Zeal Education Society's

ZEAL COLLEGE OF ENGINEERING & RESEARCH, PUNE–41

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

*NAAC Accredited with **A+** Grade / ISO 21001:2018*

DEPARTMENT OF COMPUTER ENGINEERING

Curriculum Structure and Syllabus of

T.Y. B. Tech. – Computer Engineering

(With effect from - Academic Year 2026 - 27)
(Pattern-2024)

VISION OF THE INSTITUTE

To be a premier institute in technical education by imparting academic excellence, research, social and entrepreneurial attitude.

MISSION OF THE INSTITUTE

- To achieve academic excellence through innovative teaching and learning process.
- To imbibe the research culture for addressing industry and societal needs.
- To inculcate social attitude through community engagement initiatives.
- To provide conducive environment for building the entrepreneurial skills.

DEPARTMENT OF COMPUTER ENGINEERING

VISION:

To emerge as a department of repute in Computer Engineering through innovative teaching, research, social responsibility, and entrepreneurial skills, developing responsible IT professionals.

MISSION:

M1: To provide in-depth technical education and hands-on experiences in Computer engineering using modern tools and technologies

M2: To end eavor innovative research culture to fulfill the needs of Industry and Society.

M3: To instill in students a deep sense of social responsibility.

M4: To strengthen collaboration between industry and academia, fostering the development of entrepreneurial skills among the students.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs):

PEO1: Graduates will apply knowledge of computer engineering to solve complex engineering problems, propose algorithmic solutions, thus establishing themselves as successful IT professional.

PEO2: Graduates will exhibit leadership qualities and innovative thinking, contributing to the development of cutting-edge solutions and career advancements in the field of computer engineering through research, collaborate team work and entrepreneurial initiatives.

PEO3: Graduates will maintain ethics, meet societal duties and pursue life-long learning to stay updated and contribute meaningfully to their field and the society.

PROGRAM OUTCOMES (POs):

PO1: Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4).

PO3: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)

PO4: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).

PO5: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)

PO6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment.

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)

PO8: Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

PO9: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences.

PO10: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.

PO11: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8)

PROGRAM SPECIFIC OUTCOMES (PSOs):

PSO1: Identify, formulate and analyze real-life mechanical engineering problems by applying the principles of thermal, design, manufacturing, interdisciplinary and allied engineering.

PSO2: Select and apply appropriate materials, metallurgical processes, measurement techniques, feedback control systems, hydraulic and pneumatic control systems to develop appropriate solutions to mechanical engineering problems.

PSO3: Select and apply appropriate manufacturing technologies and tools, and develop competencies for working in manufacturing and allied industries.

PSO4: Apply acquired knowledge, skills, and hands-on experiences to work professionally in mechanical and related systems.

DEPARTMENT OF COMPUTER ENGINEERING

LIST OF ABBREVIATIONS

Abbreviation	Description
BSC	Basic Science Course
ESC	Engineering Science Course
PCC	Program Core Course
PEC	Program Elective Course
MDM	Multidisciplinary Minor
OE	Open Elective - Other than a particular program
VSEC	Vocational and Skill Enhancement Course
AEC	Ability Enhancement Course
ENTR	Entrepreneurship
EC	Economics
MC	Management Courses
IKS	Indian Knowledge System
VEC	Value Education Courses
RM	Research Methodology
CEP	Community Engagement Project
FP	Field Project
PROJ	Project
INT	Internship
OJT	On Job Training
CC	Co-curricular Courses
HSSM	Humanities Social Science and Management
ELC	Experiential Learning Course
B. Tech	Bachelor of Technology
L	Lecