handle user input, article processing, and summary generation with the ChatGPT model.

Evaluation:
The evaluation will consist of the following components:

1. MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as NLP, text summarization techniques, dataset preparation, model training, and application development.

2. Live Evaluation: Industrial experts/Faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts/Faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.
Use Case 11: Event Planner using ChatGPT

Description:
The project aims to develop an Event Planner application using ChatGPT. The application will assist users in planning various types of events, such as parties, conferences, weddings, and more. Users will interact with the application through a user-friendly interface and the ChatGPT model will provide recommendations, suggestions, and assistance in organizing and managing the event.

Tasks:

1. Frontend Development: Design and implement the user interface for the Event Planner application using Streamlit. Create input forms for user preferences, event details, and a display component for recommendations and suggestions.

2. Backend Development: Develop the backend logic using Python and ChatGPT. Handle user requests, process event details, and utilize the ChatGPT model to generate recommendations, suggestions, and assistance.

3. Event Planning Knowledge Integration: Acquire and integrate event planning knowledge into the application. Gather information about venues, vendors, catering services, decorations, and entertainment options to provide accurate recommendations and suggestions.

4. Conversational User Interface: Design and implement a conversational flow that allows users to input their event preferences, receive recommendations, ask questions, and receive assistance throughout the event planning process.

5. Contextual Understanding: Enhance the ChatGPT model's contextual understanding capabilities to maintain context and provide coherent responses during event planning conversations. Handle follow-up questions and maintain a consistent user experience.

6. Application Integration: Connect the event planning components to the Streamlit frontend of the Event Planner application. Incorporate the model's responses and event planning knowledge into the application's logic for processing user input and providing event planning assistance.

Learning Outcome:
Through this project, you will gain experience and understanding in the following areas:
1. Natural Language Processing (NLP): Understanding the fundamentals of NLP and its applications in event planning. Familiarizing yourself with ChatGPT and its usage for generating event-related suggestions and recommendations.

2. Event Planning Domain Knowledge: Acquiring knowledge about event planning processes, including venue selection, budgeting, guest management, catering, decorations, entertainment, and logistics.
3. User Interface Design: Designing and implementing a user-friendly interface for the Event Planner application using Streamlit. Creating input forms for user preferences, event details, and display components for recommendations and suggestions.

4. Contextual Understanding: Enhancing ChatGPT's contextual understanding capabilities to provide coherent responses and maintain context during event planning conversations. Handling follow-up questions and maintaining a consistent user experience.
5. Application Development: Developing the server-side logic using Python and ChatGPT to handle user input, process event details, generate recommendations, and provide assistance throughout the event planning process.

Evaluation:
The evaluation will consist of the following components:

1. MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as NLP, event planning domain knowledge, user interface design, contextual understanding, and application development.

2. Live Evaluation: Industrial experts/Faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts/Faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 12: Legal Assistant using ChatGPT

Description:
The project aims to develop a Legal Assistant application using ChatGPT. The application will assist users in accessing legal information, understanding legal concepts, and providing basic legal guidance. Users will interact with the application through a user-friendly interface, and the ChatGPT model will generate responses based on legal knowledge and understanding.

Tasks:
1. Frontend Development: Design and implement the user interface for the Legal Assistant application. Use a frontend framework like Streamlit to create input forms for user queries and display legal information.

2. Backend Development: Develop the backend logic using Python and ChatGPT. Handle user requests, process legal queries, and utilize the ChatGPT model to generate legal responses.

3. Legal Knowledge Integration: Acquire and integrate legal knowledge into the application. Gather legal information from reliable sources and provide accurate responses based on legal concepts and processes.
4. Conversational User Interface: Design and implement a conversational flow that allows users to input their legal queries, receive legal information, and ask follow-up questions.

5. Contextual Understanding: Enhance the ChatGPT model's contextual understanding capabilities to maintain context and provide accurate legal responses during legal conversations.

6. Application Integration: Connect the legal assistant components to the frontend of the Legal Assistant application. Incorporate the model's responses and legal knowledge into the application's logic for processing user input and providing legal guidance.
Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. Natural Language Processing (NLP): Understanding the fundamentals of NLP and its applications in the legal domain. Familiarizing yourself with ChatGPT and its usage for generating legal information and responses.

2. Legal Domain Knowledge: Acquiring basic knowledge of legal concepts, terminology, and processes. Understanding common legal issues and providing general legal guidance.

3. User Interface Design: Designing and implementing a user-friendly interface for the Legal Assistant application. Creating input forms for user queries, legal research, and displaying legal information.

4. Contextual Understanding: Enhancing ChatGPT's contextual understanding capabilities to provide accurate legal information and maintain context during legal conversations. Handling follow-up questions and maintaining a consistent user experience.

5. Application Development: Developing the server-side logic using Python and ChatGPT to handle user input, process legal queries, and generate relevant legal responses.

Evaluation:
The evaluation will consist of the following components:

1. MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as NLP, legal domain knowledge, user interface design, contextual understanding, and application development.
2. Live Evaluation: Industrial experts/Faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts/Faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 13: AI Based Resume Builder using ChatGPT

Description:

The project aims to develop an AI-based Resume Builder application using ChatGPT. The application will assist users in creating professional resumes by providing suggestions, formatting guidance, and content recommendations. Users will interact with the application through a user-friendly frontend built using Streamlit, and the ChatGPT model will generate personalized resume content based on user input and preferences.

Tasks:
1. Frontend Development: Design and implement the user interface for the Resume Builder application using Streamlit. Create input forms for user information, resume sections, and display the generated resume content.

2. Backend Development: Develop the backend logic using Python and ChatGPT. Handle user requests, process resume information, and utilize the ChatGPT model to generate personalized resume content.

3. Resume Writing Guidelines Integration: Integrate resume writing guidelines and best practices into the application. Provide formatting suggestions, content recommendations, and tips for each resume section.

4. Conversational User Interface: Design and implement a conversational flow that allows users to input their resume information, select desired resume sections, and receive personalized content suggestions.

5. Contextual Understanding: Enhance the ChatGPT model's contextual understanding capabilities to maintain context and generate accurate and personalized resume content based on user preferences.

6. Application Integration: Connect the Resume Builder components to the frontend of the application. Incorporate the model's responses and resume writing guidelines into the application's logic for processing user input and generating customized resumes.

Learning Outcome:
Through this project, you will gain experience and understanding in the following areas:

1. Natural Language Processing (NLP): Understanding the fundamentals of NLP and its applications in resume building. Familiarizing yourself with ChatGPT and its usage for generating resume content and suggestions.
2. Resume Writing Knowledge: Acquiring knowledge of resume writing best practices, including structure, content, and formatting. Understanding different resume sections, such as education, experience, skills, and achievements.

3. User Interface Design: Designing and implementing a user-friendly interface for the Resume Builder application using Streamlit. Creating input forms for user information, resume sections, and displaying generated resume content.

4. Contextual Understanding: Enhancing ChatGPT's contextual understanding capabilities to provide accurate and personalized resume content based on user input and preferences.

5. Application Development: Developing the server-side logic using Python and ChatGPT
to handle user input, process resume information, and generate customized resume content.

Evaluation:
The evaluation will consist of the following components:

1. MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as NLP, resume writing knowledge, user interface design, contextual understanding, and application development.

2. Live Evaluation: Industrial experts/Faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts/Faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 14: Real-time Digital Fashion Designer using DALL-E API

Description:
The project aims to develop a real-time digital fashion designer application that utilizes the DALL-E API to generate unique and creative fashion designs based on user inputs. The application will

provide a user-friendly interface where users can specify their design preferences, such as style, color, patterns, and other design elements. The DALL-E API will then generate digital fashion designs that match the user's preferences in real-time.

Tasks:

1. Frontend Development: Design and implement the user interface for the digital fashion designer application using Streamlit. Create forms and input fields to capture user design preferences.

2. Backend Development: Develop the server-side logic using Python. Handle user requests, integrate with the DALL-E API, and process the API responses.

3. API Integration: Understand and utilize the DALL-E API to send requests and receive responses for generating digital fashion designs. Implement the necessary authentication and rate limiting mechanisms.

4. User Input Processing: Pre-process and validate user inputs to ensure they meet the required format and constraints. Handle input errors and provide appropriate feedback to the user.

5. Real-time Design Generation: Implement the logic to send user inputs to the DALL-E API and receive the generated fashion designs in real-time. Display the designs to the user for review and feedback.

6. Design Customization: Allow users to customize the generated fashion designs by providing options to modify design elements such as colors, patterns, and styles. Implement the logic to apply these modifications and update the design accordingly.

7. Deployment: Deploy the application to a web server or hosting platform to make it accessible to users over the internet.

Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. API Integration: Utilizing the DALL-E API to send requests and receive responses for generating digital fashion designs. Understanding the API documentation and handling API authentication and rate limits.

2. Frontend Development: Designing and implementing a user-friendly interface for the digital fashion designer application using a frontend framework like Streamlit. This includes creating forms and input fields for capturing user design preferences.

3. Backend Development: Implementing the server-side logic using Python. Handling user requests, integrating with the DALL-E API, and processing the API responses.

4. User Input Processing: Pre-processing and validating user inputs to ensure they meet the required format and constraints. Handling input errors and providing appropriate feedback to the user.

5. Real-time Design Generation: Implementing the logic to send user inputs to the DALL-E API and receive the generated fashion designs in real-time. Displaying the generated designs to the user for review and feedback.

6. Design Customization: Allowing users to customize the generated fashion designs by providing options to modify design elements such as colors, patterns, and styles. Implementing the logic to apply these modifications and update the design accordingly.

7. Deployment: Deploying the application to a web server or hosting platform to make it accessible to users over the internet.

Evaluation:

The evaluation will consist of the following components:

1. MCQ Questions: A set of 30 multiple-choice questions covering the technologies and concepts used in the project, such as API integration, frontend development, backend development, data processing, and deployment.

2. Live Evaluation: Industrial experts/Faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts/Faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use case 15: Image-to-Emoji Translator using DALL-E API

Description:

The project aims to develop an image-to-emoji translator using the DALL-E model. The application will allow users to upload an image, and the DALL-E model will generate an emoji representation of the image. The application will provide a user-friendly interface for image upload, processing, and displaying the generated emoji translation.

Tasks:

1. Frontend Development: Design and implement the user interface for the image-to-emoji translator application using Streamlit. Create an image upload functionality and display the generated emoji translation.

2. Backend Development: Develop the server-side logic using Python. Handle user requests, integrate with the DALL-E model, and process the generated emoji translations.

3. DALL-E Model Integration: Understand and integrate the DALL-E model into the application for image translation. Handle model input and output, and explore the capabilities of the model.

4. Image Processing: Pre-process and resize user-uploaded images to meet the requirements of the DALL-E model. Handle image formats, resizing, and other image-related operations.

5. Model Output Processing: Handle the output of the DALL-E model, which may be in the form of a tensor or an encoded representation. Convert the model output to a readable emoji format for display.

6. Deployment: Deploy the application to a web server or hosting platform to make it accessible to users over the internet.
Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. DALL-E Model Integration: Understanding the DALL-E model architecture and integrating it into the application for image translation. Exploring the model's capabilities and limitations.

2. Frontend Development: Designing and implementing a user interface for the image-to- emoji translator application using a frontend framework like Streamlit. Creating an image upload functionality and displaying the generated emoji translation.

3. Backend Development: Implementing the server-side logic using Python. Handling user requests, integrating with the DALL-E model, and processing the generated emoji translations.

4. Image Processing: Pre-processing and resizing user-uploaded images to meet the requirements of the DALL-E model. Handling image formats, resizing, and other image-related operations.

5. Model Output Processing: Handling the output of the DALL-E model, which may be in the form of a tensor or an encoded representation. Converting the model output to a readable emoji format for display.

6. Deployment: Deploying the application to a web server or hosting platform to make it accessible to users over the internet.

Evaluation:

The evaluation will consist of the following components:

1. MCQ Questions: A set of 30 multiple-choice questions covering the technologies and concepts used in the project, such as model integration, frontend development, backend development, image processing, and deployment.

2. Live Evaluation: Industrial experts/Faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts/Faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 16: Virtual Try-On for Accessories using DALL-E API Description:
The project aims to develop a virtual try-on application using the DALL-E model for accessories.

The application will allow users to upload a picture of themselves and virtually try on various accessories, such as glasses, hats, earrings, and necklaces. The DALL-E model will generate realistic images of the user wearing the selected accessories, providing a virtual try-on experience.

Tasks:

1. Frontend Development: Design and implement the user interface for the virtual try-on application using Streamlit. Create an image upload functionality and display the generated virtual try-on images.

2. Backend Development: Develop the server-side logic using Python. Handle user requests, integrate with the DALL-E model, and process the generated virtual try-on images.

3. DALL-E Model Integration: Understand and integrate the DALL-E model into the application for generating virtual try-on images. Handle model input and output, and explore the capabilities of the model.

4. Image Processing: Pre-process and resize user-uploaded images to meet the requirements of the DALL-E model. Handle image formats, resizing, and other image-related operations.

5. Accessory Selection: Provide a user-friendly interface for selecting and applying accessories to the uploaded image. Handle accessory options, user preferences, and apply the selected accessories to the image.

6. Model Output Processing: Handle the output of the DALL-E model, which may be in the form of a tensor or an encoded representation. Convert the model output to a displayable image format for the virtual try-on experience.

7. Deployment: Deploy the application to a web server or hosting platform to make it accessible to users over the internet.

Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. DALL-E Model Integration: Understanding the DALL-E model architecture and integrating it into the application for generating virtual try-on images. Exploring the model's capabilities and limitations.

2. Frontend Development: Designing and implementing a user interface for the virtual try-on application using a frontend framework like Streamlit. Creating an image upload functionality and displaying the generated virtual try-on images.

3. Backend Development: Implementing the server-side logic using Python. Handling user requests, integrating with the DALL-E model, and processing the generated virtual try-on images.

4. Image Processing: Pre-processing and resizing user-uploaded images to meet the requirements of the DALL-E model. Handling image formats, resizing, and other image-related operations.

5. Accessory Selection: Providing a user-friendly interface for selecting and applying accessories to the uploaded image. Handling accessory options, user preferences, and applying the selected accessories to the image.

6. Model Output Processing: Handling the output of the DALL-E model, which may be in the form of a tensor or an encoded representation. Converting the model output to a displayable image format for the virtual try-on experience.

7. Deployment: Deploying the application to a web server or hosting platform to make it accessible to users over the internet.

Evaluation:
The evaluation will consist of the following components:

1. MCQ Questions: A set of 30 multiple-choice questions covering the technologies and concepts used in the project, such as model integration, frontend development, backend development, image processing, and deployment.

2. Live Evaluation: Industrial experts/Faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts/Faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 17: Call Center Automation using Whisper API

Description:
The project aims to develop a Call Center Automation system using the Whisper API. The system will leverage the power of ChatGPT to automate call center operations and provide intelligent and natural language-based interactions with callers. By integrating the Whisper API, the system will enable seamless communication and assistance for customers, improving the efficiency and effectiveness of call center services.

Tasks:

1. Call Handling: Implement call handling functionality to receive incoming calls and interact with customers using the Whisper API.

2. Customer Support: Develop customer support features that allow customers to ask questions, seek assistance, and receive information about products, services, and common inquiries.
3. Information Retrieval: Utilize the Whisper API to retrieve relevant information and provide accurate responses to customer inquiries by making appropriate API calls.

4. Contextual Understanding: Enhance the Whisper API's contextual understanding capabilities to maintain context and provide coherent responses during customer interactions. Handle follow-up questions and maintain a consistent customer experience.

5. Backend Development: Develop the server-side logic using a backend framework like Flask or Express.js. Handle incoming calls, process customer queries, integrate the Whisper API, and utilize it for generating responses.

6. Application Integration: Connect the Whisper API and backend logic to the call center automation system. Incorporate the API's responses into the application's logic for processing customer inquiries and providing automated support.

Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. Natural Language Processing (NLP): Understanding the fundamentals of NLP and its applications in call center automation. Familiarizing yourself with the Whisper API and its capabilities for generating responses and understanding customer queries.

2. Call Center Operations: Acquiring knowledge of call center processes, including call handling, customer support, and information retrieval. Understanding common customer inquiries and challenges faced in call center environments.

3. Conversational AI Design: Designing and implementing a conversational AI system for call center automation. Creating a conversational flow that handles customer inquiries, provides support, and retrieves relevant information using the Whisper API.

4. Whisper API Integration: Integrating the Whisper API into the call center automation system to leverage its powerful language processing capabilities. Making API calls for generating responses and understanding customer queries.

5. Contextual Understanding: Enhancing the Whisper API's contextual understanding capabilities to provide coherent responses and maintain context during customer interactions. Handling complex queries and maintaining a consistent customer experience.

6. Application Development: Developing the server-side logic using a backend framework (e.g., Flask, Express.js) to handle incoming calls, process queries, integrate the Whisper API, and generate appropriate responses.

Evaluation:

The evaluation will consist of the following components:

1. MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as NLP, call center operations, conversational AI design, API integration, and application development.

2. Live Evaluation: Industrial experts/Faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts/Faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 18: Real-time Transcription Service for Online Meetings using Whisper API

Description:

The project aims to develop a real-time transcription service for online meetings using the Whisper API. The service will provide automated speech-to-text conversion during online meetings, allowing participants to have a written record of the conversation. By leveraging the powerful language processing capabilities of the Whisper API, the system will deliver accurate and real-time transcriptions to enhance communication and accessibility.

Tasks:

1. Online Meeting Integration: Integrate the real-time transcription service with popular online meeting platforms using their APIs. Capture and process audio streams from meetings.

2. Audio Streaming: Implement audio streaming functionality to handle real-time audio input from online meetings. Ensure low latency and efficient processing of audio data.

3. Whisper API Integration: Integrate the Whisper API into the real-time transcription service. Make API calls to perform speech-to-text conversion and receive transcriptions.

4. Text Processing: Implement text processing techniques to clean, format, and enhance the transcriptions. Handle punctuation, capitalization, and special characters for improved readability.

5. Backend Development: Develop the server-side logic using a backend framework like Flask or Express.js. Handle audio streaming, integrate the Whisper API, and provide real-time transcriptions.

6. Application Integration: Connect the backend logic to the online meeting platforms and the frontend of the real-time transcription service. Display the transcriptions in real-time during meetings.

Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. Speech Recognition: Understanding the fundamentals of speech recognition technology and its applications in real-time transcription. Familiarizing yourself with the Whisper API and its capabilities for converting speech to text.

2. Online Meeting Platforms: Acquiring knowledge of popular online meeting platforms (e.g., Zoom, Microsoft Teams, Google Meet) and their APIs for integrating the real-time transcription service.

3. Real-time Processing: Designing and implementing a real-time processing pipeline to capture and process audio from online meetings. Ensuring low latency and efficient transcription for a seamless user experience.

4. Whisper API Integration: Integrating the Whisper API into the real-time transcription service to perform speech-to-text conversion. Making API calls to process audio and receive transcriptions.

5. Text Processing: Implementing text processing techniques to clean, format, and enhance the transcriptions for readability and accuracy. Handling punctuation, capitalization, and special characters.

6. Application Development: Developing the server-side logic using a backend framework (e.g., Flask, Express.js) to handle audio streaming, integrate the Whisper API, and provide real-time transcriptions.

Evaluation:

The evaluation will consist of the following components:

1. MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as speech recognition, online meeting platforms, real-time processing, API integration, and application development.

2. Live Evaluation: Industrial experts/Faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts/Faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 19: Speech Analytics and Insights using Whisper API
Description:
The project aims to develop a system that utilizes the Whisper API for speech recognition to analyze and extract insights from recorded conversations. The system will perform various speech analytics tasks, such as sentiment analysis, keyword extraction, and customer behavior analysis, to provide valuable insights and actionable information from audio data.

Tasks:
1. Audio Processing: Implement audio processing techniques to handle recorded conversations. Perform audio segmentation, noise reduction, and speaker diarization to prepare the data for speech recognition and analysis.

2. Whisper API Integration: Integrate the Whisper API into the system to perform speech recognition and generate transcriptions. Make API calls to process audio recordings and receive text transcripts.

3. Sentiment Analysis: Implement a sentiment analysis algorithm to analyze the emotional tone of the conversations. Determine positive, negative, or neutral sentiments expressed by the speakers.

4. Keyword Extraction: Implement a keyword extraction algorithm to identify important terms and topics discussed in the conversations. Extract keywords and phrases that provide insights into the content.

5. Customer Behavior Analysis: Implement algorithms to analyze customer behavior in the conversations. Identify patterns, preferences, and trends to understand customer needs and behaviors.

6. Backend Development: Develop the server-side logic using a backend framework like Flask or Express.js. Handle audio processing, integrate the Whisper API, perform speech analytics tasks, and provide insights.

7. Application Integration: Connect the backend logic to the frontend of the system. Display the extracted insights and provide an interactive user interface for accessing and visualizing the analytics results.

Learning Outcome:
Through this project, you will gain experience and understanding in the following areas:
1. Speech Analytics: Understanding the fundamentals of speech analytics and its applications in extracting insights from audio data. Familiarizing yourself with the Whisper API and its capabilities for speech recognition and analysis.

2. Natural Language Processing (NLP): Acquiring knowledge of NLP techniques for sentiment analysis, keyword extraction, and customer behavior analysis. Understanding how to apply NLP algorithms to audio transcripts for extracting meaningful information.

3. Audio Processing: Learning audio processing techniques to handle recorded conversations, such as audio segmentation, noise reduction, and speaker diarization, to prepare the data for speech recognition and analysis.
4. Whisper API Integration: Integrating the Whisper API into the system to perform speech recognition and generate transcriptions. Making API calls to process audio recordings and receive text transcripts.

5. Insights Extraction: Implementing algorithms for sentiment analysis, keyword extraction, and customer behavior analysis on the transcriptions. Extracting meaningful insights and information from the audio data.

6. Application Development: Developing the server-side logic using a backend framework (e.g., Flask, Express.js) to handle audio processing, integrate the Whisper API, and provide speech analytics and insights.

Evaluation:
The evaluation will consist of the following components:

1. MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as speech analytics, NLP, audio processing, API integration, and application development.

2. Live Evaluation: Industrial experts/Faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts/Faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement.

Use Case 20: JD Based Assessment Generator

Description:

The project aims to develop a JD (Job Description) based assessment generator that utilizes natural language processing techniques to analyze job descriptions and generate assessment questions tailored to specific job roles. The system will analyze the key skills, qualifications, and responsibilities mentioned in the job description and generate relevant assessmen t questions to evaluate candidates' suitability for the role.

Tasks:

1. Job Description Analysis: Implement NLP techniques to analyze job descriptions. Extract key skills, qualifications, and responsibilities from the job descriptions to serve as the basis for assessment question generation.

2. Question Generation: Develop algorithms to generate assessment questions based on the analyzed job descriptions. Formulate relevant and appropriate questions that evaluate candidates' suitability for the specific job roles.

3. Data Integration: Integrate job description data sources into the system. Utilize existing job description databases or APIs to access a wide range of job descriptions for analysis and question generation.

4. Backend Development: Develop the server-side logic using a backend framework like Flask or Express.js. Handle job description analysis, question generation, and assessment generation. Provide APIs for accessing and generating assessments based on input job descriptions.

5. User Interface Development: Design and implement a user-friendly interface using a frontend framework like Streamlit or React. Create an intuitive user interface where users can input job descriptions, generate assessments, and review the generated questions.

6. Application Integration: Connect the backend logic with the frontend interface. Enable seamless communication between the user interface and the backend APIs for job description analysis and assessment generation.

Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. Natural Language Processing (NLP): Understanding the fundamentals of NLP and its applications in analyzing text data. Familiarizing yourself with techniques such as text classification, information extraction, and question generation.

2. Job Description Analysis: Acquiring knowledge of job description analysis and understanding how to extract key information from job descriptions, such as required skills, qualifications, and responsibilities.

3. Question Generation: Learning how to generate assessment questions based on the analysis of job descriptions. Understanding how to formulate relevant and appropriate questions that assess candidates' suitability for specific job roles.

4. Data Integration: Integrating and utilizing job description data sources to gather information and generate assessment questions. Leveraging existing job description databases or APIs to access a wide range of job descriptions for analysis.

5. Application Development: Developing the server-side logic using a backend framework (e.g., Flask, Express.js) to handle job description analysis, question generation, and assessment generation. Building an application that generates customized assessments based on input job descriptions.

6. User Interface Design: Designing and implementing a user-friendly interface for the JD-based assessment generator. Creating a user-friendly experience where users can input job descriptions, generate assessments, and review the generated questions.

Evaluation:
The evaluation will consist of the following components:

1. MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as NLP, job description analysis, question generation, data integration, and application development.

2. Live Evaluation: Industrial experts/Faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts/Faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement.

Conclusion:
We hereby thank you for inviting GUVI Geek Network Pvt Ltd., for the discussion and we assure that once we are officially engaged, we would execute the proposed plan beyond your expectation and deliver more value. Feel free to call or email me for any clarifications.

		L	T	P	C
SB8049	**FULL STACK PYTHON PROGRAMMING WITH MACHINE LEARNING**	**1**	**0**	**2**	**2**

Module 1: Web Development Fundamentals **9**

Introduction to HTML Tags - Structuring web pages using HTML – HTML5 - Introduction to CSS - CSS selectors - CSS properties and values - Styling HTML elements using CSS – Bootstrap - Using Bootstrap's grid system for responsive web design - Customizing Bootstrap components and styles - DOM Manipulation - Understanding the Document Object Model (DOM) - Accessing HTML elements using JavaScript - Manipulating and modifying HTML elements dynamically – Introduction to version control system GIT and GitHub – How to use GIT and GitHub – Differences of GIT and GitHub

Module 2: JavaScript and ReactJS **9**

JavaScript Basics - Variables, data types, and operators - Conditional statements and loops - Functions and scope - **Introduction to ReactJS -** React components and their lifecycle - JSX syntax and rendering components - Handling events in React - **State and Props -** Managing component state in React - Passing data between components using props - Handling props validation - **React Router -** Setting up routing in React applications - Navigating between different routes - Passing route parameters and accessing them in components

Module 3: Databases and Python FLASK Framework **9**

Introduction to Databases - Types of databases : Relational databases (e.g., MySQL, PostgreSQL) - NoSQL databases (e.g., MongoDB) - Structured Query Language (SQL) - Performing CRUD operations on a database - Introduction to Python FLASK Framework - Setting up a Flask project and creating routes - Using templates for dynamic web page generation - Data can be passed from Python to templates for rendering - Handling form submissions and user input - CRUD Operations in the database using Flask

Module 4: Machine Learning Introduction 9

Introduction to Machine Learning - Definition, types (supervised, unsupervised, semi-supervised, reinforcement), and applications - Steps involved in a typical machine learning workflow - Overfitting, underfitting, and model selection - Supervised Learning - Regression algorithms, Classification algorithms, Model evaluation metrics - Evaluation and Validation - Training and test data splitting - Cross-validation techniques - Evaluation metrics for regression and classification models

Module 5: Machine Learning Implementation 9

Data Preprocessing - Handling missing data - Feature scaling techniques - Encoding categorical variables - Model Training and Evaluation - Splitting data into training and testing sets - Training machine learning models using scikit-learn or other libraries - Evaluating model performance using appropriate metrics - Feature Selection and Engineering - Techniques for selecting relevant features - Creating new features from existing data - Feature importance analysis and visualization - Model Deployment - Exporting trained models for deployment - Creating a Flask API to serve predictions - Deploying the machine learning model to a production environment

TOTAL :45 PERIODS

REFERENCE BOOKS:

1. Learning Web Design: A Beginner`S Guide To HTML, CSS, JavaScript, & Web Graphics 5th Edition
2. Fullstack React: The Complete Guide to ReactJS and Friends by Anthony Accomazzo
3. Flask Web Development, 2nd Edition by Miguel Grinberg
4. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by Aurélien Géron
5. Building Machine Learning Powered Applications: Going from Idea to Product by Emmanuel Ameisen

6 Sample 20 Use cases

Use Case 1: Loan Eligibility Prediction Application

Description:

The project aims to develop a loan eligibility prediction application that utilizes machine learning algorithms to assess the eligibility of individuals for obtaining loans. The application will provide a user-friendly interface for inputting customer data and generate real-time loan eligibility predictions based on the trained model.

Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. Frontend Development: Designing and implementing the user interface for the loan eligibility prediction application using HTML, CSS, and JavaScript. This includes creating user-friendly forms for data input and result display.

2. Backend Development: Building the server-side logic using a backend framework such as Flask (Python) or Express.js (JavaScript). This involves handling user requests, data processing, and model integration.

3. Data Pre-processing: Cleaning and pre-processing the loan dataset, handling missing values, outliers, and data inconsistencies. Applying techniques such as imputation, scaling, and encoding categorical variables.

4. Feature Engineering: Identifying and creating relevant features from the loan dataset that can improve the loan eligibility prediction model's performance. This may involve feature extraction, transformation, or combination.

5. Machine Learning Model Building and Evaluation: Selecting a suitable machine learning algorithm (e.g., logistic regression, decision trees, random forests) for loan eligibility prediction. Training the model using the pre-processed data and evaluating its performance using appropriate metrics.

6. Integration with Web Application: Connecting the trained machine learning model with the backend of the web application to enable real-time loan eligibility predictions. Incorporating the model into the application's logic for processing user input and providing predictions**.**

Tasks:

1. Frontend Development: Design and implement the user interface for the loan eligibility prediction application, including data input forms and result display components.

2. Backend Development: Develop the server-side logic using a backend framework like Flask or Express.js. Handle user requests, data processing, and integrate the machine learning model into the application.

3. Data Pre-processing: Clean, transform, and pre-process the loan dataset, handling missing values, outliers, and data inconsistencies. Apply techniques like imputation, scaling, and encoding categorical variables.

4. Feature Engineering: Identify and create relevant features from the loan dataset to enhance the loan eligibility prediction model's performance. Perform feature extraction, transformation, or combination as required.

5. Machine Learning Model Building and Evaluation: Select a suitable machine learning algorithm for loan eligibility prediction. Train the model using the pre-processed data and evaluate its performance using appropriate metrics.

6. Integration with Web Application: Connect the trained machine learning model to the backend of the web application, enabling real-time loan eligibility predictions based on user input.

Evaluation:

The evaluation will consist of the following components:

1. MCQ Questions: A set of 30 multiple-choice questions covering the technologies and concepts used in the project, such as frontend development, backend development, database management, data pre-processing, ML model building, and deployment.

2. Live Evaluation: Industrial experts/Faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts/Faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement**.**

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 2: Diabetes prediction Application

Description:

The project aims to develop a diabetes prediction application that utilizes machine learning algorithms to predict the likelihood of an individual developing diabetes based on certain risk factors. The application will provide users with a user-friendly interface to input their health information and generate real-time predictions regarding their risk of developing diabetes.

Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. Frontend Development: Designing and implementing the user interface for the diabetes prediction application using HTML, CSS, and JavaScript. This includes creating user-friendly forms for data input and result display.

2. Backend Development: Building the server-side logic using a backend framework such as Flask (Python) or Express.js (JavaScript). This involves handling user requests, data processing, and model integration.

3. Data Pre-processing: Cleaning and pre-processing the diabetes dataset, handling missing values, outliers, and data inconsistencies. Applying techniques such as imputation, scaling, and encoding categorical variables.

4. Feature Engineering: Identifying and creating relevant features from the diabetes dataset that can improve the prediction model's performance. This may involve feature extraction, transformation, or combination.

5. Machine Learning Model Building and Evaluation: Selecting a suitable machine learning algorithm (e.g., logistic regression, decision trees, random forests, or support vector machines) for diabetes prediction. Training the model using the pre-processed data and evaluating its performance using appropriate metrics.

6. Integration with Web Application: Connecting the trained machine learning model with the backend of the web application to enable real-time diabetes predictions. Incorporating the model into the application's logic for processing user input and providing predictions.

Tasks:

1. Frontend Development: Design and implement the user interface for the diabetes prediction application, including data input forms and result display components.

2. Backend Development: Develop the server-side logic using a backend framework like Flask or Express.js. Handle user requests, data processing, and integrate the machine learning model into the application.

3. Data Pre-processing: Clean, transform, and pre-process the diabetes dataset, handling missing values, outliers, and data inconsistencies. Apply techniques like imputation, scaling, and encoding categorical variables.

4. Feature Engineering: Identify and create relevant features from the diabetes dataset to enhance the prediction model's performance. Perform feature extraction, transformation, or combination as required.

5. Machine Learning Model Building and Evaluation: Select a suitable machine learning algorithm for diabetes prediction. Train the model using the pre-processed data and evaluate its performance using appropriate metrics.

6. Integration with Web Application: Connect the trained machine learning model to the backend of the web application, enabling real-time diabetes predictions based on user input.

Evaluation:

The evaluation will consist of the following components:

1. MCQ Questions: A set of 30 multiple-choice questions covering the technologies and concepts used in the project, such as frontend development, backend development, database management, data pre-processing, ML model building, and deployment.

2. Live Evaluation: Industrial experts will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 3: Glass Classification Application

Description:

The project aims to develop a glass classification application using machine learning algorithms to predict the type of glass based on its chemical composition. The application will provide users with a user-friendly interface to input the chemical attributes of glass samples and generate real-time predictions regarding the glass type.

Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. Frontend Development: Designing and implementing the user interface for the glass classification application using HTML, CSS, and JavaScript. This includes creating user-friendly forms for data input and result display**.**

2. Backend Development: Building the server-side logic using a backend framework such as Flask (Python) or Express.js (JavaScript). This involves handling user requests, data processing, and model integration.

3. Data Pre-processing: Cleaning and pre-processing the glass dataset, handling missing values, outliers, and data inconsistencies. Applying techniques such as imputation, scaling, and encoding categorical variables if required.

4. Feature Engineering: Identifying and selecting relevant features from the glass dataset that can improve the classification model's performance. This may involve feature extraction, transformation, or combination.

5. Machine Learning Model Building and Evaluation: Selecting a suitable machine learning algorithm (e.g., decision trees, random forests, support vector machines) for glass classification. Training the model using the pre-processed data and evaluating its performance using appropriate metrics.

6. Integration with Web Application: Connecting the trained machine learning model with the backend of the web application to enable real-time glass classification. Incorporating the model into the application's logic for processing user input and providing predictions.

Tasks:

1. Frontend Development: Design and implement the user interface for the glass classification application, including data input forms and result display components**.**

2. Backend Development: Develop the server-side logic using a backend framework like Flask or Express.js. Handle user requests, data processing, and integrate the machine learning model into the application.

3. Data Pre-processing: Clean, transform, and pre-process the glass dataset, handling missing values, outliers, and data inconsistencies. Apply techniques like imputation, scaling, and encoding categorical variables if required.

4. Feature Engineering: Identify and select relevant features from the glass dataset to enhance the classification model's performance. Perform feature extraction, transformation, or combination as required.

5. Machine Learning Model Building and Evaluation: Select a suitable machine learning algorithm for glass classification. Train the model using the pre-processed data and evaluate its performance using appropriate metrics.

6. Integration with Web Application: Connect the trained machine learning model to the backend of the web application, enabling real-time glass classification based on user input.

Evaluation:

The evaluation will consist of the following components:

1. MCQ Questions: A set of 30 multiple-choice questions covering the technologies and concepts used in the project, such as frontend development, backend development, database management, data pre-processing, ML model building, and deployment.

2. Live Evaluation: Industrial experts will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 4: PhonePe Pulse Data Analysis

Description:

The project aims to perform data analysis on PhonePe Pulse data, a fictional mobile payment service, to gain insights and extract valuable information. The analysis will involve exploring the dataset, performing statistical calculations, and generating visualizations to understand user behaviour, transaction patterns, and other relevant metrics.

Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. Data Exploration: Analysing the PhonePe Pulse dataset to understand its structure, variables, and data types. Performing data cleaning and handling missing values or outliers as necessary.

2. Data Wrangling: Preparing the dataset for analysis by transforming, reshaping, and aggregating data. This may involve merging multiple datasets, creating new variables, or filtering data based on specific criteria.

3. Statistical Analysis: Applying statistical techniques to derive meaningful insights from the data. This may include calculating descriptive statistics, conducting hypothesis tests, and identifying correlations or relationships between variables.

4. Data Visualization: Creating visual representations of the data using charts, graphs, and plots to effectively communicate patterns and trends. This may involve using libraries such as Matplotlib, Seaborn, or Plotly in Python.

5. Exploratory Data Analysis: Conducting exploratory data analysis techniques to uncover patterns, outliers, and anomalies in the PhonePe Pulse data. This may involve segmentation, clustering, or anomaly detection techniques.

6. Insights and Recommendations: Drawing meaningful insights from the data analysis and providing recommendations based on the findings. This may involve identifying opportunities for improvement, optimizing processes, or enhancing user experience.

Tasks:

1. Data Exploration: Explore the PhonePe Pulse dataset, examine its structure, and identify relevant variables for analysis.

2. Data Cleaning: Handle missing values, outliers, and inconsistencies in the dataset, ensuring data integrity and quality.

3. Data Wrangling: Transform and reshape the data as needed, merge multiple datasets if available, and create new variables for analysis.

4. Statistical Analysis: Perform statistical calculations, including descriptive statistics, hypothesis tests, and correlations, to gain insights into the data**.**

5. Data Visualization: Create visualizations using appropriate charts, graphs, and plots to present the findings effectively.

6. Exploratory Data Analysis: Apply exploratory data analysis techniques to uncover patterns, clusters, or anomalies in the PhonePe Pusle data**.**

7. Insights and Recommendations: Derive meaningful insights from the analysis results and provide recommendations based on the findings.

Evaluation:

The evaluation will consist of the following components:

1. MCQ Questions: A set of multiple-choice questions covering the concepts and techniques related to PhonePe Pulse, data preprocessing, exploratory data analysis, statistical modelling, visualisation, and interpretation of results.

2. Live Evaluation: Industrial experts will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 5: Breast Cancer Classification Application

Description:

The project aims to develop a breast cancer prediction application that incorporates both frontend and backend components. The application will utilize machine learning algorithms to classify breast tissue as malignant (cancerous) or benign (non-cancerous) based on various features. It will provide users with a user-friendly interface to input the relevant features and generate real-time predictions regarding the likelihood of breast cancer.

Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. **Frontend Development:** Designing and implementing the user interface for the breast cancer prediction application using HTML, CSS, and JavaScript.
2. **Backend Development:** Building the server-side logic using a backend framework such as Flask
(Python) or Express.js (JavaScript).
3. **Data Pre-processing:** Cleaning and pre-processing the breast cancer dataset, handling missing values, outliers, and data inconsistencies.
4. **Machine Learning Model Building and Evaluation:** Selecting and implementing appropriate machine learning algorithms for breast cancer prediction, such as logistic regression, support vector machines, or random forests. Training the model using the pre-processed data and evaluating its performance using appropriate metrics. Tuning the hyperparameters of the model to optimize its performance and ensure robust predictions. Validating the model using techniques like cross-validation to assess its generalizability.
5. **Integration:** Connecting the trained machine learning model to the backend of the web application, enabling real-time breast cancer predictions. Incorporating the model into the application's logic for processing user data and providing predictions. Ensuring smooth integration between the frontend and backend components for a seamless user experience.

Tasks:

1. Frontend Development: Design and implement the user interface for the breast cancer prediction application, including intuitive data input forms and result display components.

2. Backend Development: Develop the server-side logic using a backend framework like Flask or Express.js, handling user requests, data processing, and integrating the machine learning model.

3. Data Pre-processing: Clean, transform, and pre-process the breast cancer dataset, handling missing values, outliers, and data inconsistencies.

4. Machine Learning Model Building and Evaluation: Select and implement suitable machine learning algorithms for breast cancer prediction. Train the model using the pre-processed data and evaluate its performance using appropriate metrics.

5. Integration: Connect the trained machine learning model to the backend of the web application, enabling real-time breast cancer predictions based on user input.

Evaluation:
The evaluation of the project will consist of the following components:

1. MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, including frontend development, backend development, data pre-processing, machine learning algorithms, and integration.

2. Live Evaluation: Industrial experts will conduct a live evaluation session to assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to breast cancer prediction.

3. Feedback: The industrial experts will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 6: Flight Ticket Price prediction Application

Description:

The project aims to develop a flight ticket price prediction application using machine learning algorithms to forecast the prices of airline tickets based on various factors such as departure city, destination, travel dates, airline, and other relevant parameters. The application will provide users with real-time predictions to help them make informed decisions when booking flights.

Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. Data Collection and Exploration: Collecting flight ticket data from reliable sources and exploring the dataset to understand its structure, variables, and data types.

2. Data Cleaning and Pre-processing: Cleaning and reprocessing the flight ticket dataset, handling missing values, outliers, and data inconsistencies. Applying techniques such as imputation, feature scaling, and encoding categorical variables if required.

3. **Feature Engineering:** Extracting and creating new features from the existing dataset that may have an impact on flight ticket prices. This may include feature transformations, aggregations, or the creation of derived variables.

4. **Machine Learning Model Building and Evaluation:** Selecting and implementing appropriate machine learning algorithms (e.g., regression, ensemble methods) for flight ticket price prediction. Training the model using the pre-processed data and evaluating its performance using appropriate metrics.

5. **Model Optimization and Validation:** Tuning the hyperparameters of the machine learning models to improve their performance. Employing techniques like cross-validation and model validation to ensure robustness and generalizability.

6. **Integration with Web Application:** Connecting the trained machine learning model with the backend of the web application to enable real-time flight ticket price predictions based on user input. Incorporating the model into the application's logic for processing user data and providing predictions.

Tasks:

1. **Data Collection and Exploration:** Collect flight ticket data from reliable sources and explore the dataset to understand its structure and variables.

2. **Data Cleaning and Pre-processing:** Clean, transform, and pre-process the flight ticket dataset, handling missing values, outliers, and data inconsistencies.

3. **Feature Engineering:** Extract and create new features from the dataset that may have an impact on flight ticket prices.

4. **Machine Learning Model Building and Evaluation:** Select and implement suitable machine learning algorithms for flight ticket price prediction. Train the model using the pre-processed data and evaluate its performance using appropriate metrics.

5. **Model Optimization and Validation:** Tune the hyperparameters of the machine learning models to improve their performance. Validate the model using techniques like cross-validation.

6. **Integration with Web Application:** Connect the trained machine learning model to the backend of the web application, enabling real-time flight ticket price predictions based on user input.

Evaluation:

The evaluation of the project will consist of the following components:

1. **MCQ Questions:** A set of multiple-choice questions covering the technologies and concepts used in the project, including frontend development, backend development, data pre-processing, machine learning algorithms, and integration.

2. **Live Evaluation:** Industrial experts will conduct a live evaluation session to assess your

understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to breast cancer prediction.

3. Feedback: The industrial experts will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 7: Loan Risk Assessment Application

Description:

The project aims to develop a loan default classification application using machine learning algorithms to predict the likelihood of loan default for borrowers based on various factors such as credit score, income, employment history, loan amount, and other relevant features. The application will provide users with real- time predictions to assist lenders in assessing the creditworthiness of loan applicants and making informed decisions.

Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. Data Pre-processing: Cleaning and pre-processing the loan dataset, handling missing values, outliers, and data inconsistencies. Applying techniques such as imputation, feature scaling, and encoding categorical variables if required.

2. Feature Selection and Engineering: Selecting and creating relevant features that have a significant impact on loan default prediction. This may involve techniques like correlation analysis, feature importance ranking, or domain knowledge-based feature engineering.

3. Machine Learning Model Building and Evaluation: Selecting and implementing appropriate machine learning algorithms (e.g., logistic regression, decision trees, random forests) for loan default classification. Training the model using the pre-processed data and evaluating its performance using appropriate metrics such as accuracy, precision, recall, and F1-score.

4. Model Optimization and Validation: Tuning the hyperparameters of the machine learning models to improve their performance. Employing techniques like cross-validation and model validation to ensure robustness and generalizability.

5. Integration with Web Application: Connecting the trained machine learning model with the backend of the web application to enable real-time loan default predictions based on user input. Incorporating the model into the application's logic for processing user data and providing predictions.

Tasks:

1. Data Pre-processing: Clean, transform, and pre-process the loan dataset, handling missing values, outliers, and data inconsistencies.

2. Feature Selection and Engineering: Select and create relevant features that have a significant impact on loan default prediction.

3. Machine Learning Model Building and Evaluation: Select and implement suitable machine learning algorithms for loan default classification. Train the model using the pre-processed data and evaluate its performance using appropriate metrics.

4. Model Optimization and Validation: Tune the hyperparameters of the machine learning models to improve their performance. Validate the model using techniques like cross-validation.

5. Integration with Web Application: Connect the trained machine learning model to the backend of the web application, enabling real-time loan default predictions based on user input.

Evaluation:

The evaluation of the project will consist of the following components:

1. MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, including frontend development, backend development, data pre-processing, machine learning algorithms, and integration.

2. Live Evaluation: Industrial experts will conduct a live evaluation session to assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to breast cancer prediction.

3. Feedback: The industrial experts will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 8: Stock Price Prediction Application

Description:

The project aims to develop a stock price prediction application for Amazon, Microsoft, Google, and Apple using regression models. The application will utilize historical stock price data, along with other relevant factors such as market trends, news sentiment, and financial indicators, to predict future stock prices. Users will be able to access the predictions, visualize historical trends, and make informed investment decisions based on the provided insights.

Learning Outcome:

Through this project, you will gain experience and understanding in the following areas:

1. Data Collection: Gathering historical stock price data for Amazon, Microsoft, Google, and Apple from reliable financial sources or APIs. Collecting additional relevant data, such as market trends and financial indicators, to enhance the prediction models.

2. Feature Engineering: Selecting and creating appropriate features from the collected data to improve the prediction accuracy. This may involve calculating technical indicators, incorporating sentiment analysis from news articles, or considering macroeconomic factors.

3. Regression Model Building: Building regression models (e.g., linear regression, polynomial regression, support vector regression) to predict the future stock prices based on historical and additional feature data. Experimenting with different models and techniques to find the best performing model.

4. Model Training and Evaluation: Splitting the data into training and testing sets. Training the regression models using the training data and evaluating their performance using appropriate metrics such as mean squared error (MSE), mean absolute error (MAE), and R-squared**.**

5. Frontend Development: Designing and implementing the user interface for the stock price prediction application using HTML, CSS, and JavaScript. Creating interactive visualizations to display historical stock prices, predicted prices, and other relevant information.

6. Backend Development: Developing the server-side logic using a backend framework like Flask or Express.js. Handling user requests, processing data, and integrating the regression models for real-time predictions.

Tasks:

1. Data Collection: Gather historical stock price data for Amazon, Microsoft, Google, and Apple. Collect additional relevant data, such as market trends and financial indicators.

2. Feature Engineering: Select and create appropriate features from the collected data to enhance the prediction models.

3. Regression Model Building: Build regression models to predict future stock prices based on historical and additional feature data.

4. Model Training and Evaluation: Train the regression models using the training data and evaluate their performance using appropriate metrics.

5. Frontend Development: Design and implement the user interface for the stock price prediction application, including interactive visualizations.

6. Backend Development: Develop the server-side logic, handling user requests, processing data, and integrating the regression models for real-time predictions.

Evaluation:
The evaluation of the project will consist of the following components:

1. MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, including frontend development, backend development, data pre-processing, machine learning algorithms, and integration.

2. Live Evaluation: Industrial experts will conduct a live evaluation session to assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to breast cancer prediction.

3. Feedback: The industrial experts will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 9: Medical Insurance Premium Prediction

Description:

The Medical Insurance Premium Prediction project aims to develop a machine learning model that can accurately predict the insurance premium for an individual based on various factors such as age, gender, BMI, smoking habits, and region. The project will involve data collection, pre-processing, model training, evaluation, and deployment to create a web application that can provide users with estimated insurance premium quotes.

Learning Outcome:

By working on this project, participants will gain expertise in the following areas:

1. Data pre-processing techniques for handling missing values, outliers, and categorical variables.
2. Building regression models using machine learning algorithms.
3. Feature selection and engineering to enhance model performance.
4. Evaluating regression models using appropriate metrics.
5. Developing a web application for user interaction and displaying predicted premium quotes.
6. Deploying the web application on a server for accessibility.

Tasks:

1. Data Collection and Pre-processing:

- Gathering a comprehensive dataset containing information on individuals' age, gender, BMI, smoking habits, and region.
- Handling missing values, outliers, and categorical variables.
- Performing feature scaling or normalization to ensure consistent ranges across variables.

2. Exploratory Data Analysis (EDA):

- Conducting statistical analysis and visualizations to gain insights into the dataset.
- Identifying correlations between variables and their impact on insurance premiums.

3. Model Selection and Training:

- Selecting appropriate regression algorithms such as linear regression, decision trees, or random forests.
- Splitting the dataset into training and testing sets.
- Training the regression models using the training data.

- Tuning hyperparameters to optimize model performance.

4. Model Evaluation:

- Evaluating the trained regression models using metrics like mean absolute error (MAE), mean squared error (MSE), or R-squared.
- Comparing the performance of different models to select the most accurate one.

5. Web Application Development:

- Designing and implementing a user-friendly web application for the insurance premium prediction.
- Creating a form to input user data such as age, gender, BMI, smoking habits, and region.
- Connecting the front-end with the back-end for data processing and prediction.

6. Deployment:

- Setting up a web server to host the insurance premium prediction application.
- Deploying the web application and ensuring its availability and accessibility.

Evaluation:

The evaluation will consist of the following components:

1. MCQ Questions: A set of 30 multiple-choice questions covering the technologies and concepts used in the project, such as frontend development, backend development, database management, data pre-processing, ML model building, and deployment.

2. Live Evaluation: Industrial experts will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Tasks:

1. Data Collection and Exploration: Gather and explore customer data, including demographics, transaction history, and customer interactions.

2. Data Pre-processing and Feature Engineering: Clean the data, handle missing values, and perform feature engineering to extract relevant features for churn prediction.

3. Feature Selection: Identify the most important features for churn prediction.

4. Classification Model Building: Build classification models (e.g., logistic regression, decision tree, random forest) to predict customer churn.

5. Model Training and Evaluation: Split the data, train the classification models, and evaluate their performance using appropriate metrics.

6. Frontend Development: Design and implement a user interface for the churn prediction system**.**

7. Backend Development: Develop the server-side logic to handle user requests and provide real-time churn predictions.

Evaluation:

The evaluation will consist of the following components:

1. MCQ Questions: A set of 30 multiple-choice questions covering the technologies and concepts used in the project, such as frontend development, backend development, database management, data pre-processing, ML model building, and deployment**.**

2. Live Evaluation: Industrial experts will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. Feedback: The industrial experts will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This evaluation aims to assess your proficiency in the covered technologies and your ability to apply them to real-world projects, as well as to provide valuable feedback from industry experts to further enhance your skills.

Use Case 11: Mobile Price Classification System

Description:

The project focuses on developing a mobile price classification system using machine learning techniques. The system aims to predict the price range of mobile phones based on various features and specifications. By analysing the dataset of mobile phones with labelled price ranges, the model will learn patterns and correlations to accurately classify the price range of new mobile phones. The project involves data pre- processing, feature selection, model building and evaluation, and the development of a user-friendly interface for accessing the price classification system.

Learning Outcome:

By working on this project, you will gain the following learning outcomes:

1. **Understanding Mobile Price Classification:** Familiarize yourself with the concept of price classification and its significance in the mobile phone market. Understand the factors that influence mobile phone pricing and their impact on different price ranges

2. **Data Pre-processing:** Learn techniques for data cleaning, handling missing values, feature scaling, and data transformation to prepare the dataset for model training.

3. **Feature Selection**: Gain knowledge of feature selection methods to identify the most relevant features that significantly contribute to the price classification.

4. **Machine Learning Model Building and Evaluation:** Develop skills in selecting appropriate machine learning algorithms (such as decision trees, random forests, or support vector machines) for price classification. Train and evaluate the models using performance metrics like accuracy, precision, recall, and F1-score.

5. **Model Interpretation:** Understand how to interpret the trained model to identify the key features and their importance in predicting the price range of mobile phones.

6. **User Interface Development:** Design and develop a user-friendly interface for the mobile price classification system. Implement features to input the mobile phone specifications and display the predicted price range.

7. **Deployment:** Deploy the mobile price classification system on a suitable platform, making it accessible for users to input mobile phone specifications and obtain the predicted price range.

Tasks:

The project tasks should be executed in the following order:

1. **Data Collection**: Collect a dataset of mobile phones with labelled price ranges. The dataset should include various features such as brand, display size, RAM, internal storage, camera quality, battery capacity, etc.

2. **Data Pre-processing**: Clean the dataset by handling missing values, removing outliers, and transforming categorical variables into numerical representations.

3. **Feature Selection**: Select the most relevant features that contribute significantly to the price classification. Consider techniques such as correlation analysis, feature importance ranking, or dimensionality reduction methods.

4. **Machine Learning Model Building:** Train and evaluate different machine learning models using the pre-processed dataset. Experiment with various algorithms and tune their hyperparameters to achieve the best performance.

5. **Model Evaluation**: Evaluate the trained models using appropriate evaluation metrics, such as accuracy, precision, recall, and F1-score. Compare the performance of different models and select the best-performing one.

6. **User Interface Development:** Design and develop a user-friendly interface for the mobile price classification system. Implement input fields for mobile phone specifications and integrate the trained model to predict the price range based on the provided specifications.

7. **Deployment**: Deploy the mobile price classification system on a suitable platform (e.g., web application, mobile app) to make it accessible for users. Ensure the system is robust, secure, and provides accurate price range predictions.

Evaluation:

The evaluation will consist of the following components:

1. **MCQ Questions**: A set of 30 multiple-choice questions covering the technologies and concepts used in the project, such as frontend development, backend development, database management, data pre-processing, ML model building, and deployment.
2. **Live Evaluation:** Industrial experts or faculty will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.
3. **Feedback**: The industrial experts or faculty will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This examination aims to assess your knowledge of the subjects presented and your ability to apply it to actual projects, as well as to offer insightful criticism from professionals in the field to help you develop your abilities.

Use Case 12: Indian Real Estate Price Prediction

Description:

The project's goal is to create a system for predicting house prices for residential properties in India. The project focuses on developing a machine learning model that can precisely estimate house prices based on several traits and parameters using a dataset specifically designed for the Indian housing industry. Data pre-processing, feature engineering, model training and evaluation, as well as the creation of a user-friendly interface to access and interact with the price prediction system, will all be part of the system.

Learning Outcome:

By working on this project, you will gain the following learning outcomes:

1. **Understanding Indian Housing Market**: Familiarise yourself with the dynamics and characteristics of the Indian housing market, including factors that influence house prices such as location, size, amenities, neighbourhood, and market trends.
2. **Data Pre-processing**: Develop skills in data pre-processing techniques such as handling missing values, feature scaling, outlier detection and removal, and encoding categorical variables to ensure data quality and compatibility with machine learning algorithms.
3. **Feature Engineering**: Learn techniques to extract meaningful features from the dataset and create new features that can better capture the underlying patterns and relationships in the housing market.
4. **Machine Learning Model Building and Evaluation**: Explore various regression algorithms such as linear regression, decision trees, random forests, or gradient boosting to build a house price prediction model. Train and evaluate the models using appropriate evaluation metrics such as mean squared error (MSE) or root mean squared error (RMSE).

5. **Model Interpretation**: Understand how to interpret the trained model to identify the key features and their impact on house prices. Analyse feature importance and coefficients to gain insights into the factors driving housing prices in India.

6. **User Interface Development**: Design and develop a user-friendly interface that allows users to input relevant features such as location, size, amenities, and other factors to obtain an estimated price for a residential property in India.

7. **Deployment**: Configure and deploy the house price prediction system on a production environment, making it accessible to users for real-time price estimation of residential properties in India.

Tasks:

The project tasks should be executed in the following order:

1. **Data Collection**: Collect a comprehensive dataset specific to the Indian housing market, including features such as location, size, number of rooms, amenities, proximity to amenities, and historical sales data.

2. **Data Pre-processing:** Clean the dataset by handling missing values, performing feature scaling, outlier detection and removal, and encoding categorical variables.

3. **Feature Engineering:** Analyse the dataset and extract meaningful features that can capture the variations and patterns in the Indian housing market. Create new features if necessary, such as price per square foot or distance to important landmarks.

4. **Machine Learning Model Building**: Select suitable regression algorithms and train multiple models using the pre-processed dataset. Experiment with different algorithms, hyperparameters, and ensemble methods to find the best-performing model.

5. **Model Evaluation**: Evaluate the trained models using appropriate evaluation metrics such as mean squared error (MSE) or root mean squared error (RMSE). Compare the performance of different models and select the model with the lowest error.

6. **User Interface Development**: Design and develop a user-friendly interface that allows users to input relevant features of a residential property in India. Integrate the trained model into the interface to provide estimated house prices based on the user's inputs.

7. **Deployment**: Configure and deploy the house price prediction system on a production environment, ensuring scalability, reliability, and security. Make the system accessible to users for real-time price estimation of residential properties in India.

Evaluation:

The evaluation will consist of the following components:

1. **MCQ Questions:** A set of 30 multiple-choice questions covering the technologies and concepts used in the project, such as frontend development, backend development, database management, data pre-processing, ML model building, and deployment.

2. **Live Evaluation**: Industrial experts or faculties will conduct a live evaluation session where

they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. **Feedback**: The industrial experts or faculties will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This examination aims to assess your knowledge of the subjects presented and your ability to apply it to actual projects, as well as to offer insightful criticism from professionals in the field to help you develop your abilities.

Use Case 13: Airbnb Price Prediction in European Cities

Description:

The project's goal is to create a mechanism for estimating prices for Airbnb listings in different European locations. The project's goal is to develop a machine learning model that can precisely estimate the costs of Airbnb rooms based on numerous variables and parameters by using a dataset that is particular to European Airbnb listings. The system entails data pre-processing, feature engineering, model training and evaluation, as well as the creation of an intuitive user interface for accessing and interacting with the price prediction system.

Learning Outcome:

By working on this project, you will gain the following learning outcomes:

1. **Understanding European Airbnb Market**: Familiarise yourself with the dynamics and characteristics of the European Airbnb market, including factors that influence accommodation prices such as location, property type, amenities, availability, and seasonality.
2. **Data Pre-processing:** Develop skills in data pre-processing techniques such as handling missing values, feature scaling, outlier detection and removal, and encoding categorical variables to ensure data quality and compatibility with machine learning algorithms.
3. **Feature Engineering**: Learn techniques to extract relevant features from the dataset and create new features that capture the underlying patterns and relationships in the European Airbnb market. Consider factors such as proximity to attractions, transportation options, and local amenities.
4. **Machine Learning Model Building and Evaluation**: Explore various regression algorithms such as linear regression, decision trees, random forests, or gradient boosting to build a price prediction model for Airbnb listings. Train and evaluate the models using appropriate evaluation metrics such as mean squared error (MSE) or root mean squared error (RMSE).
5. **Model Interpretation**: Understand how to interpret the trained model to identify the key features and their impact on Airbnb prices. Analyse feature importance and coefficients to gain insights into the factors driving accommodation prices in European cities.

6. **User Interface Development:** Design and develop a user-friendly interface that allows users to input relevant features such as location, property type, amenities, and other factors to obtain an estimated price for an Airbnb accommodation in European cities.

7. **Deployment**: Deploy the price prediction system on a suitable platform, making it accessible for users to input accommodation details and obtain the predicted price. Ensure the system is scalable, secure, and provides accurate price estimations.

Tasks:

The project tasks should be executed in the following order:

1. **Data Collection**: Collect a comprehensive dataset specific to European Airbnb listings, including features such as location, property type, amenities, availability, pricing details, and guest reviews.
2. **Data Pre-processing**: Clean the dataset by handling missing values, performing feature scaling, outlier detection and removal, and encoding categorical variables.
3. **Feature Engineering**: Analyse the dataset and extract relevant features that capture the variations and patterns in the European Airbnb market. Create new features if necessary, considering factors such as proximity to attractions, transportation options, and local amenities.

4. **Machine Learning Model Building**: Select suitable regression algorithms and train multiple models using the pre-processed dataset. Experiment with different algorithms, hyperparameters, and ensemble methods to find the best-performing model.
5. **Model Evaluation**: Evaluate the trained models using appropriate evaluation metrics such as mean squared error (MSE) or root mean squared error (RMSE). Compare the performance of different models and select the model with the lowest error.
6. **User Interface Development**: Design and develop a user-friendly interface that allows users to input relevant features of an Airbnb accommodation in European cities. Integrate the trained model into the interface to provide estimated prices based on the user's inputs.
7. **Deployment**: Deploy the Airbnb price prediction system on a suitable platform (e.g., web application) to make it accessible for users. Ensure the system is scalable, secure, and provides accurate price estimations for Airbnb accommodations in European cities.

Evaluation:

The evaluation will consist of the following components:

1. **MCQ Questions**: A set of 30 multiple-choice questions covering the technologies and concepts used in the project, such as frontend development, backend development, database management, data pre-processing, ML model building, and deployment.
2. **Live Evaluation**: Industrial experts or faculties will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the

implemented features, and your problem-solving skills related to the project.

3. **Feedback**: The industrial experts or faculties will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This examination aims to assess your knowledge of the subjects presented and your ability to apply it to actual projects, as well as to offer insightful criticism from professionals in the field to help you develop your abilities.

Use Case 14: Airline Passenger Satisfaction Prediction & Deployment

Description:

The project aims to develop a machine learning model to predict airline passenger satisfaction based on various factors and features. By utilising a dataset specific to airline passenger reviews and feedback, the project focuses on building a model that accurately classifies whether a passenger is satisfied or dissatisfied with their flying experience. The system involves data pre-processing, feature engineering, model training and evaluation, the development of a user-friendly interface for predictions, and the deployment of the system for real-time satisfaction predictions.

Learning Outcome:

By working on this project, you will gain the following learning outcomes:

1. **Understanding Airline Passenger Satisfaction**: Familiarise with the factors that contribute to passenger satisfaction in the airline industry, such as flight punctuality, in-flight services, legroom, cleanliness, customer service, and overall flight experience.
2. **Data Pre-processing**: Develop skills in data Pre-processing techniques such as handling missing values, text pre-processing, feature scaling, and encoding categorical variables to prepare the data for machine learning models.
3. **Feature Engineering**: Learn techniques to extract meaningful features from passenger reviews and feedback, such as sentiment analysis, text mining, and feature extraction from textual data.
4. **Machine Learning Model Building and Evaluation**: Explore various classification algorithms such as logistic regression, decision trees, random forests, or support vector machines to build a model for predicting passenger satisfaction. Train and evaluate the models using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score.
5. **Model Interpretation**: Understand how to interpret the trained model to identify the most influential factors contributing to passenger satisfaction. Analyse feature importance and coefficients to gain insights into the key drivers of passenger satisfaction.

6. **User Interface Development**: Design and develop a user-friendly interface that allows users to input their flight details and obtain a prediction of their satisfaction level. The interface should provide a seamless experience for users to access the prediction system.

7. **Deployment**: Deploy the airline passenger satisfaction prediction system on a suitable platform, such as a web application or mobile app. Ensure the system is scalable, secure, and can handle real-time predictions for a large number of users.

Tasks:

The project tasks should be executed in the following order:

1. **Data Collection**: Collect a dataset containing airline passenger reviews and feedback, including features such as flight details, in-flight services, seat comfort, cleanliness, and overall satisfaction ratings.
2. **Data Pre-processing**: Clean the dataset by handling missing values, perform text Pre-processing techniques such as removing stop words, stemming, and tokenization, and encode categorical variables if necessary.
3. **Feature Engineering**: Extract meaningful features from the passenger reviews and feedback
4. **Machine Learning Model Building**: Select suitable classification algorithms and train multiple models using the pre-processed dataset. Experiment with different algorithms, hyperparameters, and ensemble methods to find the best-performing model.
5. **Model Evaluation**: Evaluate the trained models using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score. Compare the performance of different models and select the model with the highest predictive accuracy.
6. **User Interface Development**: Design and develop a user-friendly interface that allows users to input their flight details and obtain a prediction of their satisfaction level. Integrate the trained model into the interface to provide accurate predictions based on the user's inputs.
7. **Deployment**: Deploy the airline passenger satisfaction prediction system on a suitable platform, such as a web application or mobile app. Ensure the system is scalable, secure, and can handle real-time predictions for multiple users simultaneously.

Evaluation:

The evaluation will consist of the following components:

1. **MCQ Questions**: A set of 30 multiple-choice questions covering the technologies and concepts used in the project, such as frontend development, backend development, database management, data pre-processing, ML model building, and deployment.
2. **Live Evaluation**: Industrial experts or faculties will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.
3. **Feedback**: The industrial experts or faculties will provide feedback on your project

implementation, highlighting strengths and areas for improvement.

This test is designed to evaluate your understanding of the material and your ability to apply it to real-world tasks. It also intends to provide you with helpful feedback from industry experts to help you improve your skills.

Use Case 15: Classification of Salary Prediction

Description:
The project aims to develop a machine learning model to predict salary categories based on various features and factors. By utilising a dataset specific to job listings and corresponding salaries, the project focuses on building a classification model that can accurately classify whether a salary falls into low, medium, or high categories. The system involves data pre-processing, feature engineering, model training and evaluation, and the development of a user-friendly interface for salary predictions.

Learning Outcome:
By working on this project, you will gain the following learning outcomes:

1. **Understanding Salary Prediction**: Familiarise yourself with the factors that influence salary levels in job listings, such as job title, experience, education level, location, industry, and skills required.
2. Data Pre-processing: Develop skills in data Pre-processing techniques such as handling missing values, feature scaling, encoding categorical variables, and addressing class imbalance if present in the dataset.
3. **Feature Engineering**: Learn techniques to extract relevant features from the job listing dataset, such as creating new features based on education and experience, performing feature selection, and engineering features that capture the importance of specific skills or qualifications.
4. **Machine Learning Model Building and Evaluation**: Explore various classification algorithms such as logistic regression, decision trees, random forests, or gradient boosting to build a model for salary prediction. Train and evaluate the models using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score.
5. **Model Interpretation**: Understand how to interpret the trained model to identify the most influential features impacting salary predictions. Analyse feature importance to gain insights into the factors driving salary categorization.
6. **User Interface Development:** Design and develop a user-friendly interface that allows users to input relevant features such as job title, experience, education, location, and skills to obtain a predicted salary category. The interface should provide a seamless experience for users to access the salary prediction system.
7. **Deployment**: Deploy the salary prediction classification system on a suitable platform, such

as a web application, making it accessible for users to input their job details and receive the predicted salary category.

Tasks:

The project tasks should be executed in the following order:

1. **Data Collection**: Collect a dataset containing job listings and corresponding salary information, including features such as job title, experience, education, location, industry, and skills required.
2. **Data Pre-processing**: Clean the dataset by handling missing values, perform feature scaling, encode categorical variables, and address class imbalance if necessary.
3. **Feature Engineering**: Analyse the dataset and extract relevant features that capture the variations and patterns in job salaries. Create new features if necessary, considering factors such as education and experience.
4. **Machine Learning Model Building**: Select suitable classification algorithms and train multiple models using the pre-processed dataset. Experiment with different algorithms, hyperparameters, and ensemble methods to find the best-performing model.
5. **Model Evaluation**: Evaluate the trained models using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score. Compare the performance of different models and select the model with the highest classification accuracy.

6. **User Interface Development**: Design and develop a user-friendly interface that allows users to input relevant features of a job listing to obtain a predicted salary category. Integrate the trained model into the interface to provide accurate predictions based on the user's inputs.
7. **Deployment**: Deploy the salary prediction classification system on a suitable platform (e.g., web application) to make it accessible for users. Ensure the system is scalable, secure, and provides accurate salary category predictions for job listings.

Evaluation:

The evaluation will consist of the following components:

1. **MCQ Questions**: A set of 30 multiple-choice questions covering the technologies and concepts used in the project, such as frontend development, backend development, database management, data pre-processing, ML model building, and deployment.
2. **Live Evaluation**: Industrial experts or faculties will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.
3. **Feedback**: The industrial experts or faculties will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This examination aims to assess your knowledge of the subjects presented and your ability to apply it to actual projects, as well as to offer insightful criticism from professionals in the field to help you develop your abilities.

Use Case 16: Advanced Analysis of World University Rankings

Description:

The project aims to perform advanced analysis of world university rankings data to gain insights and understand the factors that contribute to a university's ranking. By utilising a dataset containing various attributes of universities and their rankings, the project focuses on exploring the data, conducting statistical analysis, and developing visualisations to uncover patterns, trends, and relationships. The analysis will involve data pre-processing, exploratory data analysis, hypothesis testing, and advanced visualisation techniques.

Learning Outcome:

By working on this project, you will gain the following learning outcomes:

1. **Understanding World University Rankings**: Familiarise with the factors and methodologies used in world university rankings, such as academic reputation, faculty quality, research output, student satisfaction, international diversity, and financial resources.
2. **Data Pre-processing**: Develop skills in data Pre-processing techniques such as handling missing values, data normalisation, and data transformation to ensure the dataset is clean and suitable for analysis.
3. **Exploratory Data Analysis**: Explore the dataset through various statistical measures, such as summary statistics, correlations, and distributions. Identify key trends, patterns, and outliers within the data.
4. **Advanced Visualization**: Utilise advanced visualisation techniques, such as scatter plots, heatmaps, treemaps, and network graphs, to visually represent the relationships between variables and rankings. Use interactive visualisations to provide deeper insights and allow users to explore the data.
5. **Interpretation and Insights**: Analyse the results of the statistical tests and visualisations to derive meaningful insights about the factors that significantly impact university rankings. Draw conclusions and make recommendations based on the analysis.

Tasks:

The project tasks should be executed in the following order:

1. **Data Collection**: Collect a dataset containing world university rankings and related attributes such as academic reputation, faculty quality, research output, student satisfaction, and financial resources.
2. **Data Pre-processing**: Clean the dataset by handling missing values, removing duplicates,

and transforming variables if necessary. Ensure the dataset is ready for analysis.

3. **Exploratory Data Analysis**: Perform exploratory data analysis to understand the distributions, correlations, and summary statistics of the variables. Identify any outliers or anomalies in the data.

4. **Hypothesis Testing**: Formulate hypotheses related to university rankings and perform appropriate statistical tests to evaluate the significance of the relationships between variables and rankings. Interpret the results of the tests.

5. **Advanced Visualization**: Create visually appealing and informative visualisations to represent the relationships between variables and rankings. Utilise interactive visualisations to allow users to explore the data and gain deeper insights.

6. **Interpretation and Insights**: Analyse the results of the statistical tests and visualisations to derive meaningful insights about the factors that significantly impact university rankings. Summarise the findings and draw conclusions.

Evaluation:

The evaluation will consist of the following components:

1. **30 MCQ Questions**: A set of multiple-choice questions covering the concepts and techniques related to university world rankings, data pre-processing, exploratory data analysis, statistical modelling, visualisation, and interpretation of results.

2. **Live Evaluation**: Industrial experts or faculties will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. **Feedback**: The industrial experts or faculties will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This test is designed to evaluate your understanding of the material and your ability to apply it to real-world tasks. It also intends to provide you with helpful feedback from industry experts and faculties to help you improve your skills.

Use Case 17: Machine Learning-Based Prediction of Engineering Placements

Description:

The project aims to develop a machine learning model to predict engineering placements based on various factors such as academic performance, technical skills, internships, and extracurricular activities. By utilising a dataset specific to engineering students and their placement outcomes, the project focuses on building a classification model that can accurately predict whether a student will get placed or not. Additionally, the project includes the deployment of the model as a web application to provide placement predictions to users in a user-friendly manner.

Learning Outcome:

By working on this project, you will gain the following learning outcomes:

1. **Understanding Engineering Placements**: Familiarize yourself with the factors that influence engineering placements, such as academic performance, technical skills, internships, projects, communication skills, and personal attributes.
2. **Data Pre-processing**: Develop skills in data Pre-processing techniques such as handling missing values, feature scaling, encoding categorical variables, and addressing class imbalance if present in the dataset.
3. **Feature Engineering**: Learn techniques to extract relevant features from the student dataset, such as creating aggregate features, deriving new features from existing ones, and identifying key predictors of placements.
4. **Machine Learning Model Building and Evaluation**: Explore various classification algorithms such as logistic regression, decision trees, random forests, or support vector machines to build a model for placement prediction. Train and evaluate the models using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score.
5. **Model Interpretation**: Understand how to interpret the trained model to identify the most influential features impacting placement predictions. Analyse feature importance and coefficients to gain insights into the key factors driving placement outcomes.
6. **User Interface Development**: Design and develop a user-friendly interface that allows students to input their academic details, technical skills, internships, and extracurricular activities to obtain a placement prediction. The interface should provide a seamless experience for users to access the placement prediction system.

7. **Deployment**: Deploy the trained model as a web application to make it accessible to users.

Implement the user interface developed earlier and integrate it with the deployed model. Ensure the application is scalable, secure, and provides accurate placement predictions for engineering students.

Tasks:

The project tasks should be executed in the following order:

1. **Data Collection**: Collect a dataset containing engineering student information, including academic performance, technical skills, internships, projects, and placement outcomes.
2. **Data Pre-processing**: Clean the dataset by handling missing values, perform feature scaling, encode categorical variables, and address class imbalance if necessary.
3. **Feature Engineering**: Analyse the dataset and extract relevant features that are indicative of placement outcomes. Create aggregate features, derive new features, and identify key predictors of placements.

4. **Machine Learning Model Building:** Select suitable classification algorithms and train multiple models using the pre-processed dataset. Experiment with different algorithms, hyperparameters, and ensemble methods to find the best-performing model.

5. **Model Evaluation**: Evaluate the trained models using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score. Compare the performance of different models and select the model with the highest classification accuracy.

6. **User Interface Development**: Design and develop a user-friendly interface that allows engineering students to input their academic details, technical skills, internships, and extracurricular activities to obtain a placement prediction.

7. **Deployment**: Deploy the trained model as a web application, integrating the user interface with the deployed model. Ensure the application is accessible, scalable, and provides accurate placement predictions for users.

Evaluation:

The evaluation will consist of the following components:

1. **MCQ Questions:** A set of 30 multiple-choice questions covering the technologies and concepts used in the project, such as frontend development, backend development, database management, data pre-processing, ML model building, and deployment.

2. **Live Evaluation**: Industrial experts or faculties will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. **Feedback:** The industrial experts or faculties will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This test is designed to evaluate your understanding of the material and your ability to apply it to real-world tasks. It also intends to provide you with helpful feedback from industry experts and faculties to help you improve your skills.

Use Case 18: Analysis of Data Science Salaries in 2023

Description:
The project's objective is to examine and learn more about data science wages in 2023. The research focuses on analysing salary trends, identifying variables driving salary variances, and giving a thorough analysis of the data science job market by using a dataset specialised to data science job positions and their accompanying salaries. Data pre-processing, exploratory data analysis, statistical modelling, and visualisation approaches will all be used in the analysis.

Learning Outcome:

By working on this project, you will gain the following learning outcomes:

1. **Understanding Data Science Salaries**: Familiarise yourself with the factors that influence data science salaries, such as experience level, education, location, industry, and job responsibilities.
2. **Data Pre-processing**: Develop skills in data pre-processing techniques such as handling missing values, data cleaning, and feature engineering to ensure the dataset is clean and suitable for analysis.
3. **Exploratory Data Analysis**: Perform exploratory data analysis to understand the distribution of data science salaries, identify outliers, and explore relationships between salary and various factors such as experience, education, and location.
4. **Statistical Modelling**: Apply statistical modelling techniques such as linear regression, decision trees, or random forests to analyse the relationship between salary and predictor variables. Identify significant predictors and evaluate the model's performance.
5. **Visualisation**: Utilise data visualisation techniques to present salary trends, compare salary distributions across different factors, and visualise the impact of various predictors on salary outcomes. Use visualisations to communicate findings effectively.
6. **Interpretation and Insights**: Analyse the results of the statistical modelling and visualisations to derive meaningful insights about the factors that significantly impact data science salaries. Identify the most influential factors and provide recommendations or insights for job seekers or employers.

Tasks:

The project tasks should be executed in the following order:

1. **Data Collection**: Collect a dataset containing data science job positions and their corresponding salaries for the year 2023. Include relevant information such as experience level, education, location, industry, and job responsibilities.
2. **Data Pre-processing**: Clean the dataset by handling missing values, remove duplicates, and perform necessary data transformations. Ensure the dataset is ready for analysis.
3. **Exploratory Data Analysis**: Perform exploratory data analysis to understand the distribution of data science salaries, identify outliers, and explore relationships between salary and various factors.
4. **Statistical Modelling**: Apply appropriate statistical modelling techniques to analyse the relationship between salary and predictor variables. Train and evaluate models using appropriate evaluation metrics.
5. **Visualisation**: Create informative visualisations to present salary trends, compare salary

distributions across different factors, and visualise the impact of various predictors on salary outcomes. Use interactive visualisations if possible.

6. **Interpretation and Insights**: Analyse the results of the statistical modelling and visualisations to derive meaningful insights about the factors that significantly impact data science salaries. Summarise findings and provide recommendations or insights for job seekers or employers.

Evaluation:

The evaluation will consist of the following components:

1. **30 MCQ Questions**: A set of multiple-choice questions covering the concepts and techniques related to data science salaries, data pre-processing, exploratory data analysis, statistical modelling, visualisation, and interpretation of results.

2. **Live Evaluation**: Industrial experts or faculties will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. **Feedback:** The industrial experts or faculties will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This test is designed to evaluate your understanding of the material and your ability to apply it to real-world tasks. It also intends to provide you with helpful feedback from industry experts and faculties to help you improve your skills.

Use Case 19: World Energy Consumption in Different Region

Description:

The project's objective is to examine global energy consumption trends and gather new knowledge about them. The research focuses on studying energy consumption patterns, identifying the key sources of energy, and analysing the distribution of energy consumption across regions by utilising a comprehensive dataset on energy consumption across various countries and energy sources. Techniques for exploratory data analysis, and visualisation will all be used in the analysis.

Learning Outcome:

By working on this project, you will gain the following learning outcomes:

1. **Understanding World Energy Consumption:** Familiarise with the global energy consumption landscape, including the types of energy sources used, regional variations, and trends over time.

2. **Data Pre-processing**: Develop skills in data Pre-processing techniques such as handling missing values, data cleaning, and data transformation to ensure the dataset is clean and suitable for analysis.

3. **Exploratory Data Analysis**: Perform exploratory data analysis to understand the distribution of energy consumption across countries and regions. Identify the primary sources of energy and explore relationships between energy consumption and various factors such as population, GDP, and geographical location.

4. **Visualisation**: Utilise data visualisation techniques to present energy consumption trends, compare energy consumption across countries and regions, and visualise the impact of various factors on energy consumption patterns. Use visualisations to communicate findings effectively.

5. **Interpretation and Insights**: Analyse the results of the statistical modelling and visualisations to derive meaningful insights about the factors that influence energy consumption. Identify the primary energy sources, understand regional variations, and provide recommendations or insights for energy policymakers and stakeholders.

Tasks:

The project tasks should be executed in the following order:

1. **Data Collection**: Collect a comprehensive dataset containing energy consumption data across different countries and energy sources. Include relevant information such as energy production, energy consumption, population, GDP, and geographical location.

2. **Data Pre-processing**: Clean the dataset by handling missing values, remove duplicates, and perform necessary data transformations. Ensure the dataset is ready for analysis.

3. **Exploratory Data Analysis**: Perform exploratory data analysis to understand the distribution of energy consumption across countries and regions. Identify the primary sources of energy, examine regional variations, and explore relationships between energy consumption and other factors.

4. **Visualisation:** Create informative visualisations to present energy consumption trends, compare energy consumption across countries and regions, and visualise the impact of various factors on energy consumption patterns. Use interactive visualisations if possible.

5. **Interpretation and Insights**: Analyse the results of the statistical modelling and visualisations to derive meaningful insights about the factors that influence energy consumption. Summarise findings and provide recommendations or insights for energy policymakers and stakeholders.

Evaluation:

The evaluation will consist of the following components:

1. **30 MCQ Questions:** A set of multiple-choice questions covering the concepts and techniques related to energy consumption in different regions, data pre-processing, exploratory

data analysis, visualisation, and interpretation of results.

2. **Live Evaluation:** Industrial experts or faculties will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. **Feedback:** The industrial experts or faculties will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This test is designed to evaluate your understanding of the material and your ability to apply it to real-world tasks. It also intends to provide you with helpful feedback from industry experts and faculties to help you improve your skills.

Use Case 20: IPL Data Analysis Description:

The IPL Data Analysis project focuses on analysing the data from the Indian Premier League (IPL), a popular professional Twenty20 cricket league in India. By working on this project, participants will gain insights into team performance, player statistics, match results, and various aspects of the IPL. The project aims to provide valuable information for cricket enthusiasts, team management, and decision-making in the context of the IPL. Participants will utilize data analysis techniques, visualization tools, and statistical methods to analyse player performances, team strategies, match outcomes, and other relevant factors.

TOTAL : 45 PERIODS

Learning Outcome:

By working on the IPL Data Analysis project, participants will have the opportunity to expand their knowledge and gain expertise in the following areas:

1. Data Pre-processing and cleaning techniques for IPL data: Participants will learn how to handle missing values, inconsistencies, and outliers in the IPL dataset. They will gain experience in data cleaning and transformation techniques to ensure the data is suitable for analysis.

2. Exploratory data analysis (EDA) to uncover patterns, trends, and insights in IPL matches: Participants will explore various statistical and visualization techniques to identify patterns in player performance, team strategies, match results, and other factors relevant to the IPL. They will gain a deeper understanding of the dynamics and trends within the league.

3. Player performance analysis and statistical evaluation: Participants will analyze player statistics, such as batting average, bowling economy, strike rate, and other key metrics. They will apply statistical methods to evaluate player performance and identify impactful players in different aspects of the game.

4. Analysing team strategies, match results, and factors influencing success: Participants will delve into team strategies, tactical decisions, and their impact on match outcomes. They will identify the key factors that contribute to team success, such as batting order, bowling variations, fielding efficiency, and team composition.

5. **Utilizing data visualization tools to present insights and trends in IPL data**: Participants will gain proficiency in creating visually appealing and informative visualizations using tools such as matplotlib, seaborn, or Plotly. They will learn how to effectively communicate complex insights from IPL data through charts, graphs, and interactive visualizations.

6. **Identifying key players, team dynamics, and factors contributing to match outcomes:** Participants will analyze the performance of individual players and their impact on team success. They will gain insights into the dynamics of team performance, understanding how different players contribute to match outcomes and overall team performance.

Tasks:

1. Data Collection and Pre-processing:

- Collecting IPL data, including match results, player statistics, and team information.
- Cleaning and pre-processing the data, handling missing values and inconsistencies.

2. Exploratory Data Analysis (EDA):

- Analyzing and visualizing IPL data to identify patterns, trends, and interesting insights.
- Exploring relationships between player performance, team strategies, and match outcomes.

3. Player Performance Analysis:

- Analyzing player statistics, such as batting average, bowling economy, and fielding performance.
- Comparing player performance across different seasons and teams.

4. Statistical Evaluation:

- Applying statistical methods to evaluate player performance and identify impactful players.
- Assessing the significance of differences between player statistics using hypothesis testing.

5. Team Performance and Match Analysis:

- Analyzing team strategies, tactics, and their impact on match outcomes.
- Examining factors influencing success, such as batting order, bowling variations, and fielding efficiency.

6. Data Visualization:

- Creating visualizations (e.g., bar charts, heatmaps) to present insights and trends in IPL data.
- Developing interactive dashboards to explore match results, player statistics, and team performance.

7. Strategies and Recommendations:

- Based on analysis findings, developing strategies and recommendations for team management.
- Suggesting improvements in team selection, player roles, and match strategies.

Evaluation:

The evaluation will consist of the following components:

1. **30 MCQ Questions:** A set of multiple-choice questions covering the concepts and techniques related to energy consumption in different regions, data preprocessing, exploratory data analysis, visualisation, and interpretation of results.

2. **Live Evaluation:** Industrial experts or faculties will conduct a live evaluation session where they will assess your understanding of the project components, your ability to explain the implemented features, and your problem-solving skills related to the project.

3. **Feedback:** The industrial experts or faculties will provide feedback on your project implementation, highlighting strengths and areas for improvement.

This test is designed to evaluate your understanding of the material and your ability to apply it to real-world tasks. It also intends to provide you with helpful feedback from industry experts and faculties to help you improve your skills.

SB8050 **DevOps** **L T P C**

1 0 2 2

Module 1: Introduction to DevOps, Agile Development and Scrum

Overview of DevOps - What is DevOps and why is it important? - How DevOps helps solve various Industry problems - Dev Challenges and Solution - Ops Challenges and Solution - Case Study: Deloitte New Zealand - Case Study: Daimler - Trucks NA - DevOps Market Trends - Why Agile - Agile Manifesto

- Agile Principles - Agile Methodologies, Frameworks and processes - Agile Leadership - Agile with DevOps - Scrum theory and principles - Scrum Foundations (5 Scrum Values) - The Scrum Framework

- The Definition of Done - Running a Scrum project - Working with people and teams - Scrum in your organization - Product Owner roles and responsibilities - Development team roles and responsibilities

Module 2: Python for DevOps

Python Applications in DevOps - Variables, Operands and expressions - Conditional statements

- Loops

& Structural pattern matching - Accepting user input and eval function - Lists - Tuples - Strings manipulation - Sets and set operations - Python dictionary - User-defined functions - Function parameters and different types - Files input/output functions - Global variables - Global keyword - Lambda functions - Built-in functions - Object-oriented concepts - Public, protected and private attributes - Class variable and instance variable - Constructor and destructor - Inheritance and its types

- Method resolution order - Overloading and overriding - Setter and setter methods - Standard libraries

- Packages and import statements - Reload function - Creating a module - Important modules in python

- Sys module - OS, Math, Date-time, Random and JSON modules - Regular expression – Exception

handling - **Basics of data analysis**:- NumPy - Arrays: Array operations -Indexing, slicing, and Iterating

- NumPy array attributes - Matrix product - NumPy functions - Array manipulation

Module 3: Introduction to Containers w/ Docker, Kubernetes & OpenShift
Containerization - Namespaces - Docker - Docker Architecture - Container Lifecycle - Docker CLI - Port Binding - Detached and Foreground Mode - Dockerfile - Dockerfile Instructions - Docker Image - Docker Registry - Container Storage - Volumes - Docker Compose - Docker Swarm - Introduction to Container Orchestration - Kubernetes Core Concepts - Understanding Pods - ReplicaSet and Replication Controller - Deployments - DaemonSets - Rolling Updates and Rollbacks - Scaling Application - Get started with container technology - Openshift architecture - Installing openshift container platform - Openshift networking concepts - Creating applications with the openshift web console - Manage Containers - Manage Container Images - Create custom container images - Deploy containerized applications on OpenShift - Troubleshoot containerized applications

Module 4: Application Development using Microservices and Serverless
Introduction to Microservices - Monolithic Architecture - SOA Architecture - Key benefits of Microservices - Challenges in Microservices - Introducing Microservices Architecture - Microservices Design Patterns - Use case: Apollo Store - Decomposition Strategies - Obstacles in Decomposition - Introduction of Docker with Microservices - Managing Containers in Microservices - Operations in Openshift - Streamlining Services in OpenShift

Module 5: Continuous Integration and Continuous Delivery (CI/CD)
Version Control - Git Introduction - Git Installation - Commonly used commands in Git - Working with Remote repository - Branching and merging in Git - Merge Conflicts - Stashing, Rebasing, Reverting, and Resetting - Git Workflows - Configuration Code - Resources and Recipes - Data bags - Using community cookbooks - Chef Managed Infrastructure - ChefDK - Chef -repo - .chef Directory - Nodes - Chef Resources - Package - Resource Collection - Security Model of Chef - Introduction to Configuration Management - Introduction to Puppet - Puppet architecture - Puppet Manifest - Modules in Puppet - Server Configuration Management

Books:
1. DevOps for Beginners: DevOps Software Development Method Guide for Software Developers and IT Professionals by Joseph Joyner
2. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations by Gene Kim & Jez Humble & Patrick Debois & John Willis

Online References:
1. Udemy - DevOps Beginners to Advanced Decoding DevOps withz Projects https://www.udemy.com/course/decodingdevops/
2. Udemy - Learn DevOps: Docker, Kubernetes, Terraform and Azure DevOps https://www.udemy.com/course/devops-with-docker-kubernetes-and-azure-devops/
3. GUVI - Kubernetes https://www.guvi.in/courses/it-and-software/kubernetes/
4. Udemy - DevOps Project - 2022: CI/CD with Jenkins Ansible Kubernetes https://www.udemy.com/course/valaxy-devops/
5. GUVI - AWS https://www.guvi.in/courses/database-and-cloud-computing/aws

6. Sample 20 Use Cases

Use Case1: Microservices Deployment with Docker and Kubernetes

Description:
Containerize individual microservices of an application using Docker and orchestrate their

deployment, scaling, and load balancing using Kubernetes.
Tasks: **All the below tasks has to be completed using Agile Methodology.**

1. Identify Microservices

● Identify the components or functionalities of the application that can be split into microservices. Determine the boundaries and responsibilities of each microservice.

2. Containerize Microservices

● Create a Dockerfile for each microservice, specifying the base image, dependencies, and configuration required to run the microservice.

● Build Docker images for each microservice using the Dockerfiles.

● Push the Docker images to a container registry (e.g., Docker Hub) for easy accessibility.

3. Set Up a Kubernetes Cluster:

● Set up a Kubernetes cluster using a platform like Minikube, kind, or a managed Kubernetes service.

● Install and configure the Kubernetes command-line tool (kubectl) to interact with the cluster.

4. Create Kubernetes Deployment Manifests:

● Define Kubernetes Deployment manifests for each microservice. These manifests specify the desired state of each microservice and its container image.

● Configure resource limits, environment variables, ports, and other necessary parameters for each microservice in the manifests.

5. Create Kubernetes Service Manifests:

● Define Kubernetes Service manifests to expose the microservices internally or externally.

● Specify the service type (ClusterIP, NodePort, LoadBalancer), ports, and target microservice in the manifest.

6. Apply Deployment and Service Manifests:

● Use kubectl to apply the Deployment and Service manifests to the Kubernetes cluster.

● This will create the necessary pods (running instances of microservice containers) and services for each microservice.

7. Validate Microservices Deployment:

● Use kubectl commands to verify the status and health of the microservices, pods, and services.

● Ensure that the microservices are running and accessible within the cluster.

8. Test Communication Between Microservices:

- Set up communication between the microservices by utilizing service names or IP addresses.
- Test the communication between microservices to ensure proper integration and functionality.

9. Scale Microservices:

- Use Kubernetes commands or configure autoscaling policies to scale the microservices horizontally based on metrics like CPU usage or request load.

- Monitor the scaling behavior and ensure that the microservices can handle increased traffic.

TOTAL:45 PERIODS

Learning Outcome:

1. Understanding Microservices Architecture: Understanding of microservices architecture and its benefits, including modularity, scalability, and independent deployment of services.

2. Proficiency in Docker: How to containerize individual microservices using Docker, including writing Dockerfiles, building Docker images, and pushing them to a container registry.

3. Mastery of Kubernetes: Hands-on experience with Kubernetes, including setting up a cluster, deploying microservices using Kubernetes manifests, and managing the lifecycle of containers.

4. Orchestration and Scaling: Utilization of Kubernetes for orchestration and scaling of microservices, including scaling pods based on metrics, implementing load balancing, and ensuring high availability.

5. Service Discovery and Communication: Understanding service discovery mechanisms in Kubernetes and learning how to establish communication between microservices using Kubernetes services.

Evaluation:

1.MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as docker, kubernetes, git, openshift, chef and puppet

2.Live Evaluation: Industrial faculty / experts will conduct a live evaluation session where they will assess your understanding for project components, your ability to explain the implemented features and your problem-solving skills related to the project

3.Feedback: The industrial experts/faculty will provide feedback on your project implementation highlighting strengths and areas for improvement.

Use Case 2: Continuous Integration and Delivery Pipeline with Jenkins and Docker

Description:
Set up a CI/CD pipeline using Jenkins and Docker, automating the build, testing, and deployment of applications in a Dockerized environment.

Tasks: **All the below tasks has to be completed using Agile Methodology.**

1. Set up Jenkins:

- Install Jenkins on a server or use a cloud-based Jenkins service.

- Configure Jenkins with necessary plugins, such as the Docker plugin, Git plugin, and Pipeline plugin.

2. Set up Source Code Repository:

- Create a version control repository (e.g., Git repository) to store the application source code.
- Connect Jenkins to the repository and configure the repository's webhook or polling mechanism to trigger Jenkins builds.

3. Configure Jenkins Job:

- Create a new Jenkins job to manage the CI/CD pipeline.
- Define the job as a Jenkins pipeline, which allows for a more flexible and scriptable approach to CI/CD.

4. Define Build Stage:

- Define a build stage in the Jenkins pipeline to pull the source code from the repository and build the Docker image.
- Configure the Docker plugin to build the Docker image based on a Dockerfile in the source code repository.

5. Configure Deployment Stage:

- Define a deployment stage in the Jenkins pipeline to deploy the Docker image to the target environment.
- This can be a staging environment, production environment, or any other deployment target.

6. Configure Deployment Tools:

- Integrate deployment tools (e.g., Kubernetes, Docker Swarm) with Jenkins to manage the deployment of Docker images.
- Use the respective plugins or command-line tools to interact with the deployment target.

7. Implement Continuous Delivery:

- Set up a continuous delivery mechanism to automatically deploy the application to the production environment after successful integration and acceptance testing.
- Ensure the pipeline includes necessary approval steps or gates to prevent unauthorized deployments.

8. Configure Notifications and Reporting:

- Configure notifications to relevant stakeholders (e.g., email notifications, messaging platforms) on build status, test results, and deployment updates.

Learning Outcome:

1. Understanding CI/CD Principles: CI/CD concepts, including the benefits of automation, continuous integration, and continuous delivery, and the importance of a well-defined pipeline.

2. Proficiency in Jenkins: Setting up and configuring Jenkins, including installing plugins, creating and managing Jenkins jobs, and utilizing the Jenkins pipeline as code approach.

3. Docker Knowledge: Integration of Docker into the CI/CD pipeline, including building Docker images, pushing images to a registry, and utilizing Docker containers for consistent and reproducible builds.
4. Version Control Integration: Participants will understand how to integrate Jenkins with version control systems (e.g., Git) to automatically trigger builds and deployments upon code changes, enabling continuous integration.

Evaluation:

1.MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as docker, kubernetes, git, openshift, chef and puppet

2.Live Evaluation: Industrial faculty / experts will conduct a live evaluation session where they will assess your understanding for project components, your ability to explain the implemented features and your problem-solving skills related to the project

3.Feedback: The industrial experts/faculty will provide feedback on your project implementation highlighting strengths and areas for improvement.

Use Case 3: Blue-Green Deployment with Kubernetes

Description:

Implement a blue-green deployment strategy using Kubernetes, allowing for seamless and zero-downtime application updates and rollbacks.

Tasks: **All the below tasks has to be completed using Agile Methodology.**

1. Set up Kubernetes Cluster:

- Set up a Kubernetes cluster using a platform like Minikube, kind, or a managed Kubernetes service.

- Install and configure the Kubernetes command-line tool (kubectl) to interact with the cluster.

2 Create Kubernetes Deployment Manifests:

- Define Kubernetes Deployment manifests for both the blue and green versions of the application.

- Specify the desired state of each deployment, including the container image, resource requirements, and any environment variables.

3. Create Kubernetes Service Manifests:

- Define Kubernetes Service manifests to expose the blue and green deployments internally or externally.

- Specify the service type (ClusterIP, NodePort, LoadBalancer), ports, and target deployments in the manifests.

4. Apply Blue Deployment:

- Use kubectl to apply the blue deployment manifest to the Kubernetes cluster.

- This will create the necessary pods (running instances of the blue deployment) and services for the blue version of the application.

5. Verify Blue Deployment:

- Use kubectl commands to verify the status and health of the blue deployment, pods, and services.

- Ensure that the blue version of the application is running and accessible within the cluster.

7. Create Green Deployment:

- Update the deployment manifest to reflect the green version of the application, including the updated container image or code.

- Apply the green deployment manifest to the Kubernetes cluster, creating the necessary pods and services for the green version.

8. Configure Blue-Green Service Switching:

- Utilize Kubernetes service labels, selectors, and ingress controllers to switch traffic between the blue and green deployments seamlessly.

- Set up a load balancer or ingress rule to direct traffic to the appropriate deployment.

9. Route Traffic to Green Deployment:

- Gradually shift traffic from the blue deployment to the green deployment.

- Monitor the traffic and gradually increase the percentage of traffic routed to the green deployment based on predefined criteria or testing results.

10. Validate Green Deployment:

- Monitor the health and performance of the green deployment, pods, and services.

- Conduct additional testing and quality checks to ensure the stability and functionality of the green version of the application.

11. Monitor and Rollback (if needed):

- Continuously monitor the performance, any anomalies related to the green deployment.

- If any issues or negative impact is detected, rollback to the blue deployment by redirecting traffic back to the blue deployment.

Learning Outcomes:

1. Understanding Deployment Strategies: Understanding of deployment strategies, specifically the Blue-Green Deployment approach, and how it enables zero-downtime updates and rollback capabilities.

2. Proficiency in Kubernetes: Setting up and managing Kubernetes clusters, including deploying applications, configuring services, and utilizing Kubernetes manifests for deployment configuration.

3. Containerization and Image Management: How to containerize applications using Docker and manage container images within a Kubernetes environment, ensuring consistency and reproducibility in deployments.

4. Version Control and Release Management: Understanding the importance of version control and release management in the context of Blue-Green Deployment.

5. Testing and Quality Assurance: Implementing testing strategies within the Blue-Green Deployment process, including integration testing, performance testing, and quality assurance measures, to ensure the stability and functionality of the application.

6. Continuous Monitoring and Observability: How to monitor and observe the performance and health of the application during the Blue-Green Deployment process, using Kubernetes monitoring tools, logs, and metrics to detect anomalies and ensure optimal system behavior.

Evaluation:

1.MCQ Questions: A set of multiple choice questions covering the technologies and concepts used in the project, such as docker, kubernetes, git, openshift, chef and puppet

2.Live Evaluation: Industrial faculty / experts will conduct a live evaluation session where they will assess your understanding for project components, your ability to explain the implemented features and your problem-solving skills related to the project

3.Feedback: The industrial experts/faculty will provide feedback on your project implementation highlighting strengths and areas for improvement.

Use case 4: Deployment of a Mobile Application

Description:

Create a project to build and deploy a mobile application for a fictitious company. The application aims to provide users with a platform to browse and purchase products from the company's catalog. The project will incorporate DevOps practices, Agile principles, and Scrum framework throughout its lifecycle.

Tasks: **All the below tasks has to be completed using Agile Methodology.**

1. Containerize the Application:

○ Use Docker to containerize the mobile application and its dependencies. This ensures consistency in the development and deployment environments across different stages of the pipeline.

○ Create a Dockerfile that defines the application's runtime environment, including the base image, dependencies, and any necessary configuration.

2. Implement Continuous Integration (CI):

○ Set up a CI server like Jenkins or GitLab CI/CD to automate the build, test, and integration process.

○ Configure the CI server to pull the application's source code from a version control repository and build a Docker image using the Dockerfile.

○ Execute automated tests within the Docker container to ensure code quality and functionality.

3. Continuous Delivery and Deployment (CD):

○ Use Docker images as artifacts for deployment.

○ Set up a Docker registry (e.g., Docker Hub) to store and manage Docker images.

○ Configure the CI server to push the Docker images to the registry once the build and tests pass successfully.

○ Implement CD pipelines using tools like Jenkins, GitLab CI/CD, or Kubernetes Helm charts to automate the deployment process.

○ Define deployment configurations using Docker Compose or Kubernetes manifests to specify the required services, environment variables, and networking.

4. Infrastructure Orchestration with Docker and Kubernetes:

○ Utilize Docker Swarm or Kubernetes for container orchestration and management.

○ Define the infrastructure requirements and configuration using declarative manifests (e.g., Docker Compose files or Kubernetes YAML files).

○ Deploy the containerized application on a Docker Swarm cluster or Kubernetes cluster.

○ Leverage features like scaling, load balancing, and health checks to ensure the application's availability and scalability.

5. Monitoring and Logging:

○ Implement monitoring and logging tools (e.g., ELK stack) to gain insights into the application's performance, resource utilization, and logs.

○ Configure the monitoring system to collect metrics and monitor the Docker containers, application performance, and infrastructure health.

Learning Outcome:

1. Understanding of DevOps Principles: Deeper understanding of DevOps principles, such as collaboration, automation, and continuous delivery.

2. Proficiency in Docker: Hands-on experience with Docker, including containerizing applications,

managing Docker images, and deploying containers.

3. Mastery of CI/CD Pipeline: Practical exposure to building and configuring CI/CD pipelines using tools like Jenkins or GitLab CI/CD.

4. Application of Agile and Scrum: Explore Agile principles and Scrum framework in a real-world software development project. They will learn about iterative development, adaptive planning, and regular feedback cycles, allowing them to respond effectively to changing requirements and deliver customer-centric solutions.

5. Infrastructure Orchestration with Docker and Kubernetes: Deploying and managing containerized applications using Docker Swarm or Kubernetes. Defining and configuring the infrastructure requirements, scale applications, and ensure high availability and resilience.

6. Monitoring and Troubleshooting: Monitoring and logging tools are used to gain insights into application performance and troubleshoot issues. To collect and analyze metrics, monitor containerized environments, and effectively troubleshoot and debug container-based applications.

Evaluation:

1.MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as docker, kubernetes, git, openshift, chef and puppet

2.Live Evaluation: Industrial faculty / experts will conduct a live evaluation session where they will assess your understanding for project components, your ability to explain the implemented features and your problem-solving skills related to the project

3.Feedback: The industrial experts/faculty will provide feedback on your project implementation highlighting strengths and areas for improvement.

Use Case 5: Deploying a Scalable Web Application with Docker, Kubernetes, and OpenShift

Description:

The project involves deploying a scalable web application using Docker containers, managing orchestration with Kubernetes, and leveraging the features of OpenShift, a container platform built on Kubernetes. The web application could be a simple e-commerce website, a blogging platform, or any other application that requires scalability and high availability.

Tasks: **All the below tasks has to be completed using Agile Methodology.**

1. Containerizing the Web Application:

○ Dockerize the web application by creating a Dockerfile that defines the application's dependencies and configurations.

○ Build a Docker image for the web application, including all necessary files and dependencies.

2. Setting up Kubernetes Cluster:

○ Provision a Kubernetes cluster using Minikube.

○ Configure the cluster with the necessary resources and networking settings.

3. Deploying the Application on Kubernetes:

- ○ Create Kubernetes deployment manifests that define the desired state of the application, including the Docker image, resource limits, and environment variables.
- ○ Apply the deployment manifests to the Kubernetes cluster to create and manage the application's pods.

4. Ensuring High Availability and Scalability:

- ○ Configure Kubernetes Horizontal Pod Autoscaler (HPA) to automatically scale the application based on metrics such as CPU utilization or incoming traffic.

- ○ Set up Kubernetes Ingress to route incoming traffic to the application pods, allowing for load balancing and high availability.

5. Continuous Integration and Deployment:

- ○ Integrate the project with a CI/CD tool like Jenkins or GitLab CI/CD to automate the build, test, and deployment processes.

- ○ Create a pipeline that triggers builds upon code changes, runs tests, and deploys the updated application to the Kubernetes cluster.

6. Utilizing OpenShift Platform:

- ○ Deploy the Kubernetes cluster on OpenShift, leveraging its additional features such as integrated container registry, image streams, and build configurations.

- ○ Utilize OpenShift's web console or command-line tools to manage the application deployment, scaling, and monitoring.

Learning Outcome:

1. Containerization: Participants will gain a solid understanding of containerization principles and techniques using Docker, including creating Dockerfiles, building Docker images, and managing container configurations.

2. Kubernetes Orchestration: Participants will become proficient in deploying and managing applications on Kubernetes clusters, understanding concepts like pods, deployments, services, and ingress. They will learn how to scale applications horizontally, ensure high availability, and utilize Kubernetes features such as autoscaling and load balancing.

3. OpenShift Platform: Participants will gain hands-on experience with OpenShift, a container platform built on Kubernetes

Evaluation:

1.MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as docker, kubernetes, git, openshift, chef and puppet

2.Live Evaluation: Industrial faculty / experts will conduct a live evaluation session where they will assess your understanding for project components, your ability to explain the implemented features and your problem-solving skills related to the project

3.Feedback: The industrial experts/faculty will provide feedback on your project implementation highlighting strengths and areas for improvement.

Use Case 6: Scalable Web Application with Docker Swarm

Description:
Deploy a scalable web application using Docker Swarm, allowing for easy horizontal scaling and load balancing.

Tasks: **All the below tasks has to be completed using Agile Methodology.**

1. Set up Docker Swarm:

- Set up a group of Docker hosts that will serve as the swarm nodes. Ensure that Docker is installed on each host.
- Choose one host to be the swarm manager.

- Note down the join token displayed after initializing the swarm.

2. Containerize the Web Application:

- Create a Dockerfile that defines the application's dependencies, configurations, and instructions to build the Docker image.

- Push the Docker image to a registry if necessary.

Step 3: Create a Docker Compose File:
- Create a Docker Compose YAML file that describes the desired state of the application.

- Define services for each component of the web application, including the Docker image, ports, volumes, environment variables, and any necessary configurations.

- Specify the desired number of replicas for each service.

Step 4: Deploy the Application Stack:
- Deploy the Docker stack.
- Docker Swarm will create the specified number of replicas for each service and distribute them across the swarm nodes.
- Verify that the services are running.

Step 5: Scale the Application:
- Scale the application services and specify the desired number of replicas for each service.
- Docker Swarm will automatically adjust the number of replicas across the swarm nodes. Step 6: Load Balancing and Service Discovery:
- Docker Swarm provides built-in load balancing for services. Incoming traffic is automatically distributed among the available replicas of the service.
- Optionally, configure an external load balancer or use Docker Swarm's built-in routing mesh to expose the application services.

Step 7: Monitor and Manage the Swarm:

- Use Docker Swarm to monitor the state of services, replicas, and nodes.
- Utilize third-party monitoring tools or built-in Docker Swarm monitoring features to gain insights into the swarm's performance and health.

Learning Outcome:

1. Containerization: Containerization principles and practices using Docker, including creating Docker images, defining dependencies, and configuring containerized applications.

2. Orchestration with Docker Swarm: Set up and manage a Docker Swarm cluster, including initializing the swarm, joining nodes, and distributing services across the swarm for scalability and high availability.

3. Service Scaling and Load Balancing: How to scale services within a Docker Swarm cluster by adjusting the number of replicas.

4. Application Deployment and Stack Management: Gaining hands-on experience deploying a web application using Docker Compose, specifying services, replicas, and configurations in a YAML file. To deploy and manage the application stack within the Docker Swarm cluster.

Evaluation:

1.MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as docker, kubernetes, git, openshift, chef and puppet

2.Live Evaluation: Industrial faculty / experts will conduct a live evaluation session where they will assess your understanding for project components, your ability to explain the implemented features and your problem-solving skills related to the project

3.Feedback: The industrial experts/faculty will provide feedback on your project implementation highlighting strengths and areas for improvement.

Use Case 7: Hybrid Cloud Deployment with OpenShift. Description:
Deploy an application across multiple cloud providers using OpenShift, enabling seamless deployment and management in a hybrid cloud environment.

Tasks: **All the below tasks has to be completed using Agile Methodology.**

1. Set Up OpenShift Cluster:

- Choose a cloud provider or an on-premises environment to host your OpenShift cluster.

- Provision the required infrastructure resources, such as virtual machines or bare-metal servers, with the necessary networking and security configurations.

- Install OpenShift on the infrastructure by following the official documentation and guidelines provided by Red Hat.

- Configure the OpenShift cluster with appropriate authentication mechanisms, user access controls, and networking settings.

2. Establish Connectivity to Public Cloud:

- Set up a connection between your on-premises OpenShift cluster and the public cloud provider of your choice.

- Configure VPN or Direct Connect services to establish a secure and reliable network connection between the on-premises environment and the public cloud.

3. Define Cluster Zones:

- Divide the OpenShift cluster into zones, separating the on-premises nodes from the cloud nodes.

- Assign labels or tags to each node based on their zone and location, enabling resource allocation and scheduling based on these attributes.

4. Configure Hybrid Networking:

- Set up networking configurations that allow communication between the on-premises environment and the public cloud.

- Configure appropriate routing, load balancing, and firewall rules to enable seamless connectivity across the hybrid cloud deployment.

5. Deploy Applications:

- Containerize your applications using Docker and create appropriate Docker images.

- Define Kubernetes deployment manifests or OpenShift deployment configurations to describe the desired state of your application, including the required resources, environment variables, and service configurations.

- Deploy the applications onto the OpenShift cluster, taking advantage of the hybrid cloud deployment capabilities.

- Use OpenShift features like Route objects or Ingress controllers to expose the applications and make them accessible from both on-premises and cloud environments.

6. Implement Hybrid Cloud Data Management:

- Define a data management strategy that encompasses both on-premises and cloud storage systems.

- Utilize storage solutions like NFS, object storage, or cloud-specific storage services to ensure seamless data access and replication across the hybrid cloud deployment.

7. Monitor and Manage the Hybrid Cloud Deployment:

- Set up monitoring and logging systems to collect metrics and logs from both on-premises and cloud environments.

- Use OpenShift monitoring and observability features or third-party tools to gain insights into the performance, health, and availability of the hybrid cloud deployment.

- Implement automation and management tools to handle tasks such as scaling, rolling updates, and backup/restoration procedures across the hybrid cloud infrastructure.

Learning Outcomes:

1. Understanding Hybrid Cloud Architecture: Understanding of hybrid cloud architecture and its benefits, including leveraging both on-premises and public cloud resources for increased

scalability, flexibility, and cost-effectiveness.

2. Proficiency in OpenShift Installation and Configuration: Develop proficiency in installing and configuring OpenShift, a container platform built on Kubernetes, across multiple environments, including on-premises and public cloud.

3. Networking and Connectivity: How to establish network connectivity between the on-premises environment and the public cloud, ensuring seamless communication between the different components of the hybrid cloud infrastructure.

4. Containerization and Deployment: Hands-on experience containerizing applications using OpenShift's container runtime, building Docker images, and deploying containerized applications across the hybrid cloud environment.

5. Managing Hybrid Cloud Resources: How to manage resources in a hybrid cloud deployment, including provisioning and scaling resources both on-premises and in the public cloud, optimizing resource utilization, and ensuring high availability.

Evaluation:

1.MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as docker, kubernetes, git, openshift, chef and puppet

2.Live Evaluation: Industrial faculty / experts will conduct a live evaluation session where they will assess your understanding for project components, your ability to explain the implemented features and your problem-solving skills related to the project

3.Feedback: The industrial experts/faculty will provide feedback on your project implementation highlighting strengths and areas for improvement.

Use Case 8: Secure Containerization with Docker and Kubernetes

Description:

Implement security best practices in containerization using Docker and Kubernetes, including image vulnerability scanning, pod security policies, and network policies.

Tasks: **All the below tasks has to be completed using Agile Methodology.**

1. Secure Docker Images:

- Use a secure base image from trusted sources like the Docker official repository or certified vendors.

- Regularly update the base image to include security patches and fixes.

- Scan Docker images for vulnerabilities.

- Avoid including unnecessary or vulnerable software packages in the Docker image.

2. Secure Docker Container Runtime:

- Utilize user namespaces and restrict container privileges to minimize the impact of any potential security breaches.

- Enable Docker Content Trust to ensure the integrity and authenticity of the pulled Docker images.

- Implement container resource limits to prevent resource abuse and ensure fair resource allocation.

Step 3: Secure Kubernetes Cluster:

- Secure the Kubernetes cluster by following best practices like enabling RBAC (Role-Based Access Control), implementing network policies, and disabling insecure communication protocols.

- Regularly update Kubernetes components to benefit from security patches and fixes.

- Apply pod security policies to enforce security measures like limiting container capabilities, controlling host access, and setting security context constraints.

Step 4: Secure Kubernetes Manifests:

- Implement security-focused practices while defining Kubernetes manifests.

- Avoid hardcoding sensitive information like credentials and secrets in manifests.

- Use Kubernetes secrets to manage and protect sensitive data.

- Implement resource limits and quotas to prevent resource exhaustion attacks.

Step 5: Secure Container Communication:

- Utilize secure communication channels like HTTPS or TLS for inter-container communication and API interactions.
- Implement network isolation and segmentation to prevent unauthorized access to containers.

- Leverage Kubernetes network policies to control ingress and egress traffic between containers and pods.

Step 6: Secure Secrets Management:

- Store sensitive information like credentials, API keys, and certificates securely using Kubernetes Secrets, HashiCorp Vault, or other secure storage solutions.

Learning Outcomes:

1. Understanding Container Security Principles: Comprehensive understanding of container security principles and best practices, including the importance of secure base images, container runtime security, secure communication, and secrets management.

2. Proficiency in Docker Security Measures: Implementing security measures within Docker, such as using secure base images, enabling content trust, implementing resource limits, and securing container communication.

3. Proficiency in Kubernetes Security Measures: Secure a Kubernetes cluster by implementing RBAC, network policies, and pod security policies. They will gain expertise in securing Kubernetes components and managing secure communication within the cluster.
4. Secrets Management and Encryption: Managing sensitive information using Kubernetes secrets or secure storage solutions like HashiCorp Vault. Understanding the importance of encrypting secrets in transit and at rest.

Evaluation:

1.MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as docker, kubernetes, git, openshift, chef and puppet

2.Live Evaluation: Industrial faculty / experts will conduct a live evaluation session where they will assess your understanding for project components, your ability to explain the implemented features and your problem-solving skills related to the project

3.Feedback: The industrial experts/faculty will provide feedback on your project implementation highlighting strengths and areas for improvement.

Use Case 9: Provisioning and Configuring a Web Server with Chef

Description:

Use Chef, an infrastructure automation tool, to provision and configure a web server. Set up a basic web server using Apache HTTP Server and configure it to serve a static website.

Tasks: **All the below tasks has to be completed using Agile Methodology.**

1. Set up the Chef Environment:

- Install and configure the Chef Workstation on your local machine.

- Set up a Chef Server or use the Chef Automate hosted service for managing infrastructure and storing configuration data.

2. Create a Cookbook:

- Create a new Chef cookbook chef command.

- Define the cookbook's structure, including recipes, attributes, templates, and files.

3. Write Recipes:

- Create a recipe within the cookbook to install and configure the Apache HTTP Server.

- Specify the necessary resources, such as packages, services, and configuration files, in the recipe.
- Define attributes to customize the web server configuration, such as port number, virtual hosts, or SSL settings.

4. Test the Cookbook Locally:

- Use Test Kitchen, a testing framework for Chef, to test the cookbook locally.

5. Upload the Cookbook to the Chef Server:

- Upload the cookbook to the Chef Server.
- Ensure the cookbook is available on the Chef Server for deployment to target nodes.

6. Set up the Target Node:

- Prepare the target node by installing the Chef client and configuring it to connect to the Chef Server.

- Configure the node's run-list to include the recipe for provisioning the web server.

7. Deploy the Cookbook:

- Use the Chef client to apply the cookbook to the target node.

- The Chef client will connect to the Chef Server, download the cookbook, and apply the defined resources on the node.

8. Verify the Web Server Configuration:

- Access the target node and verify that the Apache HTTP Server is installed and running.

- Check the web server's configuration files and ensure they match the desired settings defined in the cookbook.

Learning Outcome:

1. Understanding Infrastructure Automation: Solid understanding of infrastructure automation principles and the benefits of using tools like Chef to streamline server provisioning and configuration.

2. Proficiency in Chef Workflow: Chef workflow, including setting up the Chef Workstation, creating cookbooks, writing recipes, and managing infrastructure as code.

3. Configuration Management: How to define and manage configuration settings for web servers using Chef.

Evaluation:

1.MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as docker, kubernetes, git, openshift, chef and puppet

2.Live Evaluation: Industrial faculty / experts will conduct a live evaluation session where they will assess your understanding for project components, your ability to explain the implemented features and your problem-solving skills related to the project

3.Feedback: The industrial experts/faculty will provide feedback on your project implementation highlighting strengths and areas for improvement.

Use Case 10: Configure Nginx Web Server with Puppet

Description: Use Puppet, an infrastructure automation tool, to configure a Nginx web server. Define
Puppet manifests to install Nginx, manage its configuration files, and start the web server.

Tasks: **All the below tasks has to be completed using Agile Methodology.**

1. Set up the Puppet Environment:

- Install and configure the Puppet Agent on the target server where Nginx will be configured.
- Set up the Puppet Master server or use a hosted Puppet environment for managing the Puppet infrastructure.

2. Create Puppet Manifests:

- Create a Puppet module or class for managing the Nginx web server configuration.
- Define the necessary resources, such as packages, services, configuration files, and directories, within the Puppet manifests.

3. Install Nginx Package:

- Use the resource in Puppet to install the Nginx package on the target server.
- Specify the package name and version, ensuring that it matches the desired version.

4. Manage Nginx Configuration Files:

1. Create Resources in Puppet to manage Nginx configuration files.

2. Specify the desired content of the configuration files using Puppet's template or file content functionality.

5. Assign the Manifest to the Target Node:

1. Define a Puppet node definition or classification to assign the Nginx configuration manifest to the target node.

2. Specify the node's hostname or other identifying factors to apply the configuration to the correct target server.

6. Verify the Nginx Configuration:

1. Access the target server and verify that Nginx is installed and running.

2. Check the Nginx configuration files to ensure they match the desired settings defined in the Puppet manifest.

7. Manage Configuration Updates:

1. Make changes to the Puppet manifest as needed to update the Nginx configuration.

2. Deploy the updated manifest to the target server using the Puppet Agent to apply the changes.

Learning Outcomes:

1. Proficiency in Puppet Manifest Development: Proficiency in writing Puppet manifests, defining resources, and managing configuration files using Puppet's declarative language.

2. Infrastructure Configuration Management: Gain a solid understanding of infrastructure configuration management principles and best practices. They will learn how to use Puppet to ensure consistent and desired states of Nginx web servers across multiple nodes.

3. Nginx Configuration: Participants will become familiar with Nginx web server configuration.

4. Puppet Module Creation: Create and organize Puppet modules for managing specific components like Nginx.

5. Service Management with Puppet: Managing services using Puppet, specifically in ensuring the Nginx service is properly installed, enabled, and started.

Evaluation:

1.MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as docker, kubernetes, git, openshift, chef and puppet

2.Live Evaluation: Industrial faculty / experts will conduct a live evaluation session where they will assess your understanding for project components, your ability to explain the implemented features and your problem-solving skills related to the project

3.Feedback: The industrial experts/faculty will provide feedback on your project implementation highlighting strengths and areas for improvement.

Use Case 11:

Implement Firewall Rules with Puppet

Description:

Employ Puppet to implement firewall rules on a target server. Define Puppet manifests to install and configure a firewall management tool (e.g., iptables), define firewall rules, and ensure the firewall is enabled.

Tasks: **All the below tasks has to be completed using Agile Methodology.**

1. Set up the Puppet Environment:

- Install Puppet on the Puppet Master server and Puppet Agent on the target server where the firewall rules will be implemented.

- Configure the Puppet Master server to manage the Puppet Agent and define the Puppet environment.

2. Create the Puppet Manifests:

- Create a new Puppet module for managing firewall rules. This module will contain the necessary manifests and files for configuring the firewall.

- Inside the module, define a manifest file to describe the desired state of the firewall rules.

3. Install and Configure Firewall Management Tool:

- Define a package resource in the manifest to install the firewall management tool using the appropriate package manager for your operating system.
- Ensure the package is installed and any required dependencies are resolved.
- Configure the firewall management tool to start at system boot time.

4. Define Firewall Rules:

- Define firewall rules in the manifest using the syntax supported by the chosen firewall management tool.
- Specify the desired firewall rules, including allowed ports, protocols, IP addresses, or ranges.
- Ensure the rules are written in a way that follows best practices for security and compliance.

5. Enable the Firewall:

- Define a service resource in the manifest to manage the firewall service.
- Use Puppet's service management capabilities to enable and start the firewall service.

Learning Outcome:

1. Understanding Firewall Management: Understanding of firewall management principles and practices.
2. Proficiency in Puppet Manifest Development: Writing Puppet manifests for managing firewall rules.
3. Infrastructure Security: Practical knowledge of securing their infrastructure by implementing firewall rules.
4. Puppet Resource Types: Puppet resource types used for managing firewall rules, such as package, file, and service resources.
5. Compliance and Best Practices: Industry best practices and compliance standards for firewall configuration.

Evaluation:

1.MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as docker, kubernetes, git, openshift, chef and puppet

2.Live Evaluation: Industrial faculty / experts will conduct a live evaluation session where they will assess your understanding for project components, your ability to explain the implemented features and your problem-solving skills related to the project

3.Feedback: The industrial experts/faculty will provide feedback on your project implementation highlighting strengths and areas for improvement.

Use Case 12: Web Scraping and Data Analysis Automation

Description:

Create a DevOps pipeline to automate the web scraping and data analysis process using Python libraries such as Pandas, NumPy, and BeautifulSoup. The pipeline will consist of several stages, including data extraction, transformation, and analysis. Use Docker to containerize the application and ensure its portability and reproducibility.

Tasks: **All the below tasks has to be completed using Agile Methodology.**

1. Set up the Development Environment:

○ Install Python and the necessary libraries: Pandas, NumPy, and BeautifulSoup.

○ Set up a code editor or an integrated development environment (IDE) for Python development.

○ Create a project directory and initialize a Git repository for version control.

2. Web Scraping:

○ Use the BeautifulSoup library to extract data from a target website.

○ Write Python code to scrape the desired information from the web page.

○ Save the scraped data in a structured format, such as CSV or JSON.

3. Data Transformation and Cleaning:

○ Load the scraped data into a Pandas DataFrame for further processing.

○ Use Pandas functions to clean and transform the data, such as removing duplicates, handling missing values, or converting data types.

○ Perform any necessary data preprocessing tasks, such as feature engineering or data normalization.

4. Data Analysis:

○ Utilize NumPy and Pandas functions to perform data analysis and calculations.

○ Generate statistical summaries, calculate aggregations, or create visualizations to gain insights from the data.

○ Implement data analysis algorithms or machine learning models if relevant to the project.

5. Containerize the Application with Docker:

○ Write a Dockerfile to define the Docker image for your application.

○ Specify the necessary dependencies, including Python and the required libraries.

○ Set up the container environment and define the entry point for running the application.

6. Build and deploy the Docker Image:

- ○ Build the Docker image using the Dockerfile.

- ○ Push the Docker image to a container registry for version control and sharing.

- ○ Deploy the Docker image to a target environment, such as a local Docker engine or a cloud-based container platform.

7. Automation and CI/CD:

- ○ Set up a CI/CD pipeline using a continuous integration and continuous deployment tool, such as Jenkins or GitLab CI/CD.

- ○ Configure the pipeline to automatically trigger builds, tests, and deployments when changes are pushed to the repository.

- ○ Ensure the pipeline includes automated tests for the web scraping, data transformation, and analysis processes.

Learning Outcome:

1. Web Scraping Techniques: Hands-on experience in web scraping using BeautifulSoup. To extract data from websites, navigate HTML structures, and handle different types of web content, such as tables, forms, or dynamic elements.

2. Data Cleaning and Transformation: Develop skills in data cleaning and transformation using Pandas. How to load data into a Pandas DataFrame, handle missing values, remove duplicates, perform data type conversions, and apply various data manipulation techniques.

3. Data Analysis with NumPy and Pandas: leverage NumPy and Pandas for data analysis tasks.

4. Containerization with Docker: Containerizing their Python application using Docker.

5. Data Science Workflow: Exposure to the end-to-end data science workflow, from data extraction to analysis and visualization. Understand the importance of automating repetitive tasks and building scalable pipelines for data processing.

Evaluation:

1.MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as docker, kubernetes, git, openshift, chef and puppet

2.Live Evaluation: Industrial faculty / experts will conduct a live evaluation session where they will assess your understanding for project components, your ability to explain the implemented features and your problem-solving skills related to the project

3.Feedback: The industrial experts/faculty will provide feedback on your project implementation highlighting strengths and areas for improvement.

Use Case 13: Machine Learning Model Deployment with Flask and Docker

Description:

Develop a Flask web application to serve a trained machine learning model. Containerize the application with Docker for easy deployment. Implement RESTful APIs to accept input data, apply the model for predictions, and return the results.
Tasks: **All the below tasks has to be completed using Agile Methodology.**

1. Preparing the Machine Learning Model

- Train and finalize your machine learning model using Python libraries such as scikit-learn.

2. Building the Flask Application

- Create a new directory for your Flask application.
- Set up a virtual environment and install Flask and any other necessary dependencies.
- Create a Flask application script that will serve as the entry point for the application.
- Define the necessary routes and endpoints for handling incoming requests and executing predictions using the trained model.
- Load the serialized machine learning model into memory when the application starts.

3. Testing the Flask Application

- Run the Flask application locally to ensure that it is working correctly.
- Send test requests to the defined endpoints and verify that the model predictions are returned as expected.

4. Creating a Dockerfile

- Create a Dockerfile in the project directory.
- Define the base image that will be used for the Docker container, typically a lightweight Python image.
- Copy the necessary files, including the Flask application script and any required model files, into the Docker container.
- Install the required dependencies using pip or any other package manager.
- Specify the command that should be executed when the Docker container starts, which is typically running the Flask application.

5. Building the Docker Image

- Build the Docker image using the Dockerfile.
- Verify that the Docker image was successfully built by listing the available Docker images.

6. Running the Docker Container

- Run the Docker container locally.
- Specify any necessary port mappings if the Flask application is listening on a specific port.

- Test the deployed machine learning model by sending requests to the appropriate endpoint.

7. Deploying the Docker Image

- Push the Docker image to a container registry, such as Docker Hub or a private registry, for availability and sharing.

- Pull the Docker image from the container registry into the deployment environment.

- Configure the necessary network settings, environment variables, and other deployment configurations.
- Deploy the Docker container in the target environment and ensure it is running correctly.

Learning Outcome:

1. Flask Web Framework: Proficiency in using Flask, a popular Python web framework, for building web applications. How to define routes, handle HTTP requests, and implement the necessary endpoints for serving predictions from a machine learning model.
2. Docker Containerization: Containerize a Flask application using Docker.

3. Machine Learning Model Deployment: Hands-on experience in deploying a trained machine learning model within a Flask application.

Evaluation:

1.MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as docker, kubernetes, git, openshift, chef and puppet

2.Live Evaluation: Industrial faculty / experts will conduct a live evaluation session where they will assess your understanding for project components, your ability to explain the implemented features and your problem-solving skills related to the project

3.Feedback: The industrial experts/faculty will provide feedback on your project implementation highlighting strengths and areas for improvement.

Use Case 14: Microservices Application Deployment with OpenShift

Description:

Create a simple microservices-based application and deploy it using OpenShift, a container orchestration platform. The application will consist of multiple microservices that communicate with each other to perform different functionalities. Each microservice will be deployed as a separate container within the OpenShift cluster, allowing for scalability, resilience, and easy management.

Tasks: **All the below tasks has to be completed using Agile Methodology.**

1. Designing the Microservices Architecture

- Identify the different functionalities of your application and determine how to divide them into individual microservices.

- Define the APIs and communication patterns between the microservices.

2. Setting up an OpenShift Cluster

- Install and configure an OpenShift cluster either locally using Minishift or on a cloud provider like AWS or Azure.

- Set up authentication and access controls to manage cluster security.

3. Containerizing Microservices

- Containerize each microservice using Docker. Create Dockerfiles for each microservice that define the necessary dependencies and configurations.

- Build Docker images for each microservice and push them to a container registry accessible by your OpenShift cluster.

4. Defining OpenShift Deployment Configuration

- Define the necessary Kubernetes deployment manifests for each microservice. These manifests describe the desired state of the microservices, including container images, resource requirements, and network configurations.

- Create a deployment configuration for each microservice, specifying the desired number of replicas and any scaling policies.

5. Deploying Microservices on OpenShift

- Use the OpenShift command-line interface (CLI) or web console to create projects or namespaces to isolate your application.

- Deploy each microservice using the deployment configuration manifests. OpenShift will automatically create and manage the required containers within the cluster.

- Verify that the microservices are running correctly and can communicate with each other within the OpenShift cluster.

Learning Outcomes:

1. Understanding Microservices Architecture: Understanding of the principles and benefits of a microservices architecture, including modularity, scalability, and resilience.

2. Containerization and Docker: How to containerize microservices using Docker, including creating Dockerfiles, building images, and pushing them to a container registry.

3. OpenShift and Kubernetes: Proficient in using OpenShift, a Kubernetes-based container orchestration platform, for deploying and managing microservices. How to work with Kubernetes deployment manifests, pods, services, and scaling features.

4. Deployment and Scalability: Hands-on experience in deploying microservices as containers in an OpenShift cluster. Configure scaling policies and utilize load balancing to ensure high availability and performance.

Evaluation:

1.MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as docker, kubernetes, git, openshift, chef and puppet

2.Live Evaluation: Industrial faculty / experts will conduct a live evaluation session where they will assess your understanding for project components, your ability to explain the implemented features and your problem-solving skills related to the project

3.Feedback: The industrial experts/faculty will provide feedback on your project implementation highlighting strengths and areas for improvement.

Use Case 15: Version Control and Continuous Integration with Git and Docker

Description:

Create a version control system using Git and leverage its integration with Docker for continuous integration. Create a simple web application, manage its source code using Git, and automate the build and deployment process using Docker.

Tasks: **All the below tasks has to be completed using Agile Methodology.**

1. Setting up the Git Repository

- Create a new Git repository on a Git hosting platform such as GitHub.

- Clone the repository to your local development environment using the Git command-line interface.

2. Developing the Web Application

- Create a simple web application using your preferred programming language (e.g., Python, Node.js, Ruby).

- Initialize the project directory as a Git repository.

- Add the necessary source code files to the repository.

- Commit the initial codebase and provide a descriptive commit message.

3. Configuring Continuous Integration with Git Hooks

- Set up a Git hook, such as a post-receive hook, to trigger the CI process whenever new code is pushed to the repository.

- Write a script that executes the necessary commands to build and test the Docker image.

- Configure the Git hook to execute the script when changes are detected.

4. Setting up a CI Server

- Set up a CI server, such as Jenkins or GitLab CI/CD, that will handle the CI workflow.

- Configure the CI server to monitor the Git repository for changes and trigger the build process.

- Define a build pipeline or job that executes the necessary commands to build and test the Docker image.

5. Building and Testing the Docker Image

- Configure the CI server to execute the Docker build command, using the Dockerfile defined in the repository.

- Execute any necessary tests or validation checks on the Docker image to ensure its functionality.
- Generate relevant build artifacts, such as logs or reports, for further analysis.

6. Publishing the Docker Image

- Set up a container registry, such as Docker Hub or a private registry, to store the Docker images.

- Configure the CI server to push the built Docker image to the container registry.

- Tag the Docker image with an appropriate version or tag for easy identification and tracking.

7. Deploying the Docker Image

- Configure the deployment environment, such as a testing or production server, to pull ‘ the Docker image from the container registry.

- Set up a deployment pipeline or process to deploy the Docker image to the desired environment.

- Automate the deployment process to ensure a smooth and consistent deployment experience.

Learning Outcomes:

1. Git Version Control: Understanding of using Git for version control and collaborative development. How to create and manage repositories, work with branches, and handle code changes effectively.

2. Docker Containerization: Become proficient in using Docker for containerizing applications.

3. Continuous Integration (CI) Workflow: How to set up a CI workflow using Git hooks and a CI server.

4. CI Server Configuration: Gain hands-on experience in configuring a CI server, such as Jenkins or GitLab CI/CD.

Evaluation:

1.MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as docker, kubernetes, git, openshift, chef and puppet

2.Live Evaluation: Industrial faculty / experts will conduct a live evaluation session where they will assess your understanding for project components, your ability to explain the implemented features and your problem-solving skills related to the project

3.Feedback: The industrial experts/faculty will provide feedback on your project implementation highlighting strengths and areas for improvement.

Use Case 16: Managing Application Replicas with Kubernetes Replicase. Description:

Create and manage application replicas using Kubernetes ReplicaSet. ReplicaSet is a Kubernetes resource that ensures a specified number of replicas (pods) of a pod template are running at all times. Scale and manage application replicas effectively in a Kubernetes cluster.

Tasks: **All the below tasks has to be completed using Agile Methodology.**

1. Set Up the Kubernetes Cluster

- Install and set up a Kubernetes cluster using a platform like Minikube or a cloud provider's Kubernetes service.
- Verify the cluster's health and connectivity to ensure it is ready for deploying applications.
- Configure the command-line interface (CLI) tools, such as kubectl, to interact with the Kubernetes cluster.

2. Create a Pod Template

- Define a pod template specification in a YAML file that describes the desired configuration of the application pod.
- Specify the container image, ports, environment variables, and other necessary settings in the pod template.
- Ensure the pod template is ready for scaling by including appropriate resource limits and requests.

3. Create a ReplicaSet

- Write a ReplicaSet manifest in a YAML file, specifying the desired number of replicas and the pod template to use.
- Apply the ReplicaSet manifest to the Kubernetes cluster using the kubectl command.
- Verify that the ReplicaSet is created and the desired number of pods are running by checking the status of the pods and the ReplicaSet itself.

4. Scale the Application Replicas

- Update the ReplicaSet manifest to increase or decrease the desired number of replicas.
- Apply the updated ReplicaSet manifest to the cluster using the kubectl command.
- Observe the Kubernetes scheduler creating or terminating pods to maintain the desired number of replicas.

5. Perform Rolling Updates

- Modify the pod template specification to introduce changes or updates to the application, such as a new container image or environment variable.
- Apply the updated pod template to the existing ReplicaSet by modifying the ReplicaSet manifest or using the kubectl command.

- Observe the ReplicaSet orchestrating a rolling update, gradually replacing the old pods with the updated pods without causing downtime.

Learning Outcomes:

1. Kubernetes Fundamentals: Gain a solid understanding of key Kubernetes concepts, such as pods, ReplicaSets, scaling, and resource management.
2. Deployment and Scaling: Deploy applications using ReplicaSets and scale them up or down based on demand. Understand the mechanisms behind scaling and how Kubernetes automatically manages the creation and termination of pods to maintain the desired replica count.

3. YAML Manifests: Proficient in writing and working with YAML manifests to define pod templates and ReplicaSets.

4. Rolling Updates: Perform rolling updates on your application using ReplicaSets.

Evaluation:

1.MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as docker, kubernetes, git, openshift, chef and puppet

2.Live Evaluation: Industrial faculty / experts will conduct a live evaluation session where they will assess your understanding for project components, your ability to explain the implemented features and your problem-solving skills related to the project

3.Feedback: The industrial experts/faculty will provide feedback on your project implementation highlighting strengths and areas for improvement.

Use Case 17: Implementing Security Model in Chef

Description:

Project focuses on the security aspects of using Chef for configuration management. Implement a security model in Chef to ensure that the infrastructure and configurations are protected from unauthorized access and adhere to security best practices.

Tasks: **All the below tasks has to be completed using Agile Methodology.**

1. Infrastructure Hardening

- Identify security best practices and guidelines relevant to your infrastructure.

- Implement security configurations, such as firewall rules, access controls, and secure network configurations, using Chef resources.

- Apply the necessary security patches and updates to the infrastructure components using Chef recipes or cookbooks.

2. Secure Credential Management

- Implement secure credential management using Chef data bags or encrypted data bags.

- Store sensitive information, such as passwords or API keys, securely within the Chef environment.

- Ensure that only authorized users or nodes can access and decrypt the sensitive data.

3. Role-Based Access Control (RBAC)

- Define roles and permissions for different user types within the Chef server.

- Assign appropriate permissions to users or teams based on their responsibilities and access requirements.

- Implement RBAC policies using Chef's built-in authentication and authorization mechanisms.

4. Secure Cookbook Development

- Follow secure coding practices while developing Chef cookbooks, including input validation, secure file permissions, and avoiding hardcoded secrets.
- Use trusted sources for cookbook dependencies and ensure that they are regularly updated to address security vulnerabilities.

- Perform security testing and code reviews of the cookbooks to identify and mitigate potential security issues.

5. Continuous Compliance Monitoring

- Implement Chef InSpec to define and enforce compliance policies for your infrastructure.

- Write InSpec profiles to validate the security configurations and settings of the managed nodes.

- Run regular compliance checks and monitor the results to ensure continuous compliance with security standards.

6. Secure Communication and Data Encryption

- Configure secure communication channels between the Chef server and nodes using TLS/SSL certificates.

- Enable encryption for Chef data traffic to protect sensitive information during transmission.

- Regularly review and update the SSL certificates used for secure communication.

7. Auditing and Logging

- Enable auditing and logging features in Chef to capture relevant security events and activities.

- Configure log aggregation and analysis tools to monitor and detect any security-related incidents.

- Regularly review and analyze the logs to identify potential security threats or vulnerabilities.

Learning Outcomes

Security Best Practices: Solid understanding of security best practices and guidelines for securing infrastructure and configurations in a Chef environment.

1. Infrastructure Hardening: How to implement security configurations and measures to harden your infrastructure. This includes setting up firewalls, access controls, secure network configurations, and applying security patches and updates using Chef.
2. Secure Credential Management: Understand the importance of securely managing credentials and sensitive information within Chef. How to use Chef data bags or encrypted data bags to store and manage sensitive data and ensure that only authorized entities can access and decrypt it.

3. Role-Based Access Control (RBAC): How to define roles and permissions within Chef to enforce access controls and assign appropriate permissions to users or teams.

4. Secure Cookbook Development: Secure coding practices for developing Chef cookbooks. This includes input validation, secure file permissions, avoiding hard coded secrets, and following trusted sources for cookbook dependencies to address security vulnerabilities.

Evaluation:
1.MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as docker, kubernetes, git, openshift, chef and puppet

2.Live Evaluation: Industrial faculty / experts will conduct a live evaluation session where they will assess your understanding for project components, your ability to explain the implemented features and your problem-solving skills related to the project

3.Feedback: The industrial experts/faculty will provide feedback on your project implementation highlighting strengths and areas for improvement.

Use Case 18: Automated Deployment Script with Python

Description:

Create a Python automation script to streamline the deployment process of a web application. The script will handle tasks such as pulling code from a version control repository, building and packaging the application, and deploying it to a target environment. This project will help you understand how Python can be used to automate and simplify the deployment process in a DevOps context.

Tasks: **All the below tasks has to be completed using Agile Methodology.**

1. Set Up the Project Environment

- Create a new directory for your project and set up a virtual environment using a tool like virtualenv or conda.

- Install the necessary Python packages and dependencies for the deployment script, such as GitPython for interacting with the version control repository and any specific libraries required for your deployment process.

2. Clone the Source Code Repository

- Use the GitPython library to clone the source code repository of your web application.

- Provide the necessary authentication details, such as username and password or SSH key, if required.

3. Build and Package the Application

- Define the build process for your web application, including compiling assets, running tests, and generating the necessary artifacts.

- Use Python subprocess or similar libraries to execute the build commands and capture the output.

4. Deploy the Application

- Define the deployment process for your web application, such as copying files to the target server, configuring environment variables, and restarting services.

- Utilize SSH libraries or tools like Fabric to connect to the target server and execute the deployment commands.

Learning Outcomes:

1. Automation Skills: Hands-on experience in automating the deployment process using Python. Learn how to write efficient and reusable code to perform tasks such as cloning a repository, building and packaging an application, and deploying it to a target environment. This will enhance the automation skills and enable you to streamline repetitive tasks.

2. Version Control Integration: Understand how to interact with version control systems using Python.. This knowledge will be valuable for integrating your deployment process with version control tools, ensuring that the latest code is always deployed.

3. Build and Packaging Process: Gain knowledge of defining and executing build commands for your web application.

4. Deployment Techniques: Different deployment techniques, such as copying files, configuring environment variables, and restarting services.Gain experience in executing deployment commands remotely using SSH libraries or tools like Fabric.

Evaluation:

1.MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as docker, kubernetes, git, openshift, chef and puppet

2.Live Evaluation: Industrial faculty / experts will conduct a live evaluation session where they will assess your understanding for project components, your ability to explain the implemented features and your problem-solving skills related to the project

3.Feedback: The industrial experts/faculty will provide feedback on your project implementation highlighting strengths and areas for improvement.

Use Case 19: Dockerize Machine Learning Model Deployment with Python

Description:

Create a Dockerized machine learning model deployment solution using Python, Docker, and relevant machine learning libraries.

Tasks: **All the below tasks has to be completed using Agile Methodology.**

1. Set Up the Project Environment

- Create a new directory for your project and set up a virtual environment using a tool like virtualenv or conda.

- Install the necessary Python packages for developing and deploying your machine learning model, such as scikit-learn, pandas, and Flask.

2. Develop and Train the Machine Learning Model

- Write Python code to develop and train your machine learning model using the desired libraries and datasets.

- Evaluate and validate the model's performance using appropriate metrics and techniques.

3. Containerize the Machine Learning Model with Docker

- Create a Dockerfile that specifies the necessary dependencies and configuration for running your machine learning model.

- Build a Docker image based on the Dockerfile, which includes the required Python libraries, the trained model, and any other dependencies.

4. Develop the Web Service for Model Deployment

- Create a Flask application that exposes an API endpoint for making predictions using the trained machine learning model.

- Define the routes and request handling functions to accept input data, preprocess it if necessary, and return the predictions.

5. Deploy the Dockerized Model using Docker Compose

- Write a Docker Compose configuration file that describes the services needed for deployment, including the machine learning model container and any additional services required, such as a database or message broker.

- Use Docker Compose to deploy and orchestrate the containers, ensuring they communicate properly.

6. Scaling and Load Balancing

- Explore options for scaling the deployed model to handle increased traffic or workload. This could involve replicating containers, using orchestration tools like Kubernetes, or implementing load balancing strategies.

- Test the scalability and load balancing capabilities of the deployment solution to ensure it can handle increased demand.

7. Monitoring and Logging

- Set up monitoring and logging mechanisms to track the performance, usage, and errors of the deployed model.

- Implement appropriate logging practices and integrate with monitoring tools to gather insights into the model's behavior and identify potential issues.

Learning Outcome:

1. Docker Containerization: Experience in containerizing machine learning models using Docker. How to create a Dockerfile to specify the necessary dependencies and configurations for running your model.

2. DevOps Integration: How to integrate machine learning model deployment into a DevOps workflow. How to use Docker and Docker Compose to deploy and orchestrate containers, ensuring consistent and reproducible deployments.

3. Model Deployment as a Microservice: How to develop a Flask web service that exposes an API for making predictions using the deployed machine learning model.

Evaluation:

1.MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as docker, kubernetes, git, openshift, chef and puppet

2.Live Evaluation: Industrial faculty / experts will conduct a live evaluation session where they will assess your understanding for project components, your ability to explain the implemented features and your problem-solving skills related to the project

3.Feedback: The industrial experts/faculty will provide feedback on your project implementation highlighting strengths and areas for improvement.

Use Case 20: Deployment of an Application with Chef and Kubernetes

Description:

Deploy an application using Chef and Kubernetes. Use Chef to automate the provisioning and configuration of the application's infrastructure, and Kubernetes to manage the deployment and scaling of the application.

Tasks: **All the below tasks has to be completed using Agile Methodology.**

1. Set Up the Project Environment

- Install and configure Chef on your local machine.
- Set up a Kubernetes cluster locally using tools like Minikube or on a cloud provider.

2. Define Infrastructure as Code with Chef

- Create Chef recipes and cookbooks to define the desired state of the infrastructure required for the application.
- Write recipes to install and configure dependencies, such as databases, web servers, or load balancers.
- Use Chef's resource providers to manage infrastructure resources, such as creating
  Kubernetes deployments or services.

3. Create Kubernetes Manifests

- Define Kubernetes deployment manifests for the application, specifying the container image, environment variables, ports, and resource requirements.
- Write Kubernetes service manifests to expose the application to external traffic and enable communication between different application components.

- Utilize Kubernetes secrets for managing sensitive information, such as API keys or database credentials.

4. Automate Deployment with Chef

- Use Chef's orchestration capabilities to automate the deployment of the application to the
 Kubernetes cluster.

- Write Chef recipes to apply the Kubernetes manifests and configure the desired state of the cluster.
- Utilize Chef's idempotent nature to ensure that the application remains in the desired state even after updates or scaling events.

5. Test and Validate the Deployment

- Verify that the application is successfully deployed and accessible within the Kubernetes cluster.
 Test the functionality of the deployed application and validate its behavior under different scenarios, such as high traffic or scaling events.

6. Scaling and Maintenance

- Explore options for scaling the application based on traffic demands or resource utilization metrics.

- Utilize Chef and Kubernetes to scale the application horizontally by increasing the number of replicas or vertically by adjusting resource limits.

- Perform maintenance tasks, such as rolling updates or configuration changes, using Chef to ensure minimal downtime.

Learning Outcomes:

1. Infrastructure Automation with Chef: Gain hands-on experience in using Chef for infrastructure automation.

2. Kubernetes Deployment and Orchestration: Understand the fundamentals of Kubernetes and how to leverage its container orchestration capabilities. Leverage Kubernetes' features such as scaling, load balancing, and rolling updates to achieve efficient and resilient application deployments.

3. Integration of Chef and Kubernetes: How to integrate Chef and Kubernetes to streamline application deployments. Seamlessly combine configuration management with container orchestration, enabling efficient and consistent deployments.

4. Scalability and Maintenance: Explore scaling techniques in Kubernetes to handle increased traffic or resource demands.

5. DevOps and Collaboration: Experience working in a DevOps environment where infrastructure and application teams can collaborate effectively using common tools and practices.

Evaluation:

1.MCQ Questions: A set of multiple-choice questions covering the technologies and concepts used in the project, such as docker, Kubernetes, git, openshift, chef and puppet

2.Live Evaluation: Industrial faculty / experts will conduct a live evaluation session where they will assess your understanding for project components, your ability to explain the implemented features and your problem-solving skills related to the project

3.Feedback: The industrial experts/faculty will provide feedback on your project implementation highlighting strengths and areas for improvement.

SB8051	**FULL STACK DEVELOPMENT**	**L T P C**
		1 0 2 2

COURSE OBJECTIVE

The objective of this course is to provide conceptual understanding of how full stack development can be used to innovate and improve tech processes. This course will allow you to learn how to create full-stack web applications using frontend and backend JavaScript technologies. Learn both the theoretical and practical implementation of full stack web development solutions.

This Full Stack Java Developer Course is designed to give you the essence of front-end, middleware, and back-end Java web developer technologies. You will learn to build an end-to-end application, test and deploy code, store data, and much more.

COURSE CONTENT

UNIT I Software Development Process (Agile & Scrum) 3 + 6
Software Lifecycle Models - SRS (Software Requirement Specification) - SPS (Software Project Scheduling) - GITHub (End to End Journey) - Project Management Frameworks like Scrum, Agile etc.

UNIT II Introduction to Development Frameworks 3 + 6
Project Bucketing - Different Shades of Fullstack development - Angular.Js (Introduction) - Vue.Js (Introduction) - React.Js (Introduction) - React.Js - Components and Props - State and Lifecycle - Handling Events - Hooks & Forms

UNIT III Front End Development Framework & Tools 3 + 6
Web client server architecture - HTML Structure Tags – Forms field - Bootstrap – React.Js - jQuery- ajax - Promises - API integration

UNIT IV Back End Development Frameworks & Tools 3 + 6
API Development services - Express.Js -nNode.js – mongoDB - Manual Authentication using JWT, Passport.Js etc

UNIT V Full Stack Application Development and Deployment in Cloud

(Web, Mobile) **3 + 6**

Computer Networking - Basic Linux Commands - Cloud hosting using AWS - DevOps

TOTAL : 45 PERIODS

COURSE OUTCOME

- Planning and implementing the software lifecycle models to have a hands-on understanding with GIThub, agile & Scrum methodologies.
- Implementing the framework i.e react.js to align state & lifecycle, conditional rendering, Hooks etc.
- Inplementing the frontend technologies to build client architecture, to integrate APIs using react js.
- Implementing the tech stack like mongoDB, Express.Js & Node.js to build the back-end services.
- Hosting & deployment to ensure the microservices using DevOps, Cloud hosting using AWS.

FOR FURTHER READING:

- Full Stack Web Development for Beginners: Learn Ecommerce Web Development Using HTML5, CSS3, Bootstrap, JavaScript, MySQL, and PHP - By Riaz Ahmed
- A full stack data acquisition, archive and access solution for J-TEXT based on web technologies by Wei Zheng, Yuxing Wang Ming, Zhang Feiyang Wu, Zhou Yang
- Design and development of software stack of an autonomous vehicle using robot operating by Abhisek Omkar Prasada, Pradumn Mishraa, Urja Jaina, Anish Pandey, Anushka Sinhaa, Anil Singh Yadav, Rajan Kumar, Abhishek Sharma, Gaurav Kumar

REFERENCES:

1. The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer - Chris Northwood
2. Hands-On Full Stack Development with spring root 2 and react by Jua Hinkula
3. React and React Native, Third Edition by Adam Boduch, Roy Derks

ONLINE REFERENCES:

1. Beginner Full Stack Development Course: HTML, CSS, React and Node https://www.udemy.com/course/ultimate-web/
2. Full Stack Web Developer: MEAN Stack https://intellipaat.com/full-stack-web-developer-mean-stack-certification-training/

SOFTWARE REQUIREMENT:

Javascript, HTMS, CSS, Bootstrap, JQuery, Node Js, MongoDB, SQLite

INDUSTRY SCOPE:

By completion of this course, the students can get industry scope in website development, frontend and backend development across industries and domains.

20 INDUSTRY USE CASES

1. E-Commerce Website

An e-commerce website or app is one of the best full-stack development projects you can practice at the advanced level. It is a vast project involving front and back-end technologies and database knowledge for efficient execution and implementation. The project takes time and effort because you must maintain a huge database with multiple product categories and prices.

Learning outcome:

- Develop a full stack-based system that can offer high-end webpage through the use of front-end techniques, as well as the visibility to easily react and verify the accuracy of the Items selected.
- Develop a full stack-based system that can offer Improved accessibility and can make it easier for people to select the articles, especially for those who may have difficulty physically going to market to purchase some.
- Develop a full stack-based system that can offer enhanced transparency and can make the user journey more transparent, as all registered items are recorded and verified on a publicly-available ledger.
- Develop a full stack-based system that can offer greater efficiency and can streamline the buyer seller Journey and reduce the time and resources required to complete the process
- Develop a full stack-based system that can offer enhanced accuracy and can reduce the risk of faults and errors in the Journey, as the registration is recorded and verified electronically.

Task1: Develop the framework for aligning the E-commerce web page at one place with proper categorisation with the help of development scripting.

Task 2: Enable the data with the customer manifestation to enroute the requisite web development with the help of suitable programming language.

Task 3: Design the web server to engage with the operating system user is using.

Task 4: Develop a system which showcases the inserted result on the ground of inputs user made over the webpage.

Task 5: Implement industry specific hosting using cloud at AWS, to ensure the microservices functionality and deployment.

2. Video Conferencing Website and Application

You can try your hands on the video conferencing website and application projects and help develop features like textual chatting, audio-video interactions, video recording, etc. The project requires you to implement your ideas and ensure a creative application or website.

Learning outcome:

- Develop a full stack-based system that can offer high-end webpage through the use of front-end techniques, as well as the visibility to easily react and verify the accuracy of the time-zones according to the countries.
- Develop a full stack-based system that can offer Improved accessibility and can make it easier for people to plan their meetings, especially for those who may have difficulty physically going to meet them in person.
- Develop a full stack-based system that can offer enhanced transparency and can make the user journey more transparent, as all registered users are recorded and verified on a publicly-available ledger.
- Develop a full stack-based system that can offer greater efficiency and can streamline the caller & receiver Journey and reduce the time and resources required to complete the process.
- Develop a full stack-based system that can offer enhanced accuracy and can reduce the risk of faults and errors in the Journey, as registration is verified electronically.

Task1: Develop the framework for creating the video conferencing application/website with the project management tools like JIRA & Trello.

Task 2: Enable the data with the system control of version based servers like GIT & Grunt .

Task 3: Design the database server to resemble the implementation of MongoDb, MySQL etc.

Task 4: Develop a frontend framework which showcases the inserted result Jquery, Javascript and HTML.

Task 5: Implement industry specific hosting using cloud at AWS, to ensure the microservices functionality and deployment.
3. Social Media Website and Application

The present-day world revolves around social media, and we already use many websites and applications. You can try your hands on an efficient full-stack project and develop a social media app or website with unique features to attract users.

Learning outcome:

- Develop a full stack-based system that can offer high-end webpage through the use of front-end techniques, as well as the visibility to easily react and verify the legitimacy of the users onboarded
- Develop a full stack-based system that can offer Improved accessibility and can make it easier for people to add, remove, post etc. , especially for those who may have difficulty expressing their well-being in terms of words.
- Develop a full stack-based system that can offer enhanced transparency and can make the user journey more transparent, as all registered people are recorded and verified on a publicly-available ledger.
- Develop a full stack-based system that can offer greater efficiency and can streamline the mutual interaction between two or more individuals and reduce the time and resources required to complete the process.
- Develop a full stack-based system that can offer enhanced accuracy and can reduce the risk of faults and errors in the Journey, as the registration is recorded and verified electronically.

Task1: Develop the framework for creating the social media application/website with the proper implementation of web technologies.

Task 2: Enable and enhance the data with the system control of version based servers i.e MySql & others

Task 3: Build and Nourish the database server to resemble the implementation of NoSql.

Task 4: Design a unique frontend framework which showcases the valued results with the use of Express JS & IONIC

Task 5: Implement industry specific hosting using cloud at AWS, to ensure the microservices functionality and deployment.

4. Content Management Tool and System

Content Management Tool or System is a significant factor that plays a crucial role in creating blogs. You can use the full stack development items and other drag-and-drop interfaces to create a web page that helps you add text, images, videos, and other elements required to create a blog.

Learning outcome:

- Develop a full stack-based system that can offer high-end webpage through the use of front-end techniques, as well as the visibility to easily react and verify the accuracy of the articles written.
- Develop a full stack-based system that can offer Improved accessibility and can make it easier for people to select the articles, especially for those who may have difficulty physically going to market to purchase books.
- Develop a full stack-based system that can offer enhanced transparency and can make the user journey more enriching, as all the written articles are recorded and verified on a publicly-available ledger.
- Develop a full stack-based system that can offer greater efficiency and can streamline the traction and reduce the time and resources required to complete the process.
- Develop a full stack-based system that can offer enhanced accuracy and can reduce the risk of faults and errors in the Journey, as the registration is recorded and verified electronically.

Task1: Develop the framework for creating the content management website using the tools like Java & kotlin

Task 2: Enable the data with the system control of version based servers like SQL

Task 3: Design the database server to resemble the implementation of MongoDb, SQLite & NOSql etc.
Task 4: Develop a frontend framework which showcases the inserted result with Express JS & IONIC

Task 5: Implement industry specific hosting using cloud at AWS, to ensure the microservices functionality and deployment.

5. Project Management Tool

A project management tool is one of the toughest full-stack projects and requires many features for proper execution. The project includes a framework of social media sites where users can communicate with one another. The users also get the functionality to assign tasks to others and comment on the dashboard at their convenience.

Learning outcome:

- Develop a full stack-based system that can offer high-end webpage through the use of front-end techniques, as well as the visibility to easily react and verify the basics of the projects and their guidelines.
- Develop a full stack-based system that can offer Improved accessibility and can make it easier for users to select the project, especially for those who may have difficulty physically going on-site or doing remote work.
- Develop a full stack-based system that can offer enhanced transparency and can make the user journey more transparent, as all registered users are recorded and verified on a publicly-available ledger.
- Develop a full stack-based system that can offer greater efficiency and can streamline the user & UI Interaction and reduce the time and resources required to complete the process.
- Develop a full stack-based system that can offer enhanced accuracy and can reduce the risk of faults and errors in the Journey, as the registration is recorded and verified electronically.

Task1: Develop the framework for creating the project management application/website with the project management tools like Java & kotlin

Task 2: Enable the data with the system control of version based servers like SQL

Task 3: Design the database server to resemble the implementation of MongoDb, SQLite.

Task 4: Develop a frontend framework which showcases the inserted result with IONIC

Task 5: Implement industry specific hosting using cloud at AWS, to ensure the microservices functionality and deployment.

6. To-Do List Projects

A to-do list is one of the simplest and best full-stack projects for beginners. You can create a to-do list efficiently with the feature to add items to the list. You can also provide functionalities to move content from the list after completing the task. The project will help you improve your front-end development skills and learn crucial database operations.

Learning outcome:

- Develop a full stack-based system that can offer high-end webpage through the use of front-end techniques, as well as the visibility to easily react and verify the time & task aligned.
- Develop a full stack-based system that can offer Improved accessibility and can make it easier for people to select their time & manually design their planner especially for those who usually are not able to manage their time efficiently.
- Develop a full stack-based system that can offer enhanced transparency and can make the user journey more transparent, as all registered users are recorded and verified on a publicly-available ledger.
- Develop a full stack-based system that can offer greater efficiency and can streamline the day to day activities of the user and reduce the time and resources required to complete the process.
- Develop a full stack-based system that can offer enhanced accuracy and can reduce the risk of faults and errors in the Journey, as the registration is recorded and verified electronically.

Task1: Develop the framework for creating the To-Do List Project management application/website with the project management tools like Java & kotlin

Task 2: Enable the data with the system control of version based servers like SQL

Task 3: Design the database server to resemble the implementation of MongoDb, SQLite for creating the To-Do List Project.

Task 4: Develop a frontend framework which showcases the inserted result with IONIC for creating the To-Do List Project.

Task 5: Implement industry specific hosting using cloud at AWS, to ensure the microservices functionality and deployment.

7. Chat Application and Website

Everyone chats with their family members and friends daily, and because of this, chatting applications have become an inevitable part of our lives. Chatting apps are one of the best full-stack beginner projects. You can create a chat application or website to serve the purpose of individual or group chats between networks.

Learning outcome:

- Develop a full stack-based system that can offer high-end webpage through the use of front-end techniques, as well as the visibility to easily react and reply instantly.
- Develop a full stack-based system that can offer Improved accessibility and can make it easier for people to chat & connect especially for those who may have difficulty physically connecting with people.
- Develop a full stack-based system that can offer enhanced transparency and can make the user journey more transparent, as all registered users are recorded and verified on a publicly-available ledger.
- Develop a full stack-based system that can offer greater efficiency and can streamline the users Journey and reduce the time and resources required to complete the process.
- Develop a full stack-based system that can offer enhanced accuracy and can reduce the risk of faults and errors in the Journey, as the registration is recorded and verified electronically.

Task1: Develop the framework for creating the Chat application/website with the project management tools like Java & kotlin

Task 2: Enable the data with the system control of version based servers like SQL for creating the Chat application/website.

Task 3: Design the database server to resemble the implementation of MongoDb, SQLite for creating the Chat application/website.

Task 4: Develop a frontend framework which showcases the inserted result with IONIC for creating the Chat application/website.

Task 5: Implement industry specific hosting using cloud at AWS, to ensure the microservices functionality and deployment.

8. Portfolio Website

Developers build portfolio websites as full-stack developer sample projects to showcase their skills and impress clients. As a student or professional learning web development, you must practice making portfolio websites to gain knowledge and experience in efficient front-end web development technology.

Learning outcome:

- Develop a full stack-based system that can offer high-end webpage through the use of front-end techniques, as well as the visibility to easily react and verify the accuracy of the portfolios selected.
- Develop a full stack-based system that can offer Improved accessibility and can make it easier for people to select the portfolios.
- Develop a full stack-based system that can offer enhanced transparency and can make the user journey more transparent, as all the created portfolios are recorded and verified on a publicly-available ledger.
- Develop a full stack-based system that can offer greater efficiency and can streamline the buyer seller Journey and reduce the time and resources required to complete the process.
- Develop a full stack-based system that can offer enhanced accuracy and can reduce the risk of faults and errors in the Journey, as the registration is recorded and verified electronically.

Task1: Develop the framework for creating the portfolio website with the project management tools like Java & kotlin

Task 2: Enable the data with the system control of version based servers like SQL

Task 3: Design the database server to resemble the implementation of MongoDb, SQLite for creating the portfolio website.

Task 4: Develop a frontend framework which showcases the inserted result with IONIC for creating the portfolio website.

Task 5: Implement industry specific hosting using cloud at AWS, to ensure the microservices functionality and deployment.

9. Blog Website and Application

A blog website allows users to opine their thoughts and comment on anything and everything. You can make a creative website or application using full-stack technology and take inspiration from other blogs. Ensure that you add proper authentication features so individuals with valid credentials can log in to the platform.

Learning outcome:

- Develop a full stack-based system that can offer high-end webpage through the use of front-end techniques, as well as the visibility to easily react and verify the accuracy of the articles written.
- Develop a full stack-based system that can offer Improved accessibility and can make it easier for people to select the articles, especially for those who may have difficulty physically going to market to purchase books.
- Develop a full stack-based system that can offer enhanced transparency and can make the user journey more enriching, as all the written articles are recorded and verified on a publicly-available ledger.
- Develop a full stack-based system that can offer greater efficiency and can streamline the traction and reduce the time and resources required to complete the process.
- Develop a full stack-based system that can offer enhanced accuracy and can reduce the risk of faults and errors in the Journey, as the registration is recorded and verified electronically.

Task1: Develop the framework for creating the blogging website with the project management tools like Java & kotlin

Task 2: Enable the data with the system control of version based servers like SQL for creating the blogging website.

Task 3: Design the database server to resemble the implementation of MongoDb, SQLite for creating the blogging website.

Task 4: Develop a frontend framework which showcases the inserted result with IONIC for creating the blogging website.

T**ask 5:** Implement industry specific hosting using cloud at AWS, to ensure the microservices functionality and deployment.

10. Application For Grocery Delivery

A grocery delivery website or application is among the best projects for full-stack developers to showcase their talent and skills. You can learn multiple factors related to large-scale full-stack apps or websites by making the grocery delivery application project. It involves back-end knowledge and helps you excel in making major projects in the future.

Learning outcome:

- Develop a full stack-based system that can offer high-end webpage through the use of front-end techniques, as well as the visibility to easily select and verify the accuracy of the Items selected.
- Develop a full stack-based system that can offer Improved accessibility and can make it easier for people to select the articles, especially for those who may have difficulty physically going to market to purchase some.
- Develop a full stack-based system that can offer enhanced transparency and can make the user journey more transparent, as all registered items are recorded and verified on a publicly-available ledger.
- Develop a full stack-based system that can offer greater efficiency and can streamline the buyer seller Journey and reduce the time and resources required to complete the process.
- Develop a full stack-based system that can offer enhanced accuracy and can reduce the risk of faults and errors in the Journey, as the registration is recorded and verified electronically.

Task1: Develop the framework for creating the grocery delivery application with the project management tools like Java & kotlin

Task 2: Enable the data with the system control of version based servers like SQL for creating the grocery delivery application.

Task 3: Design the database server to resemble the implementation of MongoDb, SQLite for creating the grocery delivery application.

Task 4: Develop a frontend framework which showcases the inserted result with IONIC for creating the grocery delivery application.

Task 5: Implement industry specific hosting using cloud at AWS, to ensure the microservices functionality and deployment.

11. Food Delivery Website and Application

A food delivery website or application is slightly different from grocery delivery. However, it also works on the same grounds as the latter and may require full-stack development. You can practice and improve your front- and back-end development skills when making food delivery websites or applications.

Learning outcome:

- Develop a full stack-based system that can offer high-end webpage through the use of front-end techniques, as well as the visibility to easily react and verify the accuracy of the Items selected.
- Develop a full stack-based system that can offer Improved accessibility and can make it easier for people to select the dishes, especially for those who may have difficulty physically going to market to purchase some.
- Develop a full stack-based system that can offer enhanced transparency and can make the user journey more transparent, as all registered items are recorded and verified on a publicly-available ledger.
- Develop a full stack-based system that can offer greater efficiency and can streamline the buyer seller Journey and reduce the time and resources required to complete the process.
- Develop a full stack-based system that can offer enhanced accuracy and can reduce the risk of faults and errors in the Journey, as the registration is recorded and verified electronically.

Task1: Develop the framework for creating the food delivery application with the project management tools like Java & kotlin

Task 2: Enable the data with the system control of version based servers like SQL

Task 3: Design the database server to resemble the implementation of MongoDb, SQLite for creating the food delivery application.

Task 4: Develop a frontend framework which showcases the inserted result with IONIC for creating the food delivery application.

Task 5: Implement industry specific hosting using cloud at AWS, to ensure the microservices functionality and deployment.

12. Workout Tracker Application

A workout tracking website or application is slightly different from a to-do project framework. However, it also works on the same grounds as the latter and may require full-stack development.

You can practice and improve your front- and back-end development skills when making workout tracking applications.

Learning outcome:

- Develop a full stack-based system that can offer high-end webpage through the use of front-end techniques, as well as the visibility to easily react and verify the accuracy of the inputs made.
- Develop a full stack-based system that can offer Improved accessibility and can make it easier for people to select the workout style especially for those who may have difficulty to track their workout routine.
- Develop a full stack-based system that can offer enhanced transparency and can make the user journey more transparent, as all workout style & patterns are recorded and verified on a publicly-available ledger.
- Develop a full stack-based system that can offer greater efficiency and can streamline the UI & user Interaction and reduce the time and resources required to complete the process.
- Develop a full stack-based system that can offer enhanced accuracy and can reduce the risk of faults and errors in the Journey, as the registration is recorded and verified electronically.

Task1: Develop the framework for creating the workout tracking application with the project management tools like Java & kotlin

Task 2: Enable the data with the system control of version based servers like SQL for creating the workout tracker application

Task 3: Design the database server to resemble the implementation of MongoDb, SQLite for creating the workout tracker application

Task 4: Develop a frontend framework which showcases the inserted result with IONIC for creating the workout tracker application.

Task 5: Implement industry specific hosting using cloud at AWS, to ensure the microservices functionality and deployment.

13. Calendar Scheduler Application

A calendar scheduler application is slightly different from a to-do project framework. However, it also works on the same grounds as the latter and may require full-stack development. You can practice and improve your front- and back-end development skills when making calendar scheduler applications.

Learning outcome:

- Develop a full stack-based system that can offer high-end webpage through the use of front-end techniques, as well as the visibility to easily react and verify the time & task aligned.
- Develop a full stack-based system that can offer Improved accessibility and can make it easier for people to select their time & manually design their planner especially for those who usually are not able to manage their time efficiently.
- Develop a full stack-based system that can offer enhanced transparency and can make the user journey more transparent, as all registered users are recorded and verified on a publicly-available ledger.
- Develop a full stack-based system that can offer greater efficiency and can streamline the day to day activities of the user and reduce the time and resources required to complete the process.
- Develop a full stack-based system that can offer enhanced accuracy and can reduce the risk of faults and errors in the Journey, as the registration is recorded and verified electronically.

Task1: Develop the framework for creating the calendar scheduler application with the project management tools like Java & kotlin.

Task 2: Enable the data with the system control of version based servers like SQL for creating the calendar scheduler application

Task 3: Design the database server to resemble the implementation of MongoDb, SQLite for creating the calendar scheduler application.

Task 4: Develop a frontend framework which showcases the inserted result with IONIC for creating the calendar scheduler application.

Task 5: Implement industry specific hosting using cloud at AWS, to ensure the microservices functionality and deployment.

14. Money Transfer Application

A Money Transfer application is slightly different from a chat project framework. However, it also works on the same grounds as the latter and may require full-stack development. You can practice and improve your front- and back-end development skills when making money transfer applications.

Learning outcome:

- Develop a full stack-based system that can offer high-end webpage through the use of front-end techniques, as well as the visibility to easily check and verify the accuracy of the payments made.
- Develop a full stack-based system that can offer Improved accessibility and can make it easier for people to make the transactions, especially for those who may have difficulty physically going in-person to do the needful.
- Develop a full stack-based system that can offer enhanced transparency and can make the user journey more transparent, as all registered users are recorded and verified on a publicly-available ledger.
- Develop a full stack-based system that can offer greater efficiency and can streamline the Journey between receiver sender and the bank and reduce the time and resources required to complete the process.
- Develop a full stack-based system that can offer enhanced accuracy and can reduce the risk of faults and errors in the Journey, as the registration is recorded and verified electronically.

Task1: Develop the framework for creating the money transfer application with the project management tools like Java & kotlin

Task 2: Enable the data with the system control of version based servers like SQL for creating the money transfer application

Task 3: Design the database server to resemble the implementation of MongoDb, SQLite for creating the money transfer application.

Task 4: Develop a frontend framework which showcases the inserted result with IONIC for creating the money transfer application.

Task 5: Implement industry specific hosting using cloud at AWS, to ensure the microservices functionality and deployment.

15. Internet Banking Application and Website

An Internet Banking application and website is slightly different from a money transfer project framework. However, it also works on the same grounds as the latter and may require full-stack development. You can practice and improve your front- and back-end development skills when making internet banking applications and websites.

Learning outcome:

- Develop a full stack-based system that can offer high-end webpage through the use of front-end techniques, as well as the visibility to easily check and verify the accuracy of the payments made.
- Develop a full stack-based system that can offer Improved accessibility and can make it easier for people to make the transactions, especially for those who may have difficulty physically going in-person to do the needful.
- Develop a full stack-based system that can offer enhanced transparency and can make the user journey more transparent, as all registered users are recorded and verified on a publicly-available ledger.
- Develop a full stack-based system that can offer greater efficiency and can streamline the Journey between receiver sender and the bank and reduce the time and resources required to complete the process.
- Develop a full stack-based system that can offer enhanced accuracy and can reduce the risk of faults and errors in the Journey, as the registration is recorded and verified electronically.

Task1: Develop the framework for creating the internet banking application with the project management tools like Java & kotlin for Internet banking application

Task 2: Enable the data with the system control of version based servers like SQL

Task 3: Design the database server to resemble the implementation of MongoDb, SQLite for Internet banking application.

Task 4: Develop a frontend framework which showcases the inserted result with IONIC for Internet banking application.

Task 5: Implement industry specific hosting using cloud at AWS, to ensure the microservices functionality and deployment.
16. Astrology Application and Website

An Astrology application and website is slightly different from a social media project framework. However, it also works on the same grounds as the latter and may require full-stack development. You can practice and improve your front- and back-end development skills when making astrology applications and websites.

Learning outcome:

- Develop a full stack-based system that can offer high-end webpage through the use of front-end techniques, as well as the visibility to easily react and verify the predictions made.
- Develop a full stack-based system that can offer Improved accessibility and can make it easier for people to select the sunshine.
- Develop a full stack-based system that can offer enhanced transparency and can make the user journey more transparent, as all registered users are recorded and verified on a publicly-available ledger.
- Develop a full stack-based system that can offer greater efficiency and can streamline the UI & user Interaction and reduce the time and resources required to complete the process.
- Develop a full stack-based system that can offer enhanced accuracy and can reduce the risk of faults and errors in the Journey, as the registration is recorded and verified electronically.

Task1: Develop the framework for creating the astrology application with the project management tools like Java & kotlin

Task 2: Enable the data with the system control of version based servers like SQL for creating astrology application

Task 3: Design the database server to resemble the implementation of MongoDb, SQLite for creating astrology applications.

Task 4: Develop a frontend framework which showcases the inserted result with IONIC

Task 5: Implement industry specific hosting using cloud at AWS, to ensure the microservices functionality and deployment.

17. Horoscope Application and Website

A Horoscope application and website is slightly different from a social media project framework. However, it also works on the same grounds as the latter and may require full-stack development. You can practice and improve your front- and back-end development skills when making horoscope applications and websites.

Learning outcome:

- Develop a full stack-based system that can offer high-end webpage through the use of front-end techniques, as well as the visibility to easily react and verify the predictions made.
- Develop a full stack-based system that can offer Improved accessibility and can make it easier for people to select the sunshine.
- Develop a full stack-based system that can offer enhanced transparency and can make the user journey more transparent, as all registered users are recorded and verified on a publicly-available ledger.
- Develop a full stack-based system that can offer greater efficiency and can streamline the UI & user Interaction and reduce the time and resources required to complete the process.
- Develop a full stack-based system that can offer enhanced accuracy and can reduce the risk of faults and errors in the Journey, as the registration is recorded and verified electronically.

Task 1: Develop the framework for creating the horoscope application with the project management tools like Java & kotlin

Task 2: Enable the data with the system control of version based servers like SQL for horoscope application and website
Task 3: Design the database server to resemble the implementation of MongoDb, SQLite for horoscope application and website

Task 4: Develop a frontend framework which showcases the inserted result with IONIC for horoscope application and website

Task 5: Implement industry specific hosting using cloud at AWS, to ensure the microservices functionality and deployment.

18. Stock Inventory Application

Stock Inventory Tool or System is a significant factor that plays a crucial role in managing the stock reviewed. You can use the full stack development items and other drag-and-drop interfaces to create a web page that helps you add text, images, videos, and other elements required to create an Inventory.

Learning outcome:

- Develop a full stack-based system that can offer high-end webpage through the use of front-end techniques, as well as the visibility to easily react and verify the accuracy of the items selected.

- Develop a full stack-based system that can offer Improved accessibility and can make it easier for people to select the articles, especially for those who may have difficulty physically maintaining their sale & purchase some.
- Develop a full stack-based system that can offer enhanced transparency and can make the user journey more transparent, as all registered items are recorded and verified on a publicly-available ledger.
- Develop a full stack-based system that can offer greater efficiency and can streamline the Inventory & user Journey and reduce the time and resources required to complete the process.
- Develop a full stack-based system that can offer enhanced accuracy and can reduce the risk of faults and errors in the Journey, as the registration is recorded and verified electronically.

Task 1: Develop the framework for creating the Stock Inventory application with the project management tools like Java & kotlin

Task 2: Enable the data with the system control of version based servers like SQL

Task 3: Design the database server to resemble the implementation of MongoDb, SQLite for stock inventory application

Task 4: Develop a frontend framework which showcases the inserted result with IONIC for developing the application

Task 5: Implement industry specific data modeling frameworks to enhance the backend specification like Ruby on rails, Laravel for stock inventory application

19. News Media Application

News Media application is a significant factor that plays a crucial role in keeping yourself updated. You can use the full stack development items and other drag-and-drop interfaces to create a web page that helps you add text, images, videos, and other elements required to create a news media application.

Learning outcome:

- Develop a full stack-based system that can offer high-end webpage through the use of front-end techniques, as well as the visibility to easily react and verify the legitimacy of the users onboarded
- Develop a full stack-based system that can offer Improved accessibility and can make it easier for people to add, remove, post etc. , especially for those who may have difficulty expressing their well-being in terms of words.
- Develop a full stack-based system that can offer enhanced transparency and can make the user journey more transparent, as all registered people are recorded and verified on a publicly-available ledger.
- Develop a full stack-based system that can offer greater efficiency and can streamline the mutual interaction between two or more individuals and reduce the time and resources required to complete the process.
- Develop a full stack-based system that can offer enhanced accuracy and can reduce the risk of faults and errors in the Journey, as the registration is recorded and verified electronically.

Task 1: Develop the framework for creating the news media application with the project management tools like Java & kotlin for developing the application

Task 2: Enable the data with the system control of version based servers like SQL

Task 3: Design the database server to resemble the implementation of MongoDb, SQLite for the application

Task 4: Develop a frontend framework which showcases the inserted result with IONIC for developing the application

Task 5: Implement industry specific hosting using cloud at AWS, to ensure the microservices functionality and deployment.

20. Job Search Application

Job Search application is a significant factor that plays a crucial role in keeping yourself updated. You can use the full stack development items and other drag-and-drop interfaces to create a web page that helps you add text, images, videos, and other elements required to create a job search portal

Learning outcome:

- Develop a full stack-based system that can offer high-end webpage through the use of front-end techniques, as well as the visibility to easily post and verify the accuracy of the candidates.
- Develop a full stack-based system that can offer Improved accessibility and can make it easier for people to apply for the jobs, especially for those who may have difficulty physically hustling for applying resumes.
- Develop a full stack-based system that can offer enhanced transparency and can make the user journey more transparent, as all registered users are recorded and verified on a publicly-available ledger.
- Develop a full stack-based system that can offer greater efficiency and can streamline the candidate and recruiter journey and reduce the time and resources required to complete the process.
- Develop a full stack-based system that can offer enhanced accuracy and can reduce the risk of faults and errors in the Journey, as the registration is recorded and verified electronically.

Task 1: Develop the framework for creating the job search portal with the project management tools like Java & kotlin

Task 2: Enable the data with the system control of version based servers like SQL for developing job search application

Task 3: Design the database server to resemble the implementation of MongoDb, SQLite.

Task 4: Develop a frontend framework which showcases the inserted result with IONIC for the application

Task 5: Implement industry specific hosting using cloud at AWS, to ensure the microservices functionality and deployment.

SB8052	ARTIFICIAL INTELLIGENCE AND APPLICATIONS	L	T	P	C
		1	0	2	2

COURSE OBJECTIVE

The objective of this course is to provide an insight on various tools and concepts of Artificial Intelligence. Learners will not only learn the theory but will also be able to relate to real world implementation of the same.

UNIT-I INTRODUCTION TO ARTIFICIAL INTELLIGENCE 6 hours

Prelude - Why AI? - What is AI? - AI in Practice - AI in Business - AI Platforms - Conclusion – Self Assessment

UNIT-II INTRODUCTION TO DATA SCIENCE 6 hours

Prelude - Why Deep Learning? - Deep Learning - Evolution and Business Potential -

Introduction to Artificial Neural Network - What is Deep Learning? - Convolution Neural – Networks Recurrent Neural Networks – Autoencoders - Deep Learning & Business - Deep learning Frameworks and Products - Course Summary - Self-Assessment - Deep Learning

UNIT-III COMPUTER VISION 101 & ETHICAL AI 10 hours

Prelude - Why Computer Vision? - What is Computer Vision? - Concepts and Techniques in Computer Vision - Computer Vision Tools and Platforms – Summary - Assessment - Computer Vision 101. Ethical AI: Prelude - AI and Ethics - Guidelines, Regulations & Standards for Ethical AI - Building Blocks of Responsible AI - Core Ethical Requirements of AI Systems – Conclusion - Self-Assessment: Ethical AI

UNIT-IV PROMPT ENGINEERING 8 hours

Primer – Prelude - Generative Pre-trained Models -Large Language Models - Need for Prompt Engineering - Prompt Engineering Guidelines - Text Completion - Code Completion - Troubleshooting - Limitations of Large Language Models – Summary - Self Assessment

UNIT-V AI-FIRST SOFTWARE ENGINEERING & GENERATIVE MODELS FOR DEVELOPERS 15 hours

Primer – Prelude - AI First – a paradigm shift - AI-First Interpretations - AI First SDLC Phases - Limitations of Large Language Models - Course Summary -Self-Assessment. Generative Models for Developers: Prelude - Introduction to Generative models - Generative Adversarial Network -GAN - Deep Convolutional GAN - Additional GAN architectures - Challenges and performance issues in GAN models – Summary – Project - Self-Assessment

TOTAL : 45 PERIODS

COURSE OUTCOMES

On completion of the course, students will be able to:

CO1 : Demonstrate fundamental understanding of the history of artificial intelligence (AI) and its foundations.

CO2 : Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning.

CO3 : Assess and select appropriate data analysis models for solving real-world problem.

CO4 : Demonstrate the importance of data visualization, design, and use of visual components.

FOR FURTHER READING

1 Saroj Kaushik, Artificial Intelligence, Cengage learning, 2014
2 Elaine Rich, Kevin Knight, Artificial Intelligence, Tata McGraw Hill
3 Nils J. Nilsson, Principles of Artificial Intelligence, Elsevier, 1980
4 StaurtRussel, Peter Norvig, Artificial Intelligence: A Modern Approach, Pearson Education, 3rd Edition,2009

REFERENCES

1 https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_0130944267894620162140_shared/overview (Introduction to Artificial Intelligence with Java)
2 https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01330144568120934419537_shared/overview (Learn Artificial Intelligence with TensorFlow)
3 https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_0130944022209576961567_shared/overview (Artificial Intelligence in 3 Hours)
4 https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_0130943993462046721454_shared/overview (Advanced Artificial Intelligence Projects with Python)

ONLINE REFERENCE

1 TOC - Introduction to Artificial Intelligence | Infosys Springboard (onwingspan.com)
2 TOC - Introduction to Deep Learning | Infosys Springboard (onwingspan.com)
3 TOC - Computer Vision 101 | Infosys Springboard (onwingspan.com)
4 TOC - Ethical AI | Infosys Springboard (onwingspan.com)
5 TOC - Prompt Engineering | Infosys Springboard (onwingspan.com)

6 TOC - AI-First Software Engineering | Infosys Springboard (onwingspan.com)
7 TOC - Generative models for developers | Infosys Springboard (onwingspan.com)

SOFTWARE REQUIREMENT

- Angular, npm, nodejs, mongo dB (all latest versions)
- TensorFlow, Python and C++

HARDWARE REQUIREMENT

- Windows 10+, Linux 8+, Mac 10+
- operating system with 8 GB RAM

INDUSTRY SCOPE

On Completion of these courses students will be able to adopt and implement various tools and concepts of Artificial intelligence in resolving problems and provide intelligent and smart solutions.

INDUSTRY USE CASES

1. Build a Convolution Neural Network for Image Recognition. [Reference : Introduction to Convolution Neural Network - Viewer Page | Infosys Springboard (onwingspan.com)]
2. Design Artificial Neural Networks for Identifying and Classifying an actor using Kaggle Dataset. [Reference : TOC - ANN - Exercise | Infosys Springboard (onwingspan.com)]
3. Design a CNN for Image Recognition which includes hyperparameter tuning. [Reference : TOC - CNN hyperparameter tuning | Infosys Springboard (onwingspan.com)]
4. Implement a Recurrence Neural Network for Predicting Sequential Data. [Reference : TOC - RNN - Exercise | Infosys Springboard (onwingspan.com)]
5. Implement Multi-Layer Perceptron algorithm for Image denoising hyperparameter tuning. [Reference : TOC - Image denoising hyperparameter tuning - Exercise | Infosys Springboard (onwingspan.com)]
6. Implement Object Detection Using YOLO. [Reference : TOC - Object Detection Using YOLO | Infosys Springboard (onwingspan.com)]
7. Design a Deep learning Network for Robust Bi-Tempered Logistic Loss. [Reference : TOC - Advanced Loss Functions and Parameter Tuning | Infosys Springboard (onwingspan.com)]
8. Build AlexNet using Advanced CNN. [Reference : TOC - Demonstration of AlexNet | Infosys Springboard (onwingspan.com)]
9. Demonstration of Application of Autoencoders. [Reference : Demonstration of Application of Autoencoders - Viewer Page | Infosys Springboard (onwingspan.com)]
10. Demonstration of GAN. [Reference : TOC - Demonstration of GAN | Infosys Springboard (onwingspan.com)]
11. Complete the requirements given in capstone project. [Reference : Problem Description - Viewer Page | Infosys Springboard (onwingspan.com)]

Mode of Delivery	Online (Self-Learning)
Software Configuration to be arranged in Institution Premises	○ Python and related libraries
Hardware Configuration to be arranged in Instituion Premises	○ Windows 10, 16GB RAM
Course Evaluation	Online Assessment
Multiple Hybrid Branch of Students	Applicable for Mechanical/Chemical

Internship/Placement Opportunities	https://infytq.onwingspan.com/
NOS Alignment	Yes- Infosys Industry Standard
Train-the-Trainer	Faculty Enablement Program
Commercials	Free of Cost

SB8053	CYBER SECURITY AND APPLICATIONS	L	T	P	C
		1	0	2	2

COURSE OBJECTIVE

The objective of this course is to provide knowledge on the threats and vulnerabilities to web applications. This is very crucial due to the dependencies of today's world on web apps and digital transactions. The course also provides details on how to secure our computer network systems from malicious activities and attacks.

UNIT-I NETWORKING AND WEB TECHNOLOGY 7 hours

Network Components - Network Basics - Network Communication -Web Technologies TCPIP - Web Services

UNIT-II INTRODUCTION TO CYBER SECURITY 8 hours

Recent Cyber Attacks - Cyber Security Concepts - Layers of Cyber Security - Introduction to Application Security - Secure Coding OWASP Top 10 - Coding Practices Secure Design – Closure [Practical demos and code on OWASP vulnerabilities and how to mitigate them]

UNIT-III FUNDAMENTALS OF INFORMATION SECURITY & FUNDAMENTALS OF CRYPTOGRAPHY 7 hours

Why information security? - What is information security? - Data Security - Network security - Application Security – Closure. Why Cryptography? – Cryptography - Shared Key Cryptography – Illustration - Shared Key Cryptography - Public Key Cryptography – Illustration - Public Key Cryptography – Hashing -Digital Signature – Illustration - Digital Signature - Applications of cryptography – Conclusion [Algorithmic representation of cryptographic methods]

UNIT-IV THREAT MODELING & IDENTITY AND ACCESS MANAGEMENT 6 hours

Basics of Threat Modeling - Learn Threat Modeling with a Use Case - Tool Walkthrough - MS Threat Modeling Tool – Assignment - Introduction to Identity and Access Management - What next

UNIT-V JAVA SE 11 PROGRAMMER II: SECURE CODING IN JAVA SE 11 APPLICATIONS 7 hours

Course Overview – Managing Denial of Service – Securing Information – Managing Data Integrity – Accessibility and Extensibility – Securing Objects – Serialization and Deserialization Security – JCA and its Principles – Provider Architecture – Engine Class – Key Pair Generation – Signature Management – Unsecure to Secure Object – Course Summary. [Demos of Secure Coding in Java]

UNIT-VI SECURITY STANDARDS AND REGULATIONS 5 hours

PCI DSS – ISMS -FIPS and NIST Special Publications – FISMA – GDPR – HIPAA – SOX – Conclusion

UNIT-VII IDENTITY GOVERNANCE AND ADMINISTRATION 5 hours

Need for IGA & basics concepts - IGA Basic Concepts and Onboarding - IGA Governance - Identity Administration in IGA - What next?

TOTAL : 45 PERIODS

COURSE OUTCOMES

On completion of the course, students will be able to:

CO1 : Identify network components, gain awareness on DHCP, DNS Server and TCP/IP architecture

CO2 : Gain understanding of threat modelling and its importance in the design of web applications

CO3 : Investigate how to secure web applications written using Java Technology. Apply secure coding techniques in Java, Python, C/C++ Programming Languages

CO4 : Practice identification of OWASP vulnerabilities and mitigation techniques

CO5 : Gain understanding of the importance of Security Standards and Regulations like PCI DSS, ISMS, FIPS, NIST Special Publications, FISMA, GDPR, HIPAA and SOX

CO6: Recognize Identity Governance and Administration (IGA) - what problems IGA solutions solve; governance models like - roles, certifications, policies and identity life cycle management

FOR FURTHER READING

1 Networking Fundamentals, 2019 edition, Packt, Author: Gordon Davies
2 Principles of Information Security, Authors: Michael E. Whitman and Herbert J. Mattord, Course technology incorp
3 CSSLP SECURE SOFTWARE LIFECYCLE PROFESSIONAL ALL-IN-ONE EXAM GUIDE, Third Edition, 3rd Edition, Authors: Wm. Arthur Conklin, Daniel Paul Shoemaker, Released February 2022,Publisher(s): McGraw-Hill,ISBN: 9781264258215

REFERENCE

1 https://infyspringboard.onwingspan.com/en/app/toc/lex_auth_01268375129606553635 4_shared/contents (Network Fundamentals)
2 https://infyspringboard.onwingspan.com/en/viewer/html/lex_auth_0135015696571596809160 (Certified Secure Software Lifecycle Professional (CSSLP) 2019: Secure Coding Practices)
3 https://infyspringboard.onwingspan.com/en/viewer/html/lex_auth_0135015689927557129660 (OWASP Top 10: Web Application Security)
4 https://infyspringboard.onwingspan.com/en/viewer/html/lex_auth_0135015930409779201309 3 (Defensive coding fundamentals in C and C++)
5 https://infyspringboard.onwingspan.com/en/app/toc/lex_auth_01350158164493107211192/overview (Security Programming: Python Scripting Essentials)

ONLINE REFERENCE

1 https://www.stealthlabs.com/blog/infographic-top-15-cybersecurity-myths-vs-reality/
2 https://microage.ca/cybersecurity-layering-approach/
3 https://www.oracle.com/java/technologies/javase/seccodeguide.html
4 https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling
5 https://www.checkpoint.com/cyber-hub/threat-prevention/what-is-sandboxing/
6 https://www.skillsoft.com/course/security-programming-python-scripting-essentials-be99adad-1f6b5346072b8e

SOFTWARE REQUIREMENT

- Python
- Java script, Node Js
- Java Development kit

HARDWARE REQUIREMENT

- i5 or i7 processor or R5 from AMD
- 16 GB of RAM. 500 GB storage system

INDUSTRY SCOPE

On completion of this course students will be able to identify vulnerability and security threats in web applications and learn to write secure code. This is extremely crucial, given the huge volume of digital transactions and web applications.

INDUSTRY USE CASES

1. Identification of basic network components, practice commands for TCP-IP architecture and subnetting. [Reference : Lab Guide - Viewer Page | Infosys Springboard (onwingspan.com)]
2. Build awareness on Defensive coding practices and control such as secure configuration, error handling, and session management, cryptography, input and output sanitization, error handling, input validation, logging and auditing, and session and exception management.[Reference: https://infyspringboard.onwingspan.com/web/en/viewer/html/lex_auth_0135015696571596809160]
3. Practice defensive coding practices in C/C++ such as inspections, testing, and input validation. [Reference: Defensive Coding Fundamentals for C/C++ - Viewer Page | Infosys Springboard (onwingspan.com)
4. Explore the top 10 OWASP vulnerabilities, their causes, consequences, and mitigation techniques.
   [Reference: OWASP Top 10: Web Application Security - Viewer Page | Infosys Springboard (onwingspan.com)],OWASP.org,http://cwe.mitre.org/top25/archive/2021/2021_cwe_top25.html.
   Make a report of the studied material.
5. Practice secure coding techniques in Python programming language [Reference: https://infyspringboard.onwingspan.com/en/app/toc/lex_auth_01350158164493107211192/overview
6. Create a login page with username and password which will connect to a database which will store the name and password. You can use Java and HTML code and database as per convenience. Simulate an SQL injection attack. Write embedded SQL code to avoid SQL injection attack. Document how this is taken care in the later versions of Java.
7. Create a login page with username and password which will connect to a database which will store the name and password. You can use Python as a base and database as per convenience. Simulate an SQL injection attack. Write the revised code in Python that will sanitize the inputs and help prevent an SQL injection attack.
8. Read and understand the Heartbleed vulnerability. Identify the code in C++ that can simulate this vulnerability and code to fix it. Document the secure coding practices to take care of this vulnerability and the reasons for it to happen.
9. Given a web application, try out the top 10 OWASP vulnerabilities and how to mitigate them. [Reference: TOC - Explore OWASP Top 10 Vulnerabilities | Infosys Springboard (onwingspan.com), will be given as a document with code]

Mode of Delivery	Online (Self-Learning)
Course Evaluation	Online Assessment
Multiple Hybrid Branch of Students	Applicable for All Branches of Engineering (First Year & Final Year)
NOS Alignment	Yes- Infosys Industry Standard
Train-the-Trainer	Faculty Enablement Program
Commercials	Free of Cost

SB8054	**MACHINE LEARNING**	**L**	**T**	**P**	**C**
		1	**0**	**2**	**2**

COURSE OBJECTIVE

The objective of this course is to provide a view of data science, machine learning, basic implementation using Python and how machine learning is applied in various domains in the industry.

UNIT-I EXPLORE MACHINE LEARNING USING PYTHON 18 hours

Introduction to Machine Learning - Regression – Classification – Clustering – Introduction to Artificial Neural Network

UNIT-II INTRODUCTION TO DATA SCIENCE 4 hours

Data Science: The Data Revolution - Components of Data Science - Data Science in Action – Conclusion

UNIT-III PYTHON FOR DATA SCIENCE 15 hours

Why Python Libraries – NumPy - Introduction to NumPy - Operations on NumPy – Pandas – Introduction to Pandas – Introduction to Pandas Object – Working with datasets – Pandas Plots - Matplotlib – Introduction to Matplotlib – Types of Plots – Scikit-learn – Machine Learning using sklearn. [Practical hands-on exercises using NumPy, Pandas, Matplotlib]

UNIT-IV DATA VISUALIZATION USING PYTHON 9 hours

Data visualization using Python: Data Visualization: Developing insights from data using Basic Plots using Matplotlib (Box, Scatter, Line, Bar, Pie, Histogram), Statistical analysis using Heatmap, Kernel Density plot using Seaborn, Network Graphs, Choropleth Map Using Ploty, Word Cloud. [Practical hands-on exercises for creating charts]

UNIT-V CAPSTONE PROJECT 3 hours

Hands-on Exercises for Practicing Machine Learning Models Using Capstone Project

TOTAL : 45 PERIODS

COURSE OUTCOMES

On completion of the course, students will be able to:

CO1 : Demonstrate fundamental understanding of the history of artificial intelligence (AI) and its foundations.

CO2 : Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning.

CO3 : Assess and select appropriate data analysis models for solving real-world problem.

CO4 : Demonstrate the importance of data visualization, design, and use of visual components.

FOR FURTHER READING

1 Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems by Aurelien Geron

2 Machine Learning using Python by Manaranjan Pradhan and U Dinesh Kumar

REFERENCES

1 TOC - Exploratory Data Analysis | Infosys Springboard (onwingspan.com)
2 TOC - Regression Analysis | Infosys Springboard (onwingspan.com)
3 TOC - Clustering using Python | Infosys Springboard (onwingspan.com)
4 TOC - Natural Language Processing for developers | Infosys Springboard (onwingspan.com)
5 TOC - Introduction to Natural Language Processing | Infosys Springboard (onwingspan.com)

ONLINE REFERENCE

1 https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012600400790749184237_shared/overview (Explore Machine Learning)
2 https://infyspringboard.onwingspan.com/web/en/app/toc/lex_12666306402263577000_shared/overview (Introduction to Data Science)
3 https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01333063698060902494_shared/overview (Python for Data Science)
4 https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_0126051913436938241455_shared/overview (Data Visualization Using Python)
5 https://infyspringboard.onwingspan.com/web/en/viewer/web-module/lex_auth_0131364956063416323344_shared?collectionId=lex_auth_012600400790749184237_shared&collectionType=Course&pathId=lex_auth_012600400826621952202_shared(Capstone Project)

SOFTWARE REQUIREMENT

Python software

HARDWARE REQUIREMENT

Windows 10, 16GB RAM

INDUSTRY SCOPE

On Completion of this course students will be able to apply various approach in ML and provide real life solutions to problems. They will be able to identify, predict and suggest probable outcome based on historic data.

INDUSTRY USE CASES

[For Use Cases: 1 to 9]

Lee decides to walk 10000 steps every day to combat the effect that lockdown has had on his body's agility, mobility, flexibility, and strength. Consider the following data from fitness tracker over a period of 10 days.

Day Number	Steps Walked
1	6012
2	7079
3	6886
4	7230
5	4598
6	5564
7	6971
8	7763
9	8032
10	9569

1. Represent the above data in a 10x2 array. In each row, the first element should contain day number and second element should contain steps walked. [Reference : Numpy exercise - Viewer Page | Infosys Springboard (onwingspan.com)]
2. Lee notices that the tracker's battery dies every day at 7.00 PM. Lee discovers that in an average, he walks 2000 steps every day after 7.00 PM. Perform an appropriate operation on your array to add 2000 steps to all the observations. [Reference : Numpy exercise - Viewer Page | Infosys Springboard (onwingspan.com)]
3. Write a Program that returns the steps walked if the steps walked are more than 9000. [Reference : Numpy exercise - Viewer Page | Infosys Springboard (onwingspan.com)]

4. Print an array containing steps walked [Reference : Numpy exercise - Viewer Page | Infosys Springboard (onwingspan.com)]
5. Import the given rainfall data set into Python environment as a Pandas DataFrame. [Reference : Pandas Exercise - Viewer Page | Infosys Springboard (onwingspan.com)]
6. Construct a Program to find the district (given in rainfall data set) that gets the highest annual rainfall. [Reference : Pandas Exercise - Viewer Page | Infosys Springboard (onwingspan.com)]
7. Construct a Program to display top 5 states that gets the highest annual rainfall. [Reference : Pandas Exercise - Viewer Page | Infosys Springboard (onwingspan.com)]
8. Construct a Program to display state-wise mean rainfall for all months using a pivot table and to display count of districts in each state. [Reference : Pandas Exercise - Viewer Page | Infosys Springboard (onwingspan.com)]
9. For each state, write a code to display the district that gets highest rainfall in May. Also display the recorded rainfall. [Reference : Pandas Exercise - Viewer Page | Infosys Springboard (onwingspan.com)]
10. For a given values for x and y, use Matplotlib to plot a figure.[Reference : Matplotlib Exercise - Viewer Page | Infosys Springboard (onwingspan.com)]
11. Create a bivariate plot to find if there is a correlation between credit card limit and average purchase made on the card. [Reference : Hands-On Try Out - Viewer Page | Infosys Springboard (onwingspan.com)]
12. Visualise the distribution of values for credit card limit and average purchase made on the card. Also, identify the outliers in the data, if any. Reference : Hands-On Try Out - Viewer Page | Infosys Springboard (onwingspan.com)]
13. Provide a visual representation of the number of customers in each income group using a bar chart. Reference : Hands-On Try Out - Viewer Page | Infosys Springboard (onwingspan.com)]
14. Plot the frequency distribution of the total transaction amount. Reference : Hands-On Try Out - Viewer Page | Infosys Springboard (onwingspan.com)]
15. Graphically represent the percentage of customers retained and those attrited. Highlight the latter by slicing it apart from the main pie. Reference : Hands-On Try Out - Viewer Page | Infosys Springboard (onwingspan.com)]
16. Capstone project - https://infyspringboard.onwingspan.com/web/en/viewer/web-module/lex_auth_0131364956063416323344_shared?collectionId=lex_auth_012600400790749184237_shared&collectionType=Course&pathId=lex_auth_012600400826621952202_shared(Capstone Project)

Mode of Delivery	Online (Self-Learning)
Software Configuration to be arranged in Institution Premises	○ Python and related libraries
Hardware Configuration to be arranged in Institution Premises	○ Windows 10, 16GB RAM
Course Evaluation	Online Assessment
Multiple Hybrid Branch of Students	Applicable for Mechanical/Chemical
Internship/Placement Opportunities	https://infytq.onwingspan.com/
NOS Alignment	Yes- Infosys Industry Standard
Train-the-Trainer	Faculty Enablement Program
Commercials	Free of Cost

SB8055 **BLOCKCHAIN DEVELOPMENT** **L T P C**

1 0 2 2

COURSE OBJECTIVE:

1. Ability to develop a structure of data with inherent security qualities based on principles of cryptography, decentralization, and consensus
2. Develop smart contracts using the Solidity programming language (including a deep understanding of the provided Libraries)
3. Implement the consensus methods in transactions and apply blockchain for different application domains
4. Develop your own applications for various user requirements using Blockchain and launch it for public and commercial use
5. Apply Hyperledger Fabric and Ethereum platform to implement the Block Chain Application.

COURSE OUTLINE:

UNIT I Introduction and Overview of Blockchain 3 + 6

Introduction | What is a Blockchain? | Why Blockchain? | Types of Blockchain | Scope and Importance of the Technology | Future Opportunities | Blockchain Platforms | Miners - the Nonce and the Cryptographic Puzzle | Proof of Work & Proof of Stake | Consensus Algorithms | Fault Tolerance Mechanism | Creation of a Block | Transaction Record

UNIT II Blockchain - the Future of Digital currency 3 + 6

What is Bitcoin? | Bitcoin's Monetary Policy | Bitcoin Wallets | Application of Bitcoins | Bitcoin - Script, Address and Transaction | Understanding Mining Difficulty | Virtual Tour of Bitcoin Mine | Mining Pools | Nonce Range | How Miners Pick Transaction | CPU's Vs GPU's Vs ASIC's | How does Mempools Work | Orphaned Blocks | The 51% Attack | How Wallets Work | Signatures: Private and Public Keys Demo: Setting up a Wallet, Signatures & Key

UNIT III Ethereum 3 + 6

What is Bitcoin? | Bitcoin's Monetary Policy | Bitcoin Wallets | Application of Bitcoins | Bitcoin - Script, Address and Transaction | Understanding Mining Difficulty | Virtual Tour of Bitcoin Mine | Mining Pools | Nonce Range | How Miners Pick Transaction | CPU's Vs GPU's Vs ASIC's | How does Mempools Work | Orphaned Blocks | The 51% Attack | How Wallets Work | Signatures: Private and Public Keys Demo: Setting up a Wallet, Signatures & Key

UNIT IV Smart Contracts 3 + 6

What is a Smart Contract? | Smart Contract Environment | Solidity Programming | Array, Enum and Structs | Inheritance | ERC20 | ERC721

UNIT V Hyperledger and IPFS 3 + 6

What is Hyperledger? | Importance of Hyperledger | Hyperledger Architecture & its Layers | Hyperledger Transactions | Hyperledger Fabric | Hyperledger Fabric Model | Building a Hyperledger Fabric Network | Fabric Peer | Hyperledger Fabric CA | Sawtooth | Iroha & Indy | API in Hyperledger | Network Topology in Hyperledger | IPFS Introduction | IPFS Working | IPFS for Blockchain Application

TOTAL: 45 PERIODS

COURSE OUTCOME:

On completion of the course the student:

1. can write a smart contract to insert a value into the Ethereum blockchain using metamask
2. is able to navigate the Remix IDE web URL on the browser and explore the various tabs and features of the IDE
3. can connect to the Ganache localhost through Web3 provider, Ropsten testnet using Metamask and Injected Web3
4. is able to write a Program on the arrays in the Solidity with regards to fixing the length and dynamic
5. can generate the ABI and Bytecode of a Smart Contract by compiling the solidity file
6. can develop a program that transfers the smart contracts between the parties which can aid in automated

20 INDUSTRY USE CASES

Problem Statement: 1

Blockchain technology came into the ground to overcome these issues. It offers decentralized nodes for the banking system and is used to produce a transparent banking system for its end-to-end verification advantages. This technology is a replacement for the traditional banking system with distributed, nonrepudiation, and security protection characteristics.

You are a Blockchain expert in a major corporate bank and have been tasked to create a smart contract to perform banking transactions. Create a Smart Contract for a banking application in solidity that allows users to do the following:

- Mint money into your account
- Withdraw money from your account
- Send money from your account to smart contract address
- Check balance

Problem Statement: 2

Blockchain is a technology designed to manage patient data that has the potential to support transparency and accountability. A blockchain is a ledger of transactions where an identical copy is visible to all the members of a computer network. Network members validate the data entered into the ledger, and once entered, the data is immutable.

Create a solution where you can store the electronic health record of the patients in a distributed and decentralized network. You should be able to query and change the ownership of the record as necessary.

Problem Statement: 3

Blockchain is a technology designed to manage education data that has the potential to support transparency and accountability. A blockchain is a ledger of transactions where an identical copy is visible to all the members of a computer network. Network members validate the data entered into the ledger, and once entered, the data is immutable.

Design a solution where you can store the digital certificates of the students in a distributed and decentralized network. You should be able to add the certificated details into the blockchain query the certificate details from the blockchain.

Problem Statement: 4

Blockchain is a technology which enables elections to be done transparently. We can avoid rigging or any corrupt activities using the technology and should be able to make sure that the votes are also accounted for on a real-time basis.

Design an electronic voting system, using the ethereum blockchain (smart contracts) and more precisely the RPC test which enables account generation with a private and public key. Blockchain electronic voting system using smart contracts.

Problem Statement: 5

Smart cities and smart houses are in fashion and thus all this can be kept in a blockchain. We can focus on building system which can manage all the real estate related contracts through blockchain technology which will enhance security and will provide more efficiency.

Design a smart contract using the Ethereum blockchain in a distributed and decentralized network. You should be able to add the property details to the blockchain, query the property details from the blockchain and should be able to change the ownership of the property appropriately.

Problem Statement: 6

Food items like fruits and vegetables generally do not have any expiry date mentioned so it becomes important to understand the origin of these food items and know the date when was it sent to the distributor from the farmer and so on.

Design a smart contract using the ethereum blockchain where you should be able to authenticate the food item and consume that without any worry

Problem Statement: 7 (Identity)

Blockchain is a technology that enables identities to be stored transparently. It offers decentralized nodes for end-to-end verification advantages. This technology is a replacement for traditional identity management with distributed, nonrepudiation, and security protection characteristics.
Design a smart contract using the Ethereum blockchain where you should be able to store the identity details in the blockchain and should be able to query the details of the identity from the blockchain

Problem Statement: 8
Blockchain is a technology that allows you to store books transparently. It offers decentralized nodes for the end-to-end verification advantages in the library. This technology is a replacement for a traditional book management system with distributed, non-repudiation, and security protection characteristics.
Design a smart contract using the Ethereum blockchain where you should be able to store your book details in the blockchain and should be able to query the details of the books from the blockchain and if required we should be able to change the ownership of the books and the same should be updated in the blockchain.

Problem Statement: 9
Blockchain is a technology that allows you to trace your drugs transparently. It offers decentralized nodes for the end-to-end verification to trace the drugs in a transparent manner. This technology is a replacement for traditional drug management systems with distributed, non-repudiation, and security protection characteristics.
Design a smart contract using the Ethereum blockchain where you should be able to track the drugs transparently.

Problem Statement: 10
Blockchain is a technology that allows you to trace your vaccines transparently. It offers decentralized nodes for end-to-end verification to trace the vaccines in a transparent manner.
Design a smart contract using the Ethereum blockchain where you should be able to track the vaccines and you should be able to add the details of the vaccine to the blockchain and should be able to query whenever it is required.

Problem Statement: 11
Blockchain is a technology that allows you to trace data transparently. It offers decentralized nodes for the end-to-end verification to trace the transportation data in a transparent manner.
Design a smart contract using the Ethereum blockchain where you should be able to track the National and state highways, toll collection, tracking of public infrastructure using the smart contract in the blockchain

roblem Statement: 12
Design a smart contract using the Ethereum blockchain where you track the progress on climate agreement through Blockchain. You should be able to add the confidential details of climate change into the blockchain, should be able to query the details from the blockchain and then change the confidential details whenever it is required.

Problem Statement: 13
Design a smart contract using the Ethereum blockchain where you can add the relevant documents on Micro-financing, and financing small businesses or individuals into the blockchain You should be able to add the financial details into the blockchain, should be able to query the details from the blockchain and then change the financial details whenever it is required.

Problem Statement: 14
Design a smart contract using the Ethereum blockchain where you can add the relevant documents on agriculture data into the blockchain You should be able to add the agriculture product details into the blockchain, should be able to query the details from the blockchain, and then change the details whenever it is required.

Problem Statement: 15
Blockchain is a technology designed to manage farm insurance data that has the potential to support transparency and accountability. A blockchain is a ledger of transactions where an identical copy is visible to all the members of a computer network. Network members validate the data entered into the ledger, and once entered, the data is immutable.

Design a smart contract using the Ethereum blockchain where you can add the farm insurance data into the blockchain You should be able to add details into the blockchain, should be able to query the details from the blockchain and then change the details whenever it is required to change the insurance details whenever it is required.

Problem Statement: 16
Blockchain is a technology designed to manage toll-free data that has the potential to support transparency and accountability. A blockchain is a ledger of transactions where an identical copy is visible to all the members of a computer network. Network members validate the data entered into the ledger, and once entered, the data is immutable.
Design a smart contract using the Ethereum blockchain where you can add the toll-free data into the blockchain You should be able to add details into the blockchain, should be able to query the details from the blockchain and then change the details whenever it is required. You can take all the parameters which you want to store as part of the toll data.

Problem Statement: 17
Blockchain technology is a decentralized, distributed ledger that stores the record of ownership of digital assets. Any data stored on the blockchain is unable to be modified, making the technology a legitimate disruptor for industries like payments, cybersecurity, and healthcare. Discover more about what it is, how it’s used, and its history.
Design a distributed ledger via the nodes connected to the chain. You can use any kind of electronic device for the blockchain nodes to maintain copies of the chain network functioning and can create inherent security by giving a unique alphanumeric identification number needed to shows their transactions

Problem Statement: 18
Blockchain is a technology designed to strengthen media industry data that has the potential to deal with data privacy, royalty payments, and piracy of intellectual property. A blockchain can give the industry a much-needed facelift when it comes to data rights, piracy, and payments.
Design a smart contract using the Ethereum blockchain where you can prevent digital assets, from existing in multiple places. You should be able to add details to the blockchain, should be able to preserve ownership, make piracy from the blockchain and then change the details whenever it is required to maintain the data integrity.

Problem Statement: 19
Blockchain is a technology to secure government documents and also improve bureaucratic efficiency, and accountability and reduce massive financial burdens. Blockchain has the potential to revolutionize our elections. Blockchain-based voting could improve civic engagement by providing a level of security and incorruptibility and transparency by recording a public record of all activity.
Design a smart contract using the Ethereum blockchain where you can encrypt a biometric security system making the voting platform an open-source virtual blockchain ballot box. You can take all the parameters which you want to store as part of the ballot data.

Problem Statement: 20
Blockchain is a technology that is designed to manage non-fungible tokens (NFTs). NFTs are simply digital items, like music, art, GIFs, and videos that are sold on a blockchain, ensuring that a sole owner can claim full rights to them. Consumers can now claim sole ownership over some of the most desirable digital assets for their applications.
Design a smart contract using the Ethereum blockchain where you can add the popular IPs and brand figures into digital collectibles for consumers. You should be able to add an ecosystem into

the blockchain, which allows fans and collectors to interact with icons in the form of official licensed digital collectibles.

S.No	Blockchain Reference Books
1	Saurabh Kumar and Saxena Ashutosh., 2020." Blockchain Technology: Concepts and Applications" Wiley India Pvt Ltd, First Edition, ISBN-10: 8126557664, ISBN-13: 978-8126557660
2	Arun, Jai Singh & Cuomo, Jerry & Gaur, Nitin., 2019." Blockchain for Business" Pearson Education, First edition, ISBN-10: 938958888X, ISBN-13: 978-9389588880
3	Tulajadas Choudhari, Ambadas & Sarfarz Ariff, Arshad & M.R. Sham., 2020. "Blockchain for Enterprise Application Developers" Wiley, First edition, ISBN-10: 8126599960, ISBN-13: 978-8126599967
4	Subramanian, Chandramouli & George A., Asha & K.A., Abhilash & Karthikeyan, Meena., 2020. "BLOCKCHAIN TECHNOLOGY" Universities Press (India) Pvt. Ltd., First edition, ISBN-10: 9389211638, ISBN-13: 978-9389211634

SB8056 DIGITAL MARKETING L T P C
1 0 2 2

COURSE OBJECTIVE:

1. Ability to develop a digital marketing plan that will address common marketing challenges
2. Ability to Articulate the value of integrated marketing campaigns across SEO, Paid Search, Social, Mobile, Email, Display Media, and Marketing Analytics.
3. Potential to recognize key performance Indicators tied to any digital marketing program
4. Calibre to improve Return on Investment (ROI) for any digital marketing program
5. Launch a new, or evolve an existing, career path in Digital Marketing and the ability to build their own start-ups

COURSE CONTENT:

UNIT I Introduction to Digital Marketing 3 + 6

Basics of Marketing **|** Traditional Marketing Vs Digital Marketing **|** 5P's of Marketing **|** Segment, action, and Targeting **|** Customer Lifecycle **|** Digital Marketing Modules **|** RACER Framework **|** Digital Marketing Trends

UNIT II Social Media Marketing 3 + 6

Marketing Approach **|** Gen X **|** Gen Y **|** Gen Z **|** Gen Alpha **|** Conversation Prism **|** Social Media Strategy **|** Social Media Channels Penta Social Elements (Social Talking, Listening, WOM, Feedback, Selling) Personas Influencer Marketing Sentimental Analysis Effective Social Media Marketing Usage Social Media Tools Jargons

UNIT III Content Marketing & Automation 3 + 6

Content Marketing Landscape **|** Types of Content Marketing **|** Content Marketing Strategy **|** Affiliated Marketing **|** Content Marketing Tools **|** Jargon **|** Effective Email Marketing **|** Essentials in Email Marketing **|** Types of Emails **|** Email Automation **|** Email Marketing Metrics **|** Marketing Automation **|** Martech Landscape **|** Features of Marketing Automation **|** Choosing a Marketing Automation Platform

UNIT IV Video Marketing & Digital Marketing 3 + 6

Create Economy **|** YouTube **|** YouTube Ads **|** YouTube Partner Program **|** Instagram Marketing **|** Live Streams **|** Revenue Generating Streams **|** So-Lo-Mo | Paid Search / PPC **|** Social Media Ads **|** Types of Google Ads**|** Keywords **|** How do Google Ads work? **|** Cost & Budgets**|** Remarketing & Retargeting **|** Tools

UNIT V Search Engine Optimization & Search Engine Optimization 3 + 6

SEO & SEM Fundamentals**|** How Search Engine On-Page **|** Types of Keywords **|** On-Page SEO **|** Off-Page SEO **|** SEO Tools **|** SEO Reporting **|** Digital Marketing Strategy for a Startup/Entrepreneur **|** Business Growth Opportunities **|** Career Growth Opportunities**|** Best Practices

TOTAL : 45 PERIODS

COURSE OUTCOME:

The student can :

1. develop a compelling content strategy for all kinds of business
2. build PPC campaigns and also design Paid ads and optimize great ad copywriting
3. design suitable display ads (using tools like Canva), Publish and run ads as per business goals
4. do e-mail Marketing, social media, build a template, Insert Content and Images
5. can build their own virtual organization by providing various digital marketing services

20 INDUSTRY USE CASES:

1. Brand Name Creation: Create a brand name, Brand identity Design: Design a brand logo - using Canva and Create an email account on Gmail to link all your project work - (brandname@gmail.com), use this email account centrally to access all tools and platforms by signing up with the created gmail id.
2. Social Media Campaign: Create a mock Social media campaign - choose the ideal channel (FB, Insta, Linkedin, Youtube etc)
3. Keyword Research/Analysis: Use social listening tools handled in the course to identify keywords for your content strategy (Listdown top 5 keywords and the source of SEO tools used)
4. Blogging: Create your first blog - 500-800 words on wordpress blogs (grammarly, COPYAI) and promote the blog on any of the social media of your choice not by posting the blog directly but by creating a promo link for the blog and attract a minimum of 50 Likes on the blog.
5. Building a Website: Create your first website using - WIX/Canvae
6. Build a Landing Page: Create a landing page using - mailchimp/hubspot
7. Build a content marketing plan: with a focus on top 3 content types (video, email, blogs, podcast etc) - based on a content marketing template from Hubspot
8. Brand Promo Video: Create a 60 sec promo video for your brand - using Canva, Camtasia etc and it should be a explainer video. To achieve this first write your script, then storyboard it, followed by creating an instructional design and then finally create your explainer video
9. Instagram Reel: Create an Instagram reel for your business brand and promote it attract a minimum of 100 Likes
10. Hashtag Generation: Share a list of 5 #Hashtags for the launch campaign, first identify the buzzing area, calculate the penetration power, build a curiosity around the Hashtag and finally release the Hashtag and make a minimum of 50 members from outside your friends and family to viral it.
11. Build and Email Campaign: Create a brand launch email - using Mailchimp
12. SEO :Build and execute the On-page SEO for your website
13. Create a Google Business Page
14. PPC Ad: Create a mock paid - ad campaign for your brand - Google Adwords
15. Social Ads: Create a mock sponsored post for Facebook/ Instagram/Linkedin
16. Google Analytics: Embed a Google Analytics Code on your Website
17. Sitemap: Submit your website’s sitemap to Google Search Console
18. Youtube Ad: Create a mock Youtube Ad for your brand
19. Traffic: Generate website traffic to reach at least 500 visits by the end of your project time

20. Field Visit to two businesses (eg: Malabar Gold, MRF, HAP daily, Freshworks, Zoho) meet the digital marketing managers and perform a complete in-person interview on their respective digital marketing strategies, present the research and study in a ppt format

REFERENCES:

S.No	Digital Marketing Book References
1	Gupta, Seema., 2022. "Digital Marketing" McGraw Hill, Third Edition, ISBN-10: 935532040X, ISBN-13: 978-9355320407
2	Kagan, Jeremy & Shekhar Singh, Siddharth., 2020. "Digital Marketing: Strategy & Tactics" Wiley, First Edition, ISBN-10: 9390395496, ISBN-13: 978-9390395491
3	Maity Moutusy., 2022. "Digital Marketing" Oxford University Press, Second edition, ISBN-10: 9354972470, ISBN-13: 978-9354972478
4	Hemann Chuck & Burbary Ken., 2019. "Digital Marketing Analytics" Pearson Education, Second Edition, ISBN-10: 9353430194, ISBN-13: 978-9353430191

SB8057 DATA ANALYTICS WITH TABLEAU & PYTHON

		L	T	P	C
SB8057	DATA ANALYTICS WITH TABLEAU & PYTHON	1	0	2	2

COURSE OBJECTIVE

- The main objective of the course is to prepare the students to become skilful by doing hands-on project-based learning in a real-time environment. Also making them become industry /job-ready

COURSE OUTCOMES

- To gain hands-on working skills and industry project experience by learning & Hands - on - with Tableau Platform - Data Extraction using Database & Flat files working with Metadata and Data Blending.
- Working with Filters - Organizing Data & Visual Analytics - Working WIth Mapping - Calculations - Expressions & Parameters - Using the Tableau Analytics interface/paradigm to create powerful Visualizations - Dashboards & Story effectively - Represent data using various visualization types.
- Python for Data Science - Build a web application using flask web framework & Build a number of use cases in multiple domains such as Financial Services - Insurance - Retail - Ecommerce - Telecom - Agriculture - Aviation etc.

COURSE CURRICULUM

UNIT - I Introduction to Business Intelligence 06

Business Intelligence - Data Integration - Data Processing - Data Presentation - ETL Architecture - Introduction to Data Analytics - Types of Data Analytics - Descriptive Analytics - Diagnostic Analytics - Predictive Analytics - Prescriptive Analytics - Analytics & Applications

Data Extraction: Introduction to Database - Creating Database & Table - CRUD Operation on database tables - Basic SQL Operations

Introduction to Tableau : Introduction to Tableau - Overview & Features - Connecting Tableau to Data Sources - Working with Flat files - Connecting Spreadsheets

Architecture of Tableau : Architecture of Tableau - The interface of Tableau (Layout, Toolbars, Data Pane, Analytics Pane, etc…) - Tableau field types - Saving and publishing a data source - Live vs extract connection - Various file types - Ways to share and export the work done in Tableau

UNIT - II Data Visualization 07

Charts : Histograms - Box plot - Pie - Bar - Line - Bubble - Bullet - ScatterTree - Heat maps - Maps - Text table - Highlighted tables

Working with Metadata and Data Blending: Connecting to Data Source - Tableau data types - Connection to Excel - Cubes and PDFs - Management of metadata and extracts - Joins (Left, Right, Inner, and Outer) and Union - Data blending - Data preparation

Advanced Data Manipulations: Mark and highlight - Groups - Sets - Bins - Hierarchies - Sorting and Types - Using the Formatting pane to work with the menu, fonts, alignments, settings, etc… - Editing axes and annotations

UNIT - III Adavance Tableau 08

Working with Filters,Organizing Data & Visual Analytics: Working with Filters - Filters (addition and removal) - Filtering continuous dates, dimensions, and measures - Filtering in Tableau - Types of filters - Filtering the order of operations - Organizing Data and Visual Analytics

Working WIth Mapping ,Calculations and Expressions: Working on coordinate points - Plotting longitude and latitude - Editing unrecognized locations - Working on the background image - Map visualization - custom territories - Calculation syntax and functions in Tableau - The creation of calculated fields - Quick table calculations

Working with Parameters: Creating parameters - Parameters in calculations - Using parameters with filters - Column selection parameters - Chart selection parameters - How to use

UNIT - IV Dashboards and Stories 04

Dashboard & Stories: Building and formatting a dashboard using size - objects - views - filters and legends - Best practices for making creative as well as interactive dashboards using the actions - Including the intro of story points - Creating as well as updating the story points - Adding catchy visuals in stories - Adding annotations with descriptions; dashboards and stories - Highlight actions, URL actions, and filter actions - Selecting and clearing values - Dashboard examples using Tableau workspace and Tableau interface

UNIT - V Python for data science 05

Basics of Python - Working with Data Structures - Flow Controls - Conditional Constructs – Functions Python Packages - Working with Numpy & Pandas - Introduction to Data Visualization

Build Tableau Web Application: Introduction to Flask - Working with Flask Framework - Building application with flask framework - Embedding Dashboard & Story with web application

TOTAL: 45 PERIODS

REFERENCE	
1	Python Programming: - h ttps://developers. google.com/edu/py thon
2	Tableau - https://w w w.tableau.com/learn/training
3	What is BI - https://w w w.tableau.com/data-insights

SOFTWARE REQUIREMENT

- Tableau Desktop
- Browser
- Google Colab / Anaconda Navigator

SB8058	MANUFACTURING OF MECHANICAL COMPONENTS USING CNC TURNING / MILLING	L T P C
		1 0 2 2

HARDWARE REQUIREMENT

- 8GB RAM, Processor- Intel core i3/M1
- OS - Windows10/Linux/MAC
- Minimum 5 MBPS download speed of internet connection

COURSE OBJECTIVE

1. Understanding the concept of CNC programming and operations and making applications on CNC turning and milling machines.
2. Gaining Advanced knowledge of CNC machining process and hands on experience to program and make the component.
3. Applying the concepts of machine and equipment's handling with CNC Program in CNC Turning and Milling machines.
4. Comprehending the advantages and application of CNC turning and milling machines in the manufacturing industry with industrial drawing and component.
5. Experiencing a hands-on component manufacturing as per industrial practice.

COURSE CONTENT

UNIT I CNC technology and CNC programming 3 + 6

CNC programming Introduction and demonstration of line programs - CNC programming on lathe & milling machine - CNC programming for lathe and milling machines using different machining cycles - Procedures Associated with part programming, Cutting process parameter selection, Process planning issues and path planning - G & M Codes, Interpolations, Canned Cycles and Subprograms.

UNIT II Programming generation and application in CNC Simulator 3+ 6

Functions and Identification of different parts of CNC lathe including data input – Functions and Identification of different parts of CNC mill including data input -Tool compensations Exposure for programming and simulation in Fanuc control - CNC programming on lathe & milling machine for different operations in simulator - Practice on CNC controller using on-screen simulation for generating different profile.

UNIT III Programming and Operations in CNC Turning 3 + 6

Plan and optimize programs for CNC turning operations - Calculate parameters like speed, feed etc - set a reference for the various operations - Prepare operation and operation sequence for the lathe operations like turning, grooving etc - Prepare & set CNC lathe operations and test run programmed - Execute program and inspect simple geometrical forms / standard parts - Use of various PPE's on CNC lathe machine.

UNIT IV Programming and Operations in CNC Milling 3 + 6

Plan and optimize programs for CNC Milling operations - Calculate parameters like speed feed, depth of cut etc - Set a reference for the various operations. - Various methods of work process like edge finding, block center etc - Prepare operation and operation sequence for the milling operations like face milling, End milling, Drilling, etc -Prepare & set CNC Milling operations and test run programmed - Execute program and inspect simple geometrical forms / standard parts - Use of various PPE's on CNC milling machine.

UNIT V Quality Control 3 + 6

Principles of measurement - Limits, fits and Tolerances - Direct measurement such as Vernier Caliper, Vernier Height gauge, Vernier Depth gauge, Outside Micrometer, Inside Micrometer, Depth Micrometer – Introduction to Co-ordinate Measuring Machine and measure all parameters using Co-ordinate Measuring Machine.

TOTAL : 45 PERIODS

COURSE OUTCOMES

Students will be able to:

- Create the programming's for industrial components produced by CNC machines.
- Do the Cutting process parameter selection, tools and path planning.
- Relate and analyse to get industrial skills in the field of automotive manufacturing.
- Do the industrial measuring practice and examine functionality and comparison of final product as per specification.

FOR FURTHER READING:

Introduction To Automation – Demonstrate understanding of the use and programming of industrial robots and related automation – Functions and applications of industrial robots.

REFERENCES

1.CNC Programming and CNC Machining – Centurion University
2.Training Course on CNC Lathe & Milling Machines – National University
3. Ministry of Micro, Small and Medium Enterprises, New Delhi (MSME-Technology Centre)

ONLINE REFERENCES:

- Centurion University
- https://courseware.cutm.ac.in/courses/cnc-programming-and-cnc-machining/
- National University – Science and Technology https:// portal.nu.edu.om/ pts/pdf/ training Course_59.pdf
- Ministry of Micro, Small and Medium Enterprises, New Delhi (MSME-Technology Centre)
- https://nqr.gov.in/sites/default/files/5.%20IGTR%20AURANGABAD%20ACCCM%20CURRICULUM.pdf

SOFTWARE REQUIREMENT:

- Simulation software
- Fanuc control software
- Hass Control software

HARDWARE REQUIREMENT:

- CNC Turning machine
- CNC Milling machine
- Co Ordinate Measuring Machine
- Control Panel - Hass
- Computers
- Measuring Instruments

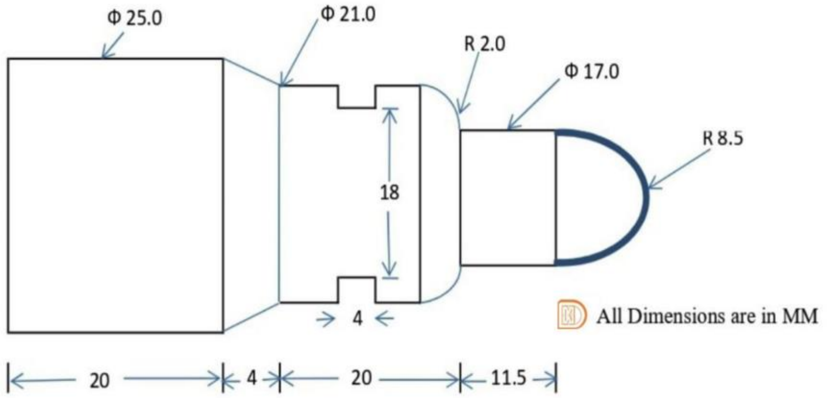
INDUSTRY SCOPE:

On Completion of this course, students get modern manufacturing industrial practice with capable of writing CNC programming for complex industrial components. Identify and Selection of appropriate measuring tools and instruments for manufactured component.

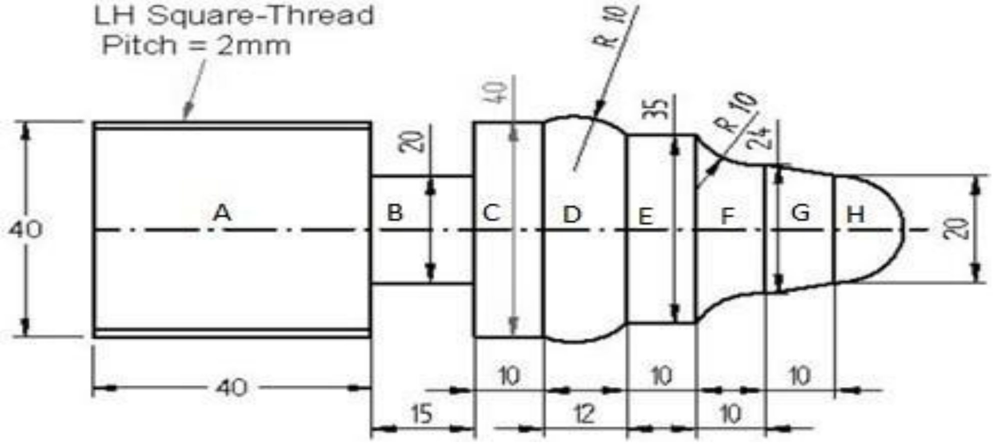
20 INDUSTRY USE CASES

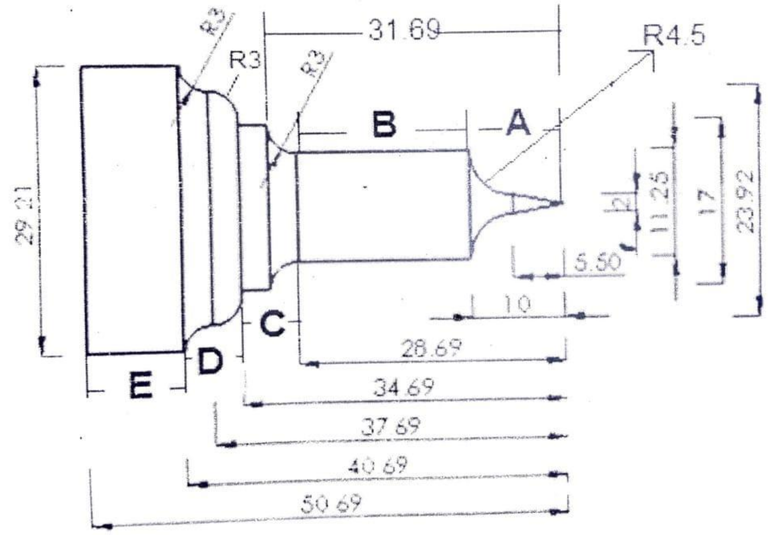
1. Create CNC Programming for below model components in multiple sectors like automobile, aerospace, medical, railway industries. Etc.,
2. Create and analyse lead time requirements.
3. Plan to achieve production resilience.
4. Stepping into evolution of CNC technology and Automation.
5. Derive the plan for reduce downtime and increase turnaround times, increasing productivity in factories.
6. Visit multiple domain factories and identify the demand of 5 axis and special purpose machines.
7. Find the more complex manufacturing setup and provide solution with IioT (Industrial Internet of Things).
8. Pick elevations of CNC Career growth and act accordingly.
9. Find the different departments support that should needed for the industrial process completion.
10. Create an interactive and future based project.
11. Create an automation process with SAP tool.
12. Derive and establish man power and work study.
13. Find the various ways to short the lead time.
14. Find and analyse to improve Zero Defects and Greater Accuracy.
15. Using Least cost method to recycle the waste material and make that to be reused.
16. Set and implement Kaizen activities in manufacturing.
17. Using the SOP guidelines monitor and regularize the process.
18. Create and standardize the action plan for machine maintenance.
19. Analyse and make use of precision cutting systems for better accuracy.
20. Using the 8D concept identify the root cause of a problem, providing corrective solutions, and preventive solutions to eliminate the recurring problems.

Model components:

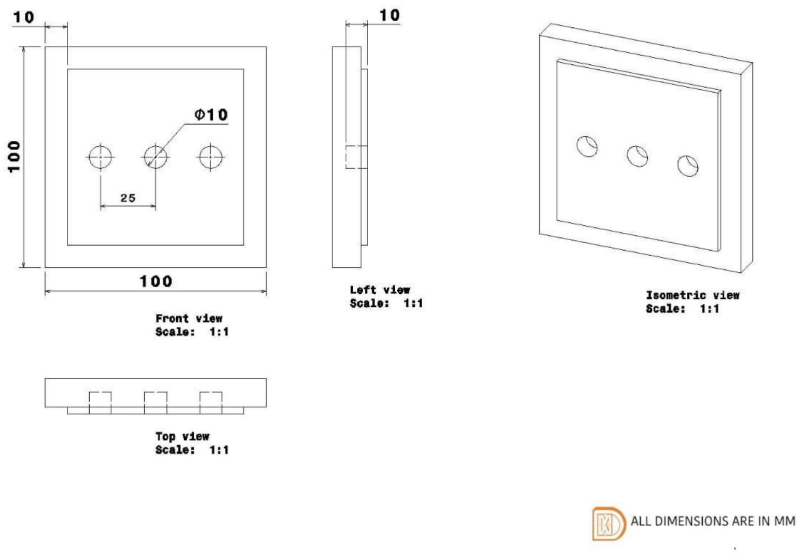


All Dimensions are in MM

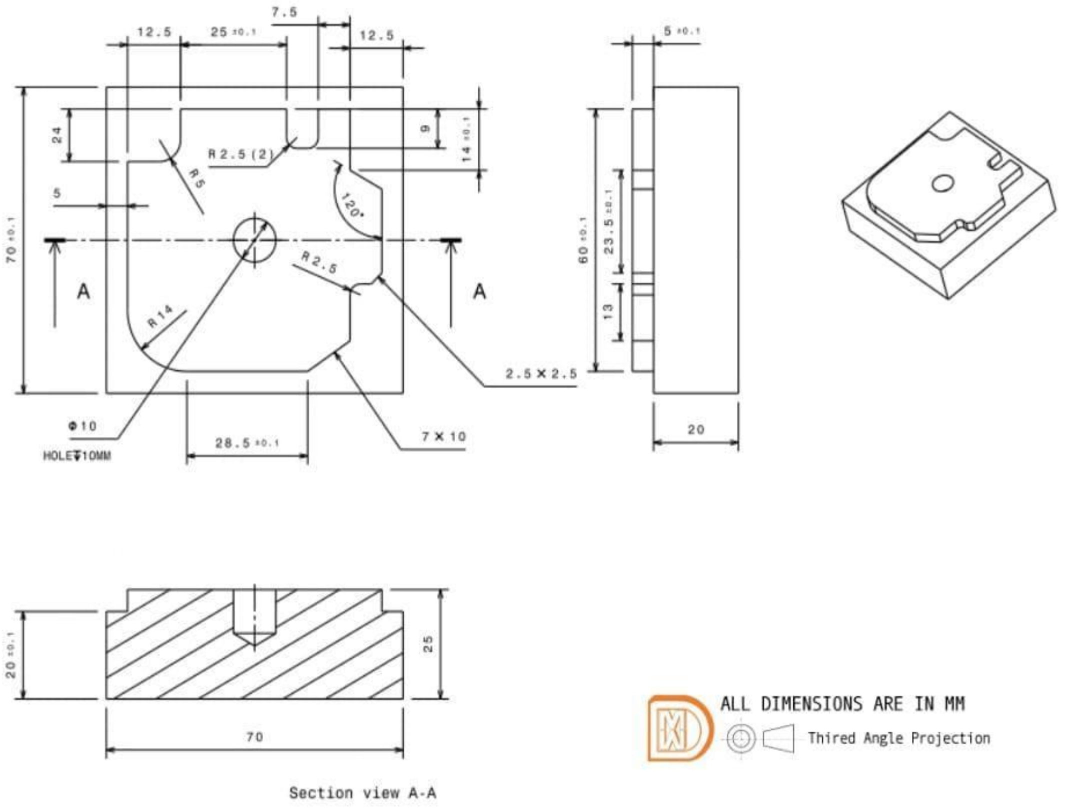


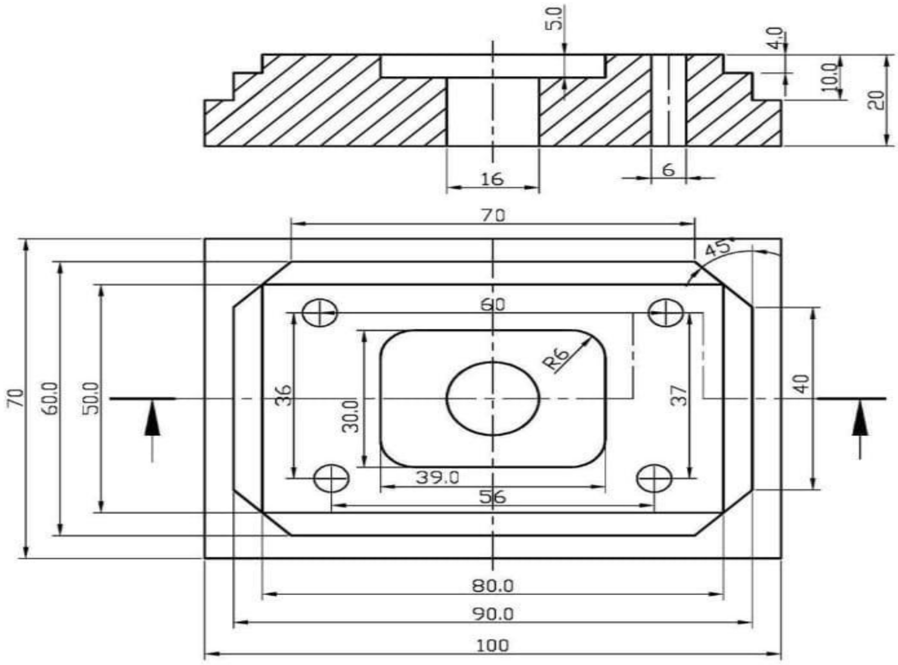


ALL DIMENSIONS ARE IN MM


10
10
100
Φ10
25
100
Front view
Scale: 1:1
Left view
Scale: 1:1
Isometric view
Scale: 1:1
Top view
Scale: 1:1
ALL DIMENSIONS ARE IN MM

CNC - MILLING


7.5
12.5
25 ±0.1
12.5
5 ±0.1
24
R 2.5 (2)
9
14 ±0.1
R 5
5
120°
70 ±0.1
A
A
R 2.5
R 14
60 ±0.1
23.5 ±0.1
13
2.5 × 2.5
Φ10
HOLE↧10MM
28.5 ±0.1
7 × 10
20
20 ±0.1
25
70
Section view A-A
ALL DIMENSIONS ARE IN MM
Thired Angle Projection



ALL DIMENSIONS ARE IN MM

FIRST ANGLE PROJECTION

SB8059	**PROGRAMMING LOGIC CIRCUIT USING SIMATIC**	**L T P C** **1 0 2 2**

COURSE OBJECTIVES:

- To introduce students to the concepts and fundamentals of Programmable Logic Controllers (PLC) in automation systems.
- To familiarize students with the programming languages used in PLC systems, particularly focusing on Siemens S7-1200 CPU and hardware.
- To provide hands-on experience in PLC communication using the TIA portal, including wiring, addressing, and commissioning tasks.
- To enable students to understand and implement bit logic instructions and structured programming using function blocks (FB), function calls (FC), and data blocks (DB).
- To explore analog signal processing and provide practical exercises for students to apply their knowledge.

COURSE CONTENT:

Unit 1: Introduction to PLC

Introduction and Classification of Automation- Introduction of PLC and Programming Languages- Introduction of Siemens S71200-CPU & Hardware.

Unit 2: TIA Portal

PLC wiring and addressing- TIA portal ad different views- Commissioning of S7-1200, and practical task

Unit 3: PLC Communication Using TIA portal

bit logic instruction and hands-on practice- Structured Programming using FB, FC and DB- Analog signal processing and hands-on Practice.

Unit 4: HMI Interfacing

Introduction, Commissioning and Interfacing of HMI- Interfacing of PLC Program with HMI using Tags- Alarms- Recipe- User Administration.

Unit 5: SCADA & PLC

Introduction of SCADA and Commissioning- Interfacing of SCADA with PLC using Tags- Screen Management

COURSE OUTCOMES:

- Students will be able to explain the concept and classification of automation and the role of PLC in industrial control systems.
- Students will acquire proficiency in programming languages and techniques specific to Siemens S7-1200 CPU and hardware.
- Students will be competent in wiring, addressing, and commissioning tasks using the TIA portal for PLC communication.
- Students will demonstrate practical skills in implementing bit logic instructions and utilizing structured programming elements like FB, FC, and DB.
- Students will be capable of processing analog signals and applying their knowledge to real-world scenarios

Software and Hardware Requirements:

- TIA Portal software (latest version) for PLC programming and commissioning.
- Siemens S7-1200 CPU and associated hardware components for hands-on practical exercises.
- HMI (Human-Machine Interface) software and hardware for interfacing with the PLC.
- SCADA (Supervisory Control and Data Acquisition) software for integrating with the PLC system.

Industry Scope:

The course provides students with the necessary skills and knowledge to pursue careers in various industries where automation systems and PLCs are extensively used, including manufacturing, process control, robotics, and energy management.

Industry used cases and their Tasks:

1. Conveyor Control System:

Task 1: Program a PLC to control the speed and direction of a conveyor belt. Task 2: Interface an HMI with the PLC to monitor and control the conveyor system.

Task 3: Implement fault detection and alarm systems for efficient operation. Task 4: Configure SCADA to visualize and analyze conveyor performance.

Task 5: Optimize the conveyor control system for maximum throughput and energy efficiency.

2. Traffic Light Control:

Task 1: Develop a PLC program to control the sequencing and timing of traffic lights.

Task 2: Design an HMI interface for monitoring and manual control of traffic lights.

Task 3: Implement communication between the PLC and HMI to display real- time traffic data.

Task 4: Integrate SCADA to monitor traffic flow and adjust signal timing based on traffic conditions.

Task 5: Evaluate the effectiveness of the traffic light control system in improving traffic efficiency and safety.

3. Water Treatment Plant Automation:

Task 1: Program a PLC to monitor and control water treatment processes, such as filtration and disinfection.

Task 2: Design an HMI interface for operators to visualize and control the water treatment plant.

Task 3: Implement communication between PLCs for interlocking and data exchange.

Task 4: Configure SCADA for remote monitoring and data analysis of water treatment parameters.

Task 5: Optimize the automation system to ensure efficient water treatment and quality compliance.

4. Packaging Line Automation:

Task 1: Develop PLC programs to control various stages of a packaging line, including filling, sealing, and labeling.

Task 2: Create an HMI interface for operators to monitor and control the packaging process.

Task 3: Implement communication between PLCs for synchronization and error handling.

Task 4: Configure SCADA to track production rates, reject rates, and overall equipment effectiveness (OEE).

Task 5: Improve the packaging line automation system for increased productivity and reduced downtime.

5. Elevator Control System:

Task 1: Program a PLC to control the operation of an elevator, including floor selection and door control.

Task 2: Design an HMI interface for users to interact with the elevator system.

Task 3: Implement communication between PLCs for multi-elevator coordination and fault handling.

Task 4: Integrate SCADA to monitor elevator performance, track usage patterns, and optimize energy consumption.

Task 5: Evaluate the elevator control system for reliability, efficiency, and passenger safety.

6. Temperature Control in HVAC:

Task 1: Develop a PLC program to control temperature setpoints, fan speed, and damper positions in an HVAC system.

Task 2: Create an HMI interface for operators to monitor and adjust HVAC parameters.

Task 3: Implement communication between the PLC and HMI for real-time data exchange.

Task 4: Configure SCADA to visualize temperature trends, energy consumption, and system alarms.

Task 5: Optimize the HVAC control system for energy efficiency and occupant comfort.

7. Pump Control in Water Distribution:

Task 1: Program a PLC to control the operation of pumps in a water distribution system based on demand.

Task 2: Design an HMI interface for operators to monitor pump status, flow rates, and pressure.

Task 3: Implement communication between PLCs for pump sequencing and fault handling.

Task 4: Configure SCADA to visualize water distribution parameters and detect anomalies.

Task 5: Improve the pump control system for efficient water management and reduced energy consumption.

8. Batch Process Automation:

Task 1: Develop PLC programs to control batch processes, such as mixing, heating, and cooling.

Task 2: Create an HMI interface for operators to monitor and control batch parameters.

Task 3: Implement communication between PLCs for recipe management and data exchange.

Task 4: Configure SCADA to track batch progress, process variables, and quality metrics.

Task 5: Optimize the batch automation system for consistent product quality and reduced cycle times.

9. Boiler Control System:

Task 1: Program a PLC to control the operation of a boiler, including temperature, pressure, and fuel flow.

Task 2: Design an HMI interface for operators to monitor boiler parameters and alarms.

Task 3: Implement communication between the PLC and HMI for real-time data visualization.

Task 4: Configure SCADA to track boiler efficiency, emissions, and safety interlocks.

Task 5: Improve the boiler control system for optimized fuel consumption and compliance with regulations.

10.Energy Monitoring and Management:

Task 1: Develop PLC programs to monitor and log energy consumption of different equipment and processes.

Task 2: Create an HMI interface for operators to visualize energy usage and identify areas of improvement.

Task 3: Implement communication between PLCs for centralized energy data collection and analysis.

Task 4: Configure SCADA to generate energy reports, set alarms for abnormal consumption, and support energy optimization strategies.

Task 5: Evaluate the energy monitoring and management system for energy savings and cost reduction.

11.Material Handling Automation:

Task 1: Program a PLC to control material handling equipment, such as conveyors, robots, and lifts.

Task 2: Design an HMI interface for operators to monitor and control material flow and equipment status.

Task 3: Implement communication between PLCs for seamless integration of different material handling systems.

Task 4: Configure SCADA to track material throughput, equipment utilization, and maintenance schedules.

Task 5: Optimize the material handling automation system for efficient production flow and reduced manual intervention.

12.Renewable Energy System Monitoring:

Task 1: Develop PLC programs to monitor and control renewable energy sources, such as solar panels and wind turbines.

Task 2: Create an HMI interface for operators to visualize real-time energy production and system performance.

Task 3: Implement communication between PLCs for data aggregation and synchronization.

Task 4: Configure SCADA to analyze renewable energy generation, track system efficiency, and detect faults.

Task 5: Improve the renewable energy system monitoring for enhanced energy utilization and maintenance planning.

13.Inventory Management:

Task 1: Program a PLC to monitor and control inventory levels in a warehouse or manufacturing facility.

Task 2: Design an HMI interface for operators to track inventory status, receive alerts, and manage stock movements.

Task 3: Implement communication between PLCs for real-time inventory data synchronization.

Task 4: Configure SCADA to visualize inventory levels, analyze consumption patterns, and generate inventory reports.

Task 5: Optimize the inventory management system for accurate stock control, reduced waste, and improved order fulfilment.

14.Waste Management and Recycling:

Task 1: Develop PLC programs to automate waste management processes, such as sorting, compaction, and recycling.

Task 2: Create an HMI interface for operators to monitor waste levels, equipment status, and recycling targets.

Task 3: Implement communication between PLCs for optimized waste routing and tracking.

Task 4: Configure SCADA to monitor waste generation rates, recycling efficiency, and environmental metrics.

Task 5: Improve the waste management system for effective recycling and waste reduction strategies.

15.Power Distribution Monitoring and Control:

Task 1: Program a PLC to monitor and control power distribution systems, including switchgear, transformers, and distribution panels.

Task 2: Design an HMI interface for operators to monitor power status, load balance, and fault conditions.

Task 3: Implement communication between PLCs for coordinated power distribution and protection.

Task 4: Configure SCADA to visualize power flow, monitor energy demand, and detect abnormalities.

Task 5: Optimize the power distribution system for efficient energy delivery, load shedding, and predictive maintenance.

16. Automated Material Testing:

Task 1: Develop PLC programs to automate material testing procedures, such as tensile testing or hardness testing.

Task 2: Create an HMI interface for operators to configure test parameters, monitor test progress, and visualize results.

Task 3: Implement communication between PLCs and testing equipment for synchronized test control and data acquisition.

Task 4: Configure SCADA to track test data, analyze test results, and generate test reports.

Task 5: Improve the automated material testing system for increased testing efficiency, accuracy, and data traceability.

17. Remote Monitoring and Control:

Task 1: Program a PLC for remote monitoring and control of industrial processes or equipment.

Task 2: Design an HMI interface for remote operators to visualize process parameters, receive alarms, and send control commands.

Task 3: Implement secure communication protocols for remote access and data transmission.

Task 4: Configure SCADA for remote data logging, trend analysis, and performance monitoring.

Task 5: Evaluate the reliability and responsiveness of the remote monitoring and control system under different network conditions.

18. Food and Beverage Production Automation:

Task 1: Develop PLC programs to automate food and beverage production processes, such as mixing, filling, and packaging.

Task 2: Create an HMI interface for operators to monitor production lines, set recipe parameters, and manage quality control.

Task 3: Implement communication between PLCs for seamless integration of different production stages.

Task 4: Configure SCADA to track production rates, material usage, and compliance with food safety standards.

Task 5: Optimize the food and beverage production automation system for increased productivity, product consistency, and reduced waste.

19. Pharmaceutical Manufacturing Automation:

Task 1: Program a PLC to automate pharmaceutical manufacturing processes, including blending, granulation, and tablet compression.

Task 2: Design an HMI interface for operators to monitor critical process parameters, track batch progress, and manage recipe changes.

Task 3: Implement communication between PLCs for recipe management and real-time data exchange.

Task 4: Configure SCADA to monitor equipment performance, track quality metrics, and comply with regulatory requirements.

Task 5: Improve the pharmaceutical manufacturing automation system for enhanced process control, product traceability, and compliance with good manufacturing practices (GMP).

20.Building Automation and Energy Management:

Task 1: Develop PLC programs to control and monitor building systems, including HVAC, lighting, and access control.

Task 2: Create an HMI interface for facility managers to visualize and adjust building parameters for occupant comfort and energy efficiency.

Task 3: Implement communication between PLCs for coordinated building system operation and energy optimization.

Task 4: Configure SCADA to analyze energy consumption, detect anomalies, and generate reports for energy management.

Task 5: Optimize the building automation system for energy savings, occupant comfort, and predictive maintenance.

SB8060 INDUSTRY 4.0 AND IT'S APPLICA TIONS IN MECHANICAL ENGINEERING

L T P C
1 0 2 2

COURSE OBJECTIVES:

- Gain a comprehensive understanding of Industry 4.0 concepts and their impact on mechanical engineering.
- Understand the architecture and components of MindSphere and its integration with MindConnect Nano and MindConnect IoT2040/2020.
- Develop practical skills in configuring, deploying, and managing Mind Connect Nano and MindConnect IoT2040/2020 devices.
- Learn data collection, analytics, and visualization techniques using MindSphere.
- Gain insights into the security and privacy considerations specific to IIoT deployments with MindSphere.

COURSE CONTENT:

UNIT 1: Introduction to Industry 4.0 and Industrial IoT (IIoT)

Overview of Industry 4.0 concepts and technologies- Role of mechanical engineering in the industry 4.0 era- Case studies of successful Industry 4.0 implementations- Overview of IIoT and its significance in industrial settings- Introduction to IoT platforms and their role in connecting devices and collecting data.

UNIT 2: Sensor Technology

Working principles of sensors and their role in converting physical quantities into electrical signals- Comparison of sensing technologies based on accuracy, sensitivity, and response time- Calibration techniques such as zero-point and span calibration-Amplification, filtering, and linearization techniques in signal conditioning, Dealing with noise and interference in sensor signals- Overview of sensor networks for data collection from multiple sensors- Communication protocols for sensor networks (Zigbee, Bluetooth, Wi-Fi)- Integration of sensors with microcontrollers, embedded systems, and IoT platforms.

UNIT 3: IIoT Gateways and Edge devices

Understanding the features and capabilities of MindSphere as an IoT operating system- Exploring the architecture and components of MindSphere- Integration of MindSphere with industrial systems and data analytics- Introduction to MindConnect Nano and MindConnect IoT2040/2020 as edge devices and gateways- Understanding their functionalities, connectivity options, and protocols supported- Configuring and deploying MindConnect Nano and MindConnect IoT2040/2020 devices.

UNIT 4: Data Collection and Analytics with MindSphere

Leveraging MindSphere for data collection from edge devices and gateways- Introduction to data analytics and visualization capabilities in MindSphere- Implementing data-driven insights and decision-making using MindSphere. **UNIT 5: MindSphere Implementation**

Practical exercises using a simulated MindSphere environment- Data visualization and analysis using MindSphere's tools and features- Basic programming concepts for interacting with MindSphere APIs

2. Energy Monitoring and Optimization

Task 1: Install IoT sensors to collect energy consumption data.

Task 2: Implement a data analytics platform to analyze energy usage patterns.

Task 3: Identify energy-saving opportunities and develop optimization strategies.

Task 4: Set up real-time monitoring of energy consumption and generate alerts for anomalies.

Task 5: Evaluate the impact of energy optimization initiatives on cost reduction and environmental sustainability.

3. Inventory Management and Tracking

Task 1: Deploy RFID or barcode systems for inventory tracking.

Task 2: Develop a centralized database for real-time inventory management.

Task 3: Implement automated replenishment systems based on inventory levels.

Task 4: Analyze historical data to optimize inventory levels and reduce stockouts.

Task 5: Evaluate the accuracy and efficiency of the inventory management system in improving supply chain operations.

4. Quality Control and Defect Detection

Task 1: Implement IoT sensors for real-time data collection during the manufacturing process.

Task 2: Develop machine learning algorithms to detect defects and anomalies.

Task 3: Create automated alerts for quality control teams based on detected defects.

Task 4: Analyze data to identify root causes of defects and implement corrective actions.

Task 5: Evaluate the effectiveness of the quality control system in reducing defects and improving product quality.

5. Supply Chain Visibility and Optimization

Task 1: Implement IoT-enabled tracking devices for shipment monitoring.

Task 2: Develop a centralized platform to track and analyze supply chain data.

Task 3: Optimize routes and transportation schedules based on real-time data.

Task 4: Implement predictive analytics to anticipate potential disruptions in the supply chain.

Task 5: Evaluate the impact of improved supply chain visibility on delivery times and customer satisfaction.

6. Remote Equipment Monitoring and Control

Task 1: Install IoT sensors to collect real-time data from remote equipment.

Task 2: Develop a remote monitoring system for equipment performance and health.

Task 3: Implement remote control capabilities for equipment operation and adjustments.

Task 4: Set up automated alerts for critical conditions or malfunctions.

Task 5: Evaluate the effectiveness of remote monitoring and control in reducing maintenance costs and improving equipment uptime.

7. Worker Safety and Health Monitoring

Task 1: Deploy wearable IoT devices to monitor worker health and safety parameters.

Task 2: Develop real-time analytics to detect potential safety hazards.

Task 3: Create automated alerts for unsafe conditions or incidents.

Task 4: Implement a centralized dashboard for monitoring worker safety and health metrics.

Task 5: Evaluate the impact of IoT-based safety measures on reducing accidents and promoting worker well-being.

8. Asset Tracking and Management

Task 1: Implement IoT-based asset tracking devices for real-time location monitoring.

Task 2: Develop an asset management system for inventory and utilization tracking.

Task 3: Optimize asset allocation and utilization based on data analytics.

Task 4: Implement automated alerts for asset maintenance and calibration schedules.

Task 5: Evaluate the efficiency and cost savings achieved through improved asset tracking and management.

9. Production Line Optimization

Task 1: Implement IoT sensors to collect real-time data from production line equipment.

Task 2: Develop analytics models to identify bottlenecks and optimize production flow.

Task 3: Implement automated scheduling and routing systems for production orders.

Task 4: Analyze data to identify opportunities for process improvement and waste reduction.

Task 5: Evaluate the impact of production line optimization on throughput, cycle times, and overall productivity.

10.Remote Monitoring and Troubleshooting of Equipment

Task 1: Deploy IoT sensors to collect real-time data from equipment in remote locations.

Task 2: Develop a remote monitoring system to track equipment performance and health.

Task 3: Implement remote troubleshooting capabilities to address equipment issues.

Task 4: Set up automated alerts for critical conditions or failures.

Task 5: Evaluate the effectiveness of remote monitoring and troubleshooting in minimizing downtime and reducing maintenance costs.

11.Product Lifecycle Management

Task 1: Implement IoT sensors for collecting data throughout the product lifecycle.

Task 2: Develop a centralized platform for tracking and analyzing product data.

Task 3: Implement real-time collaboration and information sharing among stakeholders.

Task 4: Analyze data to identify opportunities for product improvements and cost reduction.

Task 5: Evaluate the impact of product lifecycle management strategies on product quality and time-to-market.

12.Condition Monitoring and Fault Diagnosis

Task 1: Deploy IoT sensors to collect real-time data from critical equipment.

Task 2: Develop machine learning algorithms to detect equipment faults and anomalies.

Task 3: Create automated alerts for maintenance teams based on fault detection.

Task 4: Implement remote monitoring capabilities to assess equipment health in real-time.

Task 5: Evaluate the effectiveness of condition monitoring and fault diagnosis in reducing unplanned downtime and improving equipment reliability.

13.Real-time Production Data Visualization

Task 1: Integrate data from various production systems and sensors.

Task 2: Develop real-time dashboards and visualizations for production data.

Task 3: Implement data analytics to monitor key production metrics and performance indicators.

Task 4: Set up automated alerts for deviations from production targets.

Task 5: Evaluate the impact of real-time data visualization on production efficiency and decision-making.

14.Workflow Automation and Optimization

Task 1: Identify manual and repetitive **Task**s in existing workflows.

Task 2: Automate workflows using IoT and IIoT technologies.

Task 3: Implement data-driven decision-making processes in workflow optimization.

Task 4: Analyze data to identify bottlenecks and areas for process improvement.

Task 5: Evaluate the efficiency gains achieved through workflow automation and optimization.

15.Environmental Monitoring and Compliance

Task 1: Deploy IoT sensors to collect real-time environmental data.

Task 2: Develop analytics models to monitor and analyze environmental parameters.

Task 3: Implement automated alerts for non-compliance or abnormal conditions.

Task 4: Set up a centralized platform for environmental data management and reporting.

Task 5: Evaluate the effectiveness of environmental monitoring systems in ensuring regulatory compliance and reducing environmental risks.

16.Digital Twin Implementation

Task 1: Create a virtual model of a physical asset or process.

Task 2: Integrate real-time data from IoT sensors into the digital twin.

Task 3: Implement data analytics to compare real-time data with the digital twin model.

Task 4: Use the digital twin for simulation, optimization, and predictive analysis.

Task 5: Evaluate the benefits of digital twin implementation in terms of improved operational efficiency, reduced costs, and enhanced decision-making.

17.Cybersecurity and Data Protection

Task 1: Identify potential cybersecurity threats and vulnerabilities.

Task 2: Develop and implement security measures, such as authentication and encryption.

Task 3: Conduct penetration testing to assess system vulnerabilities.

Task 4: Implement data protection strategies, including data encryption and access controls.

Task 5: Evaluate the effectiveness of cybersecurity measures in safeguarding sensitive data and protecting against cyber threats.

18. Smart Grid Management

Task 1: Implement IoT devices for real-time monitoring of power generation and consumption.
Task 2: Develop analytics models for load forecasting and energy optimization.
Task 3: Implement demand response strategies to manage peak loads and grid stability.
Task 4: Integrate renewable energy sources and optimize their integration into the grid.
Task 5: Evaluate the impact of smart grid management on energy efficiency, grid stability, and cost savings.

19. Autonomous Vehicles and Transportation Optimization

Task 1: Develop autonomous vehicle control algorithms using sensor data.
Task 2: Implement vehicle-to-vehicle and vehicle-to-infrastructure communication protocols.
Task 3: Analyze traffic data to optimize route planning and congestion management.
Task 4: Evaluate the safety and efficiency of autonomous vehicle operations.
Task 5: Assess the potential environmental benefits and economic impacts of autonomous transportation systems.

20. Cloud-based Manufacturing and Collaboration

Task 1: Implement cloud-based platforms for collaborative product design and development.
Task 2: Enable real-time sharing of design files, simulations, and manufacturing instructions.
Task 3: Implement version control and document management systems.
Task 4: Analyze data to identify opportunities for process improvement and cost reduction.
Task 5: Evaluate the effectiveness of cloud-based manufacturing and collaboration in accelerating product development and improving collaboration among teams.

SB8061	3D PRINTING USING NX	L T P C 1 0 2 2

COURSE OBJECTIVES:

- To understand the principles and fundamentals of 3D printing technology.
- To develop proficiency in designing and creating 3D models using CAD software.
- To gain hands-on experience in operating and maintaining 3D printers.
- To explore the applications and potential of 3D printing in various industries.
- To foster creativity and problem-solving skills through practical projects.’

Course Contents :

Unit 1: Introduction to 3D Printing

Overview of 3D printing technology and its historical background- Different types of 3D printers and their working principles- Introduction to CAD software and its role in 3D modeling- Basic concepts of 3D modeling and designing for 3D printing.

Unit 2: CAD Design for 3D Printing

In-depth exploration of Siemens CAD NX software- Understanding the user interface and basic tools of CAD NX. - Creating and modifying 3D models for 3D printing- Design optimization for 3D printing, including considerations for support structures and material properties- best practices for designing functional and printable objects.

Unit 3: 3D Printing Technologies

Detailed study of various 3D printing technologies, such as FDM, SLA, SLS, and more-

Advantages, limitations, and applications of each technology- Selection criteria for choosing the appropriate 3D printing technology based on the desired outcomes- post-processing techniques for improving the quality and aesthetics of 3D printed objects.

Unit 4: 3D Printing Workflow and Operation

Understanding the complete workflow of 3D printing, from design to final product- Preparing 3D models for printing, including file formats and slicing - Operating and calibrating 3D printers for optimal results- Troubleshooting common issues during the printing process- Maintenance and safety protocols for 3D printers.

Unit 5: Applications and Advanced Topics

Exploring the cutting-edge applications of 3D printing in fields like medicine, architecture, fashion, and electronics- Advanced topics in 3D printing research, such as bio fabrication and nanoscale printing- Intellectual property issues and legal considerations in 3D printing- Industry 4.0 integration with 3D printing, including digital twin technology and smart manufacturing- Future trends and developments in 3D printing, such as 4D printing and self-assembling structures**.**

TOTAL : 45 PERIODS

Software and Hardware Requirements :

- Siemens CAD NX software
- Access to a 3D printer
- Computer with minimum system requirements to run CAD NX software
- Basic peripherals like mouse, keyboard, and display

References :

- "The 3D Printing Handbook: Technologies, design, and applications" by Ben Redwood, Filemon Schöffer, and Brian Garret
- "Design for 3D Printing: Scanning, Creating, Editing, Remixing, and Making in Three Dimensions" by Samuel N. Bernier, Bertier Luyt, and Tatiana Reinhard
- "3D Printing for Dummies" by Kalani Kirk Hausman and Richard Horne

Industry Scope :

3D printing has a vast scope across industries such as healthcare, automotive, aerospace, architecture, consumer products, education, and more. It is revolutionizing manufacturing processes, enabling rapid prototyping, customization, and cost-effective production of complex parts and products.

20 Industry Used Cases and their Tasks:

1. Design and print a customized smartphone case with unique features.

Task 1: Sketch and conceptualize the design.

Task 2: Create a 3D model using CAD software. Task 3: Optimize the design for 3D printing.

Task 4: Print the smartphone case using appropriate materials. Task 5: Evaluate the fit and functionality of the printed case.

2. Develop a wearable fitness tracker band with integrated sensors.

Task 1: Research and select suitable sensors for the tracker.

Task 2: Design a comfortable and adjustable wristband.

Task 3: Integrate the sensors into the wristband design.

Task 4: Print the wristband using flexible and durable materials.

Task 5: Test the functionality and accuracy of the fitness tracker.

3. **Create a personalized keychain with intricate designs.**

Task 1: Design and sketch unique patterns or shapes for the keychain.

Task 2: Digitize the designs and create a 3D model.

Task 3: Optimize the model for 3D printing, considering size and detail.

Task 4: Print the keychain using appropriate materials and colors.

Task 5: Evaluate the quality and aesthetic appeal of the printed keychain.

4. **Manufacture a miniature architectural model of a famous landmark.**

Task 1: Research the landmark and gather reference images or blueprints.

Task 2: Use CAD software to create a scaled-down 3D model of the landmark.

Task 3: Print the model in multiple parts and assemble them.

Task 4: Apply finishing touches, such as painting or texturing.

Task 5: Present the final model with accurate details and proportions.

5. **Design and print a functional and ergonomic desk organizer.**

Task 1: Analyze the storage requirements and functionality of a desk organizer.

Task 2: Sketch and design compartments and slots for different items.

Task 3: Model the design in CAD software, considering ease of use.

Task 4: Print the organizer with appropriate materials and surface finishes.

Task 5: Evaluate the usability and effectiveness of the printed organizer.

6. **Create a customized jewelry piece, such as a pendant or bracelet.**

Task 1: Explore different jewelry designs and inspirations.

Task 2: Design a unique jewelry piece using CAD software.

Task 3: Optimize the design for 3D printing, considering aesthetics and wearability.

Task 4: Print the jewelry using high-quality materials, such as precious metals or resin.

Task 5: Polish and refine the printed piece for a professional finish.

7. **Develop a functional drone frame with optimized weight and aerodynamics.**

Task 1: Research existing drone frames and analyze their pros and cons.

Task 2: Design a lightweight and sturdy drone frame using CAD software.

Task 3: Optimize the design for weight reduction and aerodynamic efficiency.

Task 4: Print the drone frame using strong and lightweight materials, such as carbon fiber.

Task 5: Test the printed frame's performance in flight and evaluate its stability and durability.

8. **Design and print a custom-made prosthetic hand or arm prototype.**

Task 1: Study existing prosthetic designs and understand their functionality.

Task 2: Design a prosthetic hand or arm with adjustable features.

Task 3: Model the design in CAD software, ensuring comfortable fit and range of motion.

Task 4: Print the prototype using flexible and lightweight materials.

Task 5: Evaluate the functionality and usability of the printed prototype.

9. **Create a scale model of a race car with detailed interior and exterior features.**

Task 1: Research the specifications and design elements of race cars.

Task 2: Create a detailed 3D model of the race car using CAD software.

Task 3: Optimize the model for 3D printing, considering size and detail.

Task 4: Print the model in multiple parts and assemble them accurately.

Task 5: Present the final model with accurate proportions and realistic features.

10. **Develop a prototype of a sustainable packaging solution.**

Task 1: Identify the packaging requirements and environmental goals.

Task 2: Design an innovative and eco-friendly packaging solution.

Task 3: Model the design in CAD software, considering functionality and materials.

Task 4: Print the prototype using bio-based or recyclable materials.

Task 5: Evaluate the usability, durability, and sustainability of the printed prototype.

11. **Design and print a custom-made smartphone stand with adjustable angles.**

Task 1: Research existing smartphone stand designs and analyze their functionality.

Task 2: Design a versatile smartphone stand with adjustable angles and stability.

Task 3: Model the design in CAD software, ensuring compatibility with different phone sizes.

Task 4: Print the smartphone stand using durable and lightweight materials.

Task 5: Test the printed stand with different phone models and evaluate its usability and stability.

12. **Create a prototype of a solar-powered charger for mobile devices.**

Task 1: Research solar energy and its applications in charging devices.

Task 2: Design a compact and portable solar-powered charger with integrated battery storage.

Task 3: Model the design in CAD software, considering the positioning of solar panels and connectivity options.

Task 4: Print the prototype using lightweight and weather-resistant materials.

Task 5: Test the functionality and efficiency of the printed charger with various mobile devices.

13.Develop a customized handle or grip for sports equipment, such as a tennis racket or golf club.

Task 1: Analyze the ergonomic requirements and grip preferences for the specific sports equipment.

Task 2: Design a comfortable and non-slip handle or grip using CAD software.

Task 3: Optimize the design for 3D printing, considering texture and material properties.

Task 4: Print the handle or grip using materials that offer enhanced grip and durability.

Task 5: Evaluate the comfort, grip, and performance of the printed handle or grip during sports activities.

14.Create a unique puzzle or brain teaser toy with intricate interlocking pieces.

Task 1: Explore different types of puzzles and brain teasers for inspiration.

Task 2: Design an original and challenging puzzle with interlocking pieces.

Task 3: Model the design in CAD software, ensuring precision and proper fit.

Task 4: Print the puzzle using suitable materials and colors for aesthetics.

Task 5: Test the difficulty level and user experience of the printed puzzle.

15.Design and print a decorative lampshade with intricate patterns and lighting effects.

Task 1: Explore various lampshade designs and lighting techniques.

Task 2: Design a unique and visually appealing lampshade with intricate patterns.

Task 3: Model the design in CAD software, considering light diffusion and structural integrity.

Task 4: Print the lampshade using translucent materials that enhance lighting effects.

Task 5: Evaluate the aesthetics and functionality of the printed lampshade when illuminated.

16.Develop a miniature greenhouse for indoor gardening or plant propagation.

Task 1: Research the requirements for successful indoor gardening and plant propagation.

Task 2: Design a compact and efficient greenhouse with ventilation and adjustable shelves.

Task 3: Model the design in CAD software, considering space utilization and plant care.

Task 4: Print the greenhouse using materials that provide suitable insulation and light transmission.

Task 5: Test the functionality and effectiveness of the printed greenhouse for plant growth and maintenance.

17.Create a functional and collapsible storage solution for camping or outdoor activities.

Task 1: Analyze the storage needs and portability requirements for camping equipment.

Task 2: Design a versatile and collapsible storage solution with compartments and carrying handles.

Task 3: Model the design in CAD software, ensuring ease of assembly and disassembly.

Task 4: Print the storage solution using lightweight and durable materials.

Task 5: Evaluate the usability and durability of the printed storage solution during camping or outdoor activities.

18.Design and print a custom-made headphone stand or holder with cable management features.

Task 1: Research existing headphone stand designs and cable management solutions.

Task 2: Design a headphone stand or holder that offers proper support and cable organization.

Task 3: Model the design in CAD software, considering aesthetics and stability.

Task 4: Print the headphone stand using sturdy and stylish materials.

Task 5: Evaluate the functionality and convenience of the printed headphone stand.

19.Develop a prototype of a smart home device, such as a voice-activated assistant or temperature controller.

Task 1: Research smart home technologies and their applications.
Task 2: Design a user-friendly and aesthetically pleasing smart home device.
Task 3: Model the design in CAD software, ensuring integration of necessary sensors and connectivity options.

Task 4: Print the prototype using materials that provide a professional and durable finish.

Task 5: Test the functionality and interaction capabilities of the printed smart home device.

20.Create a customized board game with unique playing pieces and interactive elements.

Task 1: Brainstorm a game concept and mechanics that offer engaging gameplay.

Task 2: Design a game board and playing pieces with customized shapes and artwork.

Task 3: Model the design in CAD software, ensuring compatibility with the gameplay mechanics.

Task 4: Print the game components using materials that offer durability and visual appeal.

Task 5: Playtest the printed board game, evaluating the gameplay experience and rule clarity.

SB8062 ELECTRIC VEHICLE TECHNOLOGY AND MANUFACTURING L T P C
1 0 2 2

Course Objectives:

- To provide a comprehensive understanding of electric vehicle architecture, components, and subsystems.
- To familiarize students with the assembly processes involved in manufacturing electric vehicles.
- To equip students with the knowledge and skills to conduct testing, validation, and compliance procedures for electric vehicles.
- To introduce students to the principles of lean manufacturing and process optimization in the context of electric vehicle production.
- To enable students to understand the maintenance and service requirements of electric vehicles and develop effective troubleshooting and customer support strategies.

Course Content:

Unit 1: Electric Vehicle Components and Subsystems
Introduction to electric vehicle architecture and components- Electric motor and drivetrain systems in EVs- Battery technologies and energy management systems- Charging systems and infrastructure for EVs

Unit 2: Electric Vehicle Assembly Processes

Overview of electric vehicle manufacturing processes- Body and chassis assembly techniques for EVs- Integration of electric power train components- Quality control and inspection in EV assembly

Unit 3: Electric Vehicle Testing and Validation

Test planning and procedures for electric vehicles- Performance testing and characterization of EV subsystems- Safety and reliability testing for EVs- Regulatory compliance and certification for electric vehicles

Unit 4: Lean Manufacturing and Process Optimization

Principles of lean manufacturing in the context of EV production- Value stream mapping and waste reduction techniques- Process optimization and continuous improvement in EV manufacturing- Ergonomics and workplace design considerations

Unit 5: Electric Vehicle Maintenance and Service

Electric vehicle maintenance requirements and best practices- Troubleshooting and diagnostics for EV systems- Safety considerations in EV maintenance and service- Customer service and EV user support

TOTAL : 45 PERIODS

Course Outcome:

- Demonstrate a thorough understanding of electric vehicle architecture andthe function of various components and subsystems.
- Apply appropriate assembly techniques for electric vehicle body and chassis assembly and integrate electric powertrain components effectively.
- Develop test plans and conduct performance testing, safety testing, and reliability testing for electric vehicle subsystems.
- Apply lean manufacturing principles to identify and reduce waste, optimize processes, and implement continuous improvement strategies inelectric vehicle manufacturing.
- Demonstrate knowledge of electric vehicle maintenance requirements, perform troubleshooting and diagnostics for EV systems, and provide effective customer service and support to EV users.

Reference Links:

- Electric Vehicle Technology Explained, Author: James Larminie and John Lowry, Publisher: Wiley, Year: 2012, ISBN: 978-1119973855
- Electric Vehicle Battery Systems, Author: Sandeep Dhameja, Publisher: Elsevier, Year: 2019, ISBN: 978-0128132495
- Electric and Hybrid Vehicles: Design Fundamentals, Second Edition,Author: Iqbal Husain, Publisher: CRC Press, Year: 2018, ISBN: 978- 1466560837
- Electric Vehicle Technology Explained: Enabling Technologies,Infrastructure, and Applications, Author: James Larminie and John Lowry,Publisher: Wiley, Year: 2012, ISBN: 978-0470666031
- Electric Vehicle Manufacturing and Assembly ,Author: Gary K. Richardson, Publisher: SAE International, Year: 2015, ISBN: 978- 0768079931
- Electric Vehicle Technology and Maintenance, Author: Jan Wiedemann, Publisher: Springer,Year: 2020, ISBN: 978-3030464114
- Lean Assembly: The Nuts and Bolts of Making Assembly Operations Flow, Author: Michel Baudin, Publisher: CRC Press,Year: 2015,ISBN: 978- 1498719762

Software and Hardware Requirements:

- CAD software for electric vehicle design and assembly simulations (e.g., NX-CAD, Tecnomatix)
- Simulation software for electric powertrain modelling (e.g., AMESIM)

- FEA testing software (e.g., Simcenter 3D)
- EV kit.

Industry Scope:

The course provides knowledge and skills that are applicable in the automotive industry, electric vehicle manufacturing companies, research and development organizations, and service centers specializing in electric vehicles. Mechanical engineering professionals in these industries can leverage the concepts and technologies covered in the course to contribute to the design, production, testing, and maintenance of electric vehicles.

20 Industry Use Cases:

1. Assembling electric motors and drivetrains for electric bicycles:
Task 1: Design and build a small-scale electric bicycle prototype using off-the- shelf components.
Task 2: Compare and evaluate different motor and drivetrain configurationsfor electric bicycles.
Task 3: Create a step-by-step guide on assembling an electric bicycledrivetrain system.
Task 4: Conduct performance tests to measure the speed and torque of various electric bicycle motors.
Task 5: Investigate the impact of different gear ratios on the efficiency andrange of an electric bicycle.

2. Integrating battery packs and energy management systems in electric scooters:
Task 1: Design and build a battery pack for an electric scooter using rechargeable lithium-ion cells.
Task 2: Develop an energy management system to optimize battery usageand extend the range of an electric scooter.
Task 3: Evaluate the performance and efficiency of different battery chemistries for electric scooters.
Task 4: Implement a charging algorithm for the battery pack to maximize its lifespan.
Task 5: Conduct experiments to compare the range and acceleration ofelectric scooters with different battery configurations.

3. Testing the range and performance of electric golf carts:

Task 1: Design and conduct a range test for electric golf carts using different driving conditions and terrains.
Task 2: Measure and compare the power consumption of electric golf cartsat different speeds.
Task 3: Analyze the impact of battery capacity and weight on the range and performance of electric golf carts.
Task 4: Investigate the effect of aerodynamics on the energy efficiency of electric golf carts.
Task 5: Develop a simulation model to predict the range of electric golf cartsbased on various factors.

4. Validating the safety and functionality of electric tricycles:

Task 1: Perform a safety inspection on electric tricycles to ensure compliance with relevant regulations and standards.
Task 2: Conduct a braking performance test to evaluate the stopping distanceand stability of electric tricycles.
Task 3: Assess the visibility and effectiveness of lighting systems on electric tricycles.
Task 4: Analyze the stability and handling characteristics of electric tricycles through controlled maneuvers.
Task 5: Investigate the impact of different suspension setups on the comfortand safety of electric tricycles.

5. Implementing lean manufacturing principles to streamline the production of electric motorcycles:

Task 1: Conduct a value stream mapping analysis to identify waste and inefficiencies in the production process of electric motorcycles.
Task 2: Implement 5S workplace organization techniques to improve the cleanliness and organization of the assembly line.
Task 3: Optimize the layout of the production area to reduce material handling and increase workflow efficiency.
Task 4: Apply just-in-time (JIT) principles to minimize inventory and reduce production lead time for electric motorcycles.
Task 5: Implement visual management tools, such as Kanban boards, to improve communication and coordination between assembly line workers.

6. Optimizing the assembly process for electric skateboards.

Task 1: Design and implement a streamlined assembly line layout for electric skateboard production, considering ergonomic factors and efficient workflow.
Task 2: Develop a standardized checklist and quality control procedures to ensure the accurate installation of components during the skateboardassembly process.
Task 3: Evaluate and improve the efficiency of the battery installation process, exploring methods such as automation or specialized tools.
Task 4: Investigate different adhesive options for attaching grip tape to electric skateboard decks, focusing on ease of application and durability.
Task 5: Implement a lean manufacturing approach by identifying and eliminating waste in the assembly process, such as reducing unnecessary motion or
inventory.

7. Conducting quality control inspections for electric off-road vehicles.
Task 1: Develop a comprehensive quality control checklist to ensure all components of the electric off-road vehicle are inspected thoroughlybefore delivery.
Task 2: Create a testing protocol to assess the performance and durability of
critical off-road vehicle subsystems, such as the suspension or drivetrain.
Task 3: Implement a visual inspection process using imaging technology to identify any defects or inconsistencies in the vehicle's body orpaintwork.
Task 4: Establish a data logging system to monitor and analyze key performance metrics of the off-road vehicle during quality control testing.
Task 5: Investigate and implement non-destructive testing methods, such as ultrasound or thermography, to detect any hidden defects or abnormalities in vehicle components.

8. Troubleshooting and diagnosing issues in electric-powered lawnmowers.
Task 1: Create a diagnostic flowchart or decision tree to guide technicians in identifying common electrical and mechanical issues in electric lawnmowers.
Task 2: Develop a step-by-step troubleshooting guide for diagnosing and resolving battery-related problems in electric lawnmower systems.
Task 3: Design and implement a test setup to assess the motor performance and control system functionality in electric lawnmowers.
Task 4: Investigate and document the most common failure modes in electric lawnmowers, along with recommended repair procedures for each issue.
Task 5: Collaborate with lawnmower manufacturers to analyze warranty claims data and identify recurring problems, leading to proactive measures for preventing those issues in future models.

9. Performing maintenance and service Task on electric-powered wheelchairs.
Task 1: Develop a comprehensive maintenance schedule for electric wheel chairs, outlining routine Task such as battery inspection, tire replacement, and control system calibration.
Task 2: Design and implement a battery management system that monitors and optimizes the charging and discharging cycles of electric wheelchair batteries to extend their lifespan.
Task 3: Create a user-friendly troubleshooting guide for common issues faced by wheelchair users, providing step-by-step instructions for identifying and resolving problems.
Task 4: Develop a training program for wheelchair users and caregivers, covering basic

maintenance Task such as cleaning, lubrication, andadjusting seating positions.
Task 5: Investigate and recommend ergonomic enhancements to improve user comfort and reduce the risk of musculoskeletal disorders associated with operating electric wheelchairs.

10.Testing the charging infrastructure for electric buses at a transit depot.

Task 1: Conduct a comprehensive site survey to assess the electrical capacityand compatibility of the transit depot for accommodating the charging infrastructure required for electric buses.
Task 2: Develop a testing protocol to evaluate the reliability and performance of charging stations, including factors such as charging speed, efficiency, and interoperability with different bus models.
Task 3: Create a monitoring system to collect and analyze charging data, allowing for the identification of any inefficiencies or anomalies in the charging process.
Task 4: Investigate the feasibility of implementing smart charging solutions,such as demand response systems or vehicle-to-grid integration, to optimize the energy usage and minimize the peak load on the electrical grid.
Task 5: Collaborate with bus manufacturers and charging equipment suppliers to ensure compatibility and optimize the integration between the electric buses and the charging infrastructure at the transit depot.

11.Validating the performance and efficiency of electric delivery trucks:

Task 1: Analyze the energy consumption of different delivery routes and propose optimization strategies.
Task 2: Develop a data logging system to measure and compare the performance of electric delivery trucks under various load conditions.

Task 3: Conduct a comparative study on the range and efficiency of different electric delivery truck models available in the market.

Task 4: Design and implement a simulation model to assess the impact of driving behavior on the energy consumption of electric delivery trucks.

Task 5: Create a performance benchmarking tool to evaluate the acceleration, braking, and overall efficiency of electric delivery trucks.

12.Optimizing the manufacturing process for electric-powered drones:

Task 1: Investigate different assembly line layouts and identify the most efficient configuration for drone production.
Task 2: Develop a quality control system to monitor the manufacturing process and identify areas for improvement.
Task 3: Implement lean manufacturing techniques, such as Kanban systemsor value stream mapping, to optimize the production flow of electricdrones.
Task 4: Study the feasibility of automation and robotics in drone manufacturing and propose ways to streamline the process.
Task 5: Conduct a cost analysis to identify potential cost-saving measures inthe manufacturing of electric-powered drones.

13.Conducting safety inspections on electric-powered forklifts:

Task 1: Develop a comprehensive checklist for inspecting electric forklifts, covering critical safety components and systems.
Task 2: Create a training program for forklift operators on pre-operation safety checks and preventive maintenance procedures.
Task 3: Conduct a risk assessment to identify potential safety hazards associated with electric

forklift operation and suggest mitigation strategies.
Task 4: Investigate the effectiveness of different safety technologies, such as proximity sensors or backup cameras, in preventing accidents involving electric forklifts.
Task 5: Design a data collection system to monitor and analyze safetyincidents related to electric forklifts, aiming to identify trends and implement corrective measures.

14.Troubleshooting and repairing electric-powered recreational vehicles (RVs):
Task 1: Develop a diagnostic tool or checklist to identify common electricalissues in electric RVs.
Task 2: Create a maintenance schedule for electric RVs, including battery health checks, motor inspections, and electrical system maintenance.
Task 3: Conduct a study on the most common causes of electric RV failuresand propose preventive measures.
Task 4: Investigate the compatibility and performance of different charging stations for electric RVs and provide recommendations.
Task 5: Develop a guide for troubleshooting and repairing specific electric components in RVs, such as inverters, solar panels, or electric heating systems.

15.Testing the reliability and durability of electric-powered boats:
Task 1: Design and implement a series of stress tests to evaluate the performance and durability of electric boat batteries under differentoperating conditions.
Task 2: Conduct a comparative study on the reliability of electric propulsion systems versus traditional combustion engines for boats.
Task 3: Develop a water-resistant sealing system for electric boat components to ensure their reliability in harsh marine environments.
Task 4: Create a testing protocol to assess the impact of charging cycles on the longevity of electric boat batteries.
Task 5: Investigate the effects of different environmental factors, such as water salinity or temperature, on the performance and reliability of electric-powered boats.

16.Conducting maintenance and service on electric-powered jet skis:
Task 1: Developing a maintenance checklist for electric-powered jet skis, including Task such as battery inspection, electrical system diagnostics, and motor lubrication.
Task 2: Designing a preventive maintenance schedule for electric-powered jet skis, outlining recommended intervals for battery charging, component inspections, and software updates.
Task 3: Creating a troubleshooting guide for common issues in electric- powered
jet skis, including step-by-step procedures for diagnosing and resolving electrical or mechanical problems.
Task 4: Developing a battery management system for electric-powered jet skis, focusing on monitoring battery health, optimizing charging cycles, and implementing safety measures.
Task 5: Conducting a comparative study on different maintenance techniques for electric-powered jet skis, evaluating the effectivenessof regular maintenance versus reactive repairs.

17.Validating the charging efficiency of electric-powered streetlights:
Task 1: Designing an experimental setup to measure and compare the charging efficiency of different electric-powered streetlights using various charging technologies.
Task 2: Analyzing the energy consumption patterns of electric-powered streetlights in different scenarios, such as peak and off-peak hours, and identifying areas for optimization.
Task 3: Investigating the impact of environmental factors, such as temperature and humidity, on the charging efficiency of electric- powered streetlights and proposing solutions to mitigate any performance issues.
Task 4: Developing a monitoring system to track the charging efficiency and overall energy consumption of electric-powered streetlights, providing real-time data for analysis and optimization.
Task 5: Conducting a field study to validate the charging efficiency of electric- powered streetlights in a real-world setting, comparing the results with theoretical calculations and industry standards.

18.Optimizing the assembly process for electric-powered rickshaws:
Task 1: Analyzing the existing assembly process for electric-powered rickshaws and identifying

areas of inefficiency or bottlenecks.
Task 2: Implementing lean manufacturing principles, such as value stream mapping and waste reduction techniques, to streamline the assemblyprocess and improve overall productivity.
Task 3: Designing ergonomic workstations and tools specifically tailored forthe
assembly of electric-powered rickshaws, considering factors such as worker comfort and safety.
Task 4: Developing standardized work instructions and training materials to ensure consistency and quality throughout the assembly process of electric- powered rickshaws.
Task 5: Conducting time-motion studies to optimize the sequencing of assembly Task and identify opportunities for automation or processsimplification in the assembly of electric-powered rickshaws.

19.Conducting performance tests on electric-powered go-karts:
Task 1: Designing a performance testing track or course specifically for electric- powered go-karts, considering factors such as speed, acceleration, and handling characteristics.
Task 2: Collecting data through instrumentation and sensors to measure and analyze the performance metrics of electric-powered go-karts, including lap times, power output, and energy consumption.
Task 3: Comparing the performance of different electric-powered go-karts by conducting controlled experiments and benchmarking tests.
Task 4: Investigating the effects of variables such as battery capacity, motor power, and weight distribution on the performance of electric-powered go-karts, and optimizing these factors for improved performance.
Task 5: Conducting durability tests on electric-powered go-kart components, such as batteries, motors, and chassis, to assess their reliability and identify any performance limitations.

20.Troubleshooting and diagnosing issues in electric-powered food trucks:
Task 1: Developing a comprehensive troubleshooting guide for common electrical and mechanical issues in electric-powered food trucks, providing step- by-step procedures for diagnosing and resolvingproblems.
Task 2: Designing a diagnostic tool or system that can quickly identify and report potential issues in the electrical system, battery, or other components of electric-powered food trucks.
Task 3: Conducting a thorough inspection and maintenance program for electric- powered food trucks, including regular checks of critical components, wiring integrity, and electrical connections.
Task 4: Analyzing real-world case studies of electrical failures or malfunctions
in electric-powered food trucks, identifying root causes and proposing preventive measures or design improvements.
Task 5: Developing a training program for food truck operators and technicians, focusing on troubleshooting techniques, preventive maintenance, and safe handling of electrical components in electric-powered food trucks.

SB8063 APPLIED ML FOR HEALTH CARE INDUSTRY L T P C

1 0 2 2

Unit No.	Name	Theory Hours	Practical Hours	Total Hours
1	Understanding CRISP-DM Framework. And see how real time projects can be done.	4	2	6
2	Understanding the Health Care Domain	3	6	9
3	Understanding the data given and prepare it according to the problem Statement.	3	10	13
4	Apply the relevant models and Evaluate the model with test data from historic data. Also try to evaluate with new data. (Real time testing)	3	6	9
5	Do deployment	2	6	8

Total Hours	**15 ***	**30**	**45**

Pre-Requisites:

- Linear Algebra
- Programming in Python

Requirements

Hardware:

- 4GB RAM
- i3/i5 processor

Software:

- Programming Language –Python 3.0 (Open Source)

Skill Objectives:

- Develop models of data from the Health Care Industry.
- Analyze results to identify potential areas for improvement in the prediction of diseases. Ranging from anything like a common cold to cancer.

CONTENT

UNIT I Understanding CRISP-DM Framework. And see how real time projects can be done. 06 Hours

Theory 04 Hours

Concepts, Instances, Attributes, what is a Concept, what is an Example, What is in an Attribute, Preparing the Input, CRISP-DM, Best practices for data mining

Skill Training 02 Hours

Using python look at how CRISP-DM Framework is used in real time projects

UNIT II Understanding the Health Care Domain 09 Hours

Theory 03 Hours

Introduction to Discovering drugs, Virtual assistance, Wearables, Tracking Patient Health, Diagnostics, Predictive Analytics in Healthcare, manage chronic diseases, Monitor and analyze the demand for pharmaceutical logistics, predict future patient crisis, Deliver faster hospital data documentation, Medical Image Analysis with case studies.

Skill Training 06 Hours

Using a case study make the student learn how to understand the Domain pertaining to Health Care Industry.

UNIT III Understanding the data given and prepare it according to the problem Statement. 13 Hours

Theory 03 Hours

What is Data Wrangling, Acquiring data, Common data formats, data cleaning, transformation, aggregation, basics of chart and graph design, as well as data visualization best practices

Skill Training **10 Hours**

Understand Data and perceive data, apply transformations, changing from one data type to the other as required, transforming data from character to integer, Aggregation of data and learn about charts and data visualizations in Python.

UNIT IV Apply the relevant models and Evaluate the model with test data from historic data. Also try to evaluate with new data. (Real time testing) **09 Hours**

Theory **03 Hours**

Linear and logistic regression, KNN, SVM, Decision Tree, Random Forest, Naïve Bayes and K Means including model selection, model evaluation, and variable selection. Students will learn to apply the techniques to solve real-world prediction problems

Skill Training **06 Hours**

Learn to apply which algorithm in which scenario and working of each of this algorithm. Evaluation and how the variable can be selected.

UNIT V Deployment **08 Hours**

Theory **02 Hours**

Learn how to put the model on a server and also create a small User Interface to interact with the model. Learn the deployment process.

Skill Training **06 Hours**

Design of the frontend tool to deploy the model.

TOTAL: 45 PERIODS

Skill Outcomes:

1. Learn the steps of the CRISP-DM Framework for standardizing across projects
2. Visualize the transformations of data using statistical techniques.
3. Prepare data by removing the missing values, Outliers and apply required transformations on the data.
4. Apply the relevant models and learn to evaluate the model
5. Create value proposition out of the analysis
6. Design to deploy the project for client consumption.

References

1. Making sense of data I: a practical guide to exploratory data analysis and data mining
2. Making Sense of Data II: A Practical Guide to Data Visualization, Advanced Data Mining Methods
3. CRISP-DM Framework
4. Data Analytics: Practical Guide to Leveraging the Power of Algorithms, Data Science, Data Mining
5. Data Mining: Concepts and Techniques Solution Manual
6. Data Mining Practical Machine Learning Tools and Techniques 3rd Edition, Mantesh

Test Project 1:

Title: Prediction of Parkinson Disease.

Description:

A hospital has got data collected of its patients who had suffered from Parkinson's Disease. Now they want to build a model that predicts using the historical data with that of a new patient. Using the CRISPM-DM framework implement a model to predict whether someone has the disease or not.

Note:

- This is a classification problem and we need to find which classification algorithm / Model gives you the best level of accuracy in the classification.

Test Project 2:

Title: Prediction of Heart Failure

Description:

In this project you have to predict the heart failure in advance. Collection of the parameter at a specified time should give the prediction in advance giving the individual sufficient time to reach hositpal.

Note:

- This is a classification problem and we need to find which classification algorithm / Model gives you the best level of accuracy in the classification in order to say whether a heart failure in going to occur.

Industry Scope:

These are essential tools required in the Health Care Industry. These tools the accuracy if increased can be productized and sold to other for consumption.

Industry Test Cases:

Four use cases in the healthcare industry, along with their respective problem statements, challenges, goals, data sources, and the outcomes the solutions should provide:

Use Case 1: Patient Risk Stratification

Problem Statement: Identify high-risk patients and predict potential health complications to enable proactive interventions and personalized care.

Challenges to be Addressed:

- Identifying patients at risk of developing chronic diseases or experiencing adverse health events.
- Integrating and analyzing diverse patient data from multiple sources.
- Implementing scalable and real-time risk prediction models.

Goals:

- Develop predictive models to stratify patients based on their risk levels.
- Enable proactive interventions to prevent or manage health complications.
- Improve patient outcomes and reduce healthcare costs.

Data Sources:

- Electronic health records (EHR) and medical histories.
- Vital signs monitoring data.
- Genetic and genomic data.

Outcomes:

- Early identification of high-risk patients for targeted interventions.
- Improved patient engagement and adherence to treatment plans.
- Reduced hospital readmissions and healthcare costs.

Use Case 2: Disease Outbreak Detection
Problem Statement: Detect and monitor disease outbreaks in real-time to facilitate early response and preventive measures.
Challenges to be Addressed:

- Timely identification and tracking of infectious diseases.
- Integrating and analyzing data from various sources, including social media and healthcare facilities.
- Implementing automated surveillance systems to enable rapid response.
-

Goals:
- Develop algorithms to detect disease outbreaks based on patterns and anomalies.
- Enable real-time monitoring of disease spread and transmission.
- Facilitate early intervention and public health measures.

Data Sources:
- Disease reporting systems.
- Social media data for monitoring public health discussions.
- Hospital admissions and emergency department records.

TOTAL : 45 PERIODS

OUTCOMES*:*
- Early detection and response to disease outbreaks.
- Improved public health planning and resource allocation.
- Minimized disease transmission and improved population health.

Use Case 3: Predictive Analytics for Patient Readmission
Problem Statement: Predict and reduce hospital readmissions by identifying factors contributing to readmission and implementing preventive strategies.
Challenges to be Addressed:

- Identifying patients at high risk of readmission based on clinical, socioeconomic, and behavioral factors.
- Analyzing large-scale patient data to identify readmission patterns and risk factors.
- Implementing interventions and care transitions to prevent unnecessary readmissions.

Goals:
- Develop predictive models to identify patients at risk of readmission.
- Implement interventions and care plans to reduce readmission rates.
- Improve care coordination and patient outcomes.

Data Sources:
- Electronic health records (EHR) and medical histories.
- Socioeconomic and demographic data.
- Patient surveys and feedback.

Outcomes:
- Reduced hospital readmissions and healthcare costs.
- Enhanced patient care coordination and transition.
- Improved patient satisfaction and outcomes.

Use Case 4: Fraud Detection in Healthcare Insurance
Problem Statement:

Identify and prevent healthcare insurance fraud through advanced analytics and anomaly detection.
Challenges to be Addressed:

- Identifying fraudulent patterns and behaviors in healthcare insurance claims.
- Analyzing large volumes of claims data in real-time.

- Implementing automated fraud detection systems to minimize false positives.

Goals:

- Develop algorithms to detect fraudulent activities in healthcare insurance claims.
- Reduce fraud-related losses and costs for insurance providers.
- Improve efficiency in claims processing and investigation.
-

Data Sources:

- Healthcare insurance claims data.
- Provider and patient data.
- External data sources for cross-referencing (e.g., public records, billing codes).

Outcomes:

- Early detection and prevention of healthcare insurance fraud.
- Improved accuracy in claims processing and reduced false positives.
- Cost savings for insurance providers and improved trust in the system.

These use cases in the healthcare industry address various challenges and goals, leveraging diverse data sources to develop solutions that improve patient care, public health, fraud detection, and cost efficiencies in the healthcare system.

SB8064 INDUSTRY 4.0- DIGITAL TWIN FOR PROCESS INDUSTRY L T P C
1 0 2 2

COURSE OBJECTIVE:

1. The main aim is understand the overview of Industry 4.0 & its benefits to the industry to enhance business excellence.
2. The objective of a digital twin is to identify intolerable deviations from optimal conditions along any of the various dimensions
3. The key objective of Industry 4.0 is to drive manufacturing forward: to be faster, more efficient, and customer-centric while pushing beyond automation and optimization to discover new business opportunities and models
4. The use of digital twins enables more effective research and design of products, with an abundance of data created about likely performance outcomes.

SYLLABUS

UNIT 1: INTRODUCTION to COMOS, Details of COMOS: Front End Engineering & Design (FEED) phase, Efficient creation of intelligent Process Flow Diagrams, Process Flow Diagram ,Creating Structures and objects, Editing and placing object, Importing simulation data, Case and process data management, Copy methods, Editor, P&ID , Conversion of PFDs into Piping and Instrumentation Diagrams (P&IDs) without data loss, Object Creation. Fast & easy EI&C engineering with COMOS Automation.

UNIT2: COMOS ELECTRICAL DESIGNS: Easy and fast cabinet layout planning, including automation solutions, Creation of logical diagrams and operation charts for the graphical visualization. Electric Circuit Diagram & Distribution Cabinet Design in the COMOS.,

UNIT3: SCADA: virtual representation of a process plant. Real time data monitoring and control. Alarm, warning, data logging, trends, emergency operations. SCADA can be interfaced with PLC, DCS, SIMIT, COMOS for easier commissioning and installation of a process plant.

UNIT 4: INTRODUCTION to SIMIT. Creating the level process station, Simulation platform for virtual commissioning of Process industry, Comprehensive tests of automation applications like PLC& DCS and Coupling of SIMIT with COMOS.

UNIT 5: INTRODUCTION to DCS (PCS7), SCADA for process control station, Configuration of

CPU module with interface modules, Input and output configuration, CFC programming, Implementation of PID Controller in DCS, coupling between COMOS – PCS7.

TOTAL : 45 PERIODS

COURSE OUTCOME:

Digital Twin for Process Industry using COMOS-SIMIT-DCS (PCS7)-SCADA
Students will be able to learn following knowledge in software to make Industry 4.0 integrations.

- COMOS is plant engineering software from Siemens. The applications for this software are in the process industries for the engineering, operation, and maintenance of process plants as well as their asset management.

- Students can learn the standard symbols for the process plant like control valves, field equipments, controllers,...
- Students can read a P&ID diagram and they can able to recreate it in the COMOS software.
- Students can gain more idea about datasheet of plant equipment.
- In which get more knowledge in PFD & PID of a Process Industry.
- Creating a basic Electric circuit diagram & cabinet Design.
- Coupling of COMOS with SIMIT and PCS7 for a process plant.
- SIMIT is a simulation platform that is capable of comprehensive automation project testing, and process commissioning all in one platform.
- Students get knowledge on virtual simulation and they can perform plant processes before getting into the plant.

- DCS

The students can able to configure any DCS controller for a plant by using controllers Model number. Programming of CFC using Physical and memory addresses. PID controller can be implemented easily. Network configuration and IP addressing of a controller can be done easily.

- SCADA

The students can able to create SCADA screen for process plant without anyone’s help. They can able to monitor the real time data and control the plant using SCADA. Further operations like alarm, warning, emergency operations can be learned.

INDUSTRY SCOPE:

On Completion of this course, students get industry scope in process industry, Power lant to get a digitalisation using COMOS, SIMIT, and DCS. Digital twin of process industry, Design of plant layout, pipe layout, Process Flow Diagram, Piping and Instrumentation Diagram, Electrical circuit diagram and Electrical Cabinet Design, Automation using PLC, DCS Controllers.

INDUSTRY USE CASES

1) **Create a digital twin of single process in a process industry**.

In an industry create a digital twin of one process like (level, flow, pressure, temperature) in that PFD and PID in the same way they are using in that industry to make in simulation of the process to get a similar output.

2) Analysing the various parameter in a process industry.

In an industry, there are various parameters available like level, flow, pressure, temperature in the different process system. They all used to analysis the process done in the correct way. They are using in that industry to change the parameter to the process to get a various output.

3) Creating a plant layout & process layout of an industry.

Design a plant layout and process layout of newly created process industry for knowing the PFD, PID of the plant cost analytic of industry.

4) Simulation of process industry using SIMIT

Virtual commissioning of An process industry using COMOS and SIMIT and find out the various simulation done through the automation like (PLC, DCS, ...) they are controlled by this and simulation each process.

5) Online monitoring of industry using SIMIT and COMOS

The student get more knowledge in online monitoring of all the field equipment using the MQTT in the COMOS and update in the IOT all this process are useful for industry.

6) Implementation of new idea in simulation and run through it.

The student can Implementation of new idea and create different type of process in the industry using COMOS and SIMIT and get more information in the simulation itself and then go to real prototype for the industry setup

7) Creation of PFD & PID of process plant

They get the more knowledge in the creation of Process Flow Diagram and Piping and Instrumentation Diagram of an process station to know the valve of using the COMOS are understand.

8) Creating an Electric Circuit Diagram of an industry

Design a circuit diagram of an whole industry with the help of the COMOS to get more information like single line diagram, connection diagram, in the system to update the intelligent data in the process industry.

9) Creation of Electric Cabinet Design using COMOS

Design a cabinet design of the various cabinet having some PLC, DCS, Circuit breaker, etc.. are coupled each other in the cabinet are drawn in COMOS by using the different diagram available they are loop diagram, terminal diagram, junction diagram are used to identify the wiring in cabinet to equipment.

10) Creation of CPU and Interface module and IO configuration in PCS7

The student has to note down the hardware model numbers and they have to configure it in PCS7 software. They have to create a project and hardware configuration from the catalogue Configurations of IP address and interface module address to be assigned as per hardware.

- Digital input/outputs
- Analog input/outputs
- Addition of above IO address to symbol table access the addresses in the program

11).CFC programming

Creation of logic gates, Timer, counter in the CFC. Creation of small example programs like Conveyor control, sensor interface, PID controller creation.

12). Network & OS configuration with WINCC explorer

Creation of network configuration which includes PC, CPU module, IM module for SCADA interface. Configuring SCADA screen for the created CFC program.

13). User Interface creation using Graphics designer

Creation of plant design using graphics designer tool. Implementation of button, LED, bar, trends, faceplates, piping and plant equipments from the library. Assigning the tags for all objects placed in the design area.

14). Runtime Activation

Activation of runtime screen for the plant. Control and monitor the plant from the created design. The following actions will be performed: ON/OFF control using buttons, PID faceplate actions, data logging, alarm, warning, emergency actions, trends.

15). Level Control Process using SIMIT

Design of level process control station using SIMIT. Implementation of level transmitter, process tank, control valves, pumps,...

16). Design of logic gates, traffic light control system and simple circuit diagram with circuit breaker

Design of logic gates like AND, OR, NAND... using switch and LED in SIMIT Simulation of traffic light control system using SIMIT. Design a process control station using circuit breaker in SIMIT. Implementation of simocode ,over load relay, VFD, ...

17).Coupling of PLC and SIMIT

Interfacing PLC with SIMIT for transferring real time data. Development of process control station in SIMIT and control via PLC.

18). Level Process using Raspberry Pi and interface with COMOS

Run a Level Process using Raspberry and interface with COMOS for real time data monitoring through MQTT. The plant equipments (level transmitter, control valves) will be connected to raspberry Pi IO’s. The process plant will be controlled and monitored by raspberry Pi. The real time data will be shared to COMOS through MQTT.

19). Flow Process using RP2040 and interface with COMOS

Run a flow Process using RP2040 and interface with COMOS for real time data monitoring through MQTT. The plant equipments (flow transmitter, control valves) will be connected to RP2040 IO’s. The process plant will be controlled and monitored by RP2040. The real time data will be shared to COMOS through MQTT.

20).Temperature Process using Siemens IoT Gateway IOT2050 and interface with COMOS

Run a temperature Process using IoT Gateway IOT2050 and interface with COMOS for real time data monitoring through MQTT. The plant equipments (temperature transmitter, heater) will be connected to IoT Gateway IOT2050 IO’s. The process plant will be controlled and monitored by IoT Gateway IOT2050. The real time data will be shared to COMOS through MQTT.

SB8065 DRONE BASICS, DESIGN, ASSEMBLY, TEST **L T P C**
1 0 2 2

Unit No.	Name	Theory Hours	Practical Hours	Total Hours
1	Drone Basics and Aerodynamics	4	1	5
2	Types of Drones, Parts, Working principle	3	6	9
3	Drone assembly, Test, Calibration	4	9	13
4	Drone simulation, Component selection, Software simulation with various different	2	13	8

5	Flight test, Trimming and Trouble shoot, Type of application, career guidance, opportunities, How to start your drone startup	2	6	15
Total Hours		**15 ***	**35**	**50**

*** Pre-recorded videos**

Pre-Requisites:

- Laptop / PC
- Drone Components
- Drone Transmitter
- Open Area for flying activities

Requirements

Software :

- Open source Drone simulation and testing software related to Drones

Hardware :

- Computer (8GB RAM, i3/i5 Processor)
- Soldering Station
- Soldering Accessories
- Safety Equipment's
- Electrical/ Electronic Component Consumables
- Drone kit (Fly sky 6-8 channel transmitter, Receiver, Flight computer (Ardupilot / KK Board / Pix hawk), Power Distribution Board, LiPo Battery 2200 MAH, BLDC motors, Drone frame, ESC 4 numbers
- Drone assembly tools – Allen key sets

Skill Objectives:

- Learn the basics of flight control, Drone aerodynamics
- Drone parts and its working principle
- Drone assembly, calibration and flying

CONTENT

UNIT I BASICS & AERODYNAMICS 05 Hours

Theory 04 Hours

Aerodynamics, Bernoulli's effect, Aerofoil theory 3 Axis control

Skill Training 01 Hours

Sotwares : NASA Foilsim – Hand on paper aero design - 3 Axis control system

UNIT II DRONES and PARTS 09 Hours

Theory 03 Hours

Drone - Overview, History, Challenges, Market - Types of Drones, Parts of drone, working principle

Skill Training **06 Hours**

Hands on identification of Drone parts and How they work, working etc.,

UNIT III DRONE ASSEMBLY, TEST, CALIBRATION **13 Hours**

Theory **04 Hours**

Drone assembly layout, connections, how to test, how to calibrate

Skill Training **09 Hours**

Hands on assembly, test, calibration using software Mission Planner software
(Pix hawk & APM) for DJI NAZA (NAZA M lite)

UNIT IV DRONE SIMULATION, COMPONENT SELECTION, SOFTWARE SIMULATION WITH VARIOUS DIFFERENT SET OF COMPONENTS

08 Hours

Theory **02 Hours**

How to use Phoenix RC Simulator, E-Calc multi rotor for component selection and simulation

Skill Training **06 Hours**

Hands on Simulator training, E-Calc usage and test with various components

UNIT V FLIGHT TEST, TRIMMING AND TROUBLE SHOOT, CAREER IN DRONE SECTOR

15 Hours

Theory (Pre-recorded videos) **02 Hours**

Testing of Assembled drone, trimming it

Skill Training **13 Hours**

Hands on testing, trimming, flying etc.,
Type of application, career guidance, opportunities, How to start your drone startup

Skill Outcomes:

1. Learning basics of flight, Aerodynamics and theories behind it
2. Learning the parts of drone, working principle and trouble shooting
3. Component selection and simulation with selected components
4. Assembling the drone parts and testing it after calibration of assembled drone
5. Flight test and trimming.
6. Type of application, career guidance, opportunities, How to start your drone startup

TOTAL: 45 PERIODS

References

1. Handbook of Unmanned aerial vehicles, Valavanis, K. (Kimon), editor.; Vachtsevanos, George J., editor.2014
2. Introduction to multicopter design and Control, Quan Quan
3. https://ardupilot.org/planner/docs/mission-planner-overview.html
4. https://dl.djicdn.com/downloads/naza-m%20lite/en/NAZA-M%20LITE_User_Manual_v2.00_en.pdf
5. https://youtu.be/OWaXIK9sHeE

6. https://youtu.be/b8jToUL2AY8
7. https://youtu.be/mLGOzD0wsjE
8. https://youtu.be/tq5tplhjp_Y

Industry use cases:

IIT Madras research park based companies

1. Detect technology https://detecttechnologies.com/

Use of drones in pipeline patrol in chemical industries

2. E-planes https://www.eplane.ai/

Use of Drones as intercity transport

3. Tunga Aerospace https://tungasystems.com/

Use of drones for reconnaissance, defence, military, surveillance

4. Garuda aerospace https://www.garudaaerospace.com/

Use of drones in Agriculture, mapping, cleaning, disaster management etc.,

Tasks:

1.Students have to Come up 3-4 unique applications on drone design technology Example hands on tasks taught to students

1. Settings up a weather monitoring station on drone
2. Setting up a spy camera on drone
3. Setting up a laser lighting system/ lora / Lidar based communication system on the drone system on the drone

SB8066 ARCHITECTURE VIZUALISATION USING AR/VR

L	T	P	C
1	0	2	2

COURSE OBJECTIVE:

- Create a 3D Building Exterior Elevation.
- Create Interior design of building.
- Presentation using Virtual reality.

COURSE CONTENT:

UNIT I Creating a 3D scan, stitching & building a 3D model 3 + 6

What is photogrammetry - What is LIDAR - LIDAR apps and Technique, Scan LIDAR Meshes - Convert to CAD - Import CAD scan - Stitching images - Creating walls, Floors & Roofs, Doors & Windows

UNIT II Building an Interior Design using Blender 3 + 6

Importing 3D - Cleaning up 3D scans - Creating Interiors - College lobby, Library, Study room, Washroom, Conference Hall, Institution Computer lab

UNIT III Creating an Exterior Design for the Building 3 + 6

Creating elevation – Stairs, Window, Sunlight - Textures details – Considerations for sunlight, Room & object textures

UNIT IV Scene Building for VR 3 + 6

Scene building - Preparing for VR - Importing VR Models - Adding Skybox

UNIT V VR Presentation 3 + 6

Creating VR tools - Creating interactive menus, options, buttons, UI - Creating Walkthrough - Virtual Reality Presentation.

TOTAL : 45 PERIODS

COURSE OUTCOME:

1. Create a 3D Exterior Elevation and Interior model for the given building plan.
2. Render the 3D model of the given building and create a realistic views and Videos.
3. Design for performance of building
4. Present the Model using Virtual reality.

FOR FURTHER READING:

Virtual Reality – Walkthrough using VR – Parametric architecture

REFERENCES:

1. SketchUp for Site Design: A Guide to Modelling Site Plans, Terrain, and Architecture 2nd Edition by Daniel Tal.
2. SketchUp for Builders: A Comprehensive Guide for Creating 3D Building Models Using SketchUp by John Brock.
3. SketchUp for Interior Design: 3D Visualizing, Designing, and Space Planning.
4. The SketchUp Workflow for Architecture: Modelling Buildings, Visualizing Design, and Creating Construction Documents with SketchUp Pro and Layout.

ONLINE REFERNCES:

- Architectural Visualization with V-Ray Next for SketchUp, a course by María Alarcón
- Photorealism for Interior Spaces with Lumion, a course by Angular Lab
- https://help.sketchup.com/en/sketchup/getting-started-self-paced-tutorials
- https://www.udemy.com/share/106hka3@K3X-wvSJ_iKej00tbRmgtLOxImvNOPkJ5ilN1Pa4hIV7hhFuhS62zd2r1JRZ2AI/
- http://www.sketchup.com/products/sefaira

SOFTWARE REQUIREMENTS

1. Sketchup
2. Twin motion
3. Unity or Unreal
4. Blender
5. Photoshop

HARDWARE REQUIREMENTS

1. PC – Minimum requirements
   a. 16 – GB RAM
   b. 2GB - Dedicated GPU
   c. 500 GB HDD
2. Oculus Quest 2 (VR Presentation)
3. iPad PRO 2022 with LiDAR Camera

20 USE CASES - FINAL PROJECT SHOWCASE (EACH INDIVIDUAL STUDENT):

1. Pre-built/Existing Architecture Space: Each team of students would 3D scan the department of their choice from their college using photogrammetry & LiDAR, and build 3D modelling using Blender, Lighting & Texturing and develop the content for VR, adding VR Walkthrough elements with Interactive UI & Menu. (More than 20 + use cases)

2. To-be built/New Architecture Space: Each team of student would choose from a set of 15 spaces from the categories of office, residential, factory from the scratch (2D drawing) and build 3D modelling using Blender, Lighting & Texturing and develop the content for VR, adding VR Walkthrough elements with Interactive UI & Menu. (More than 20 use cases)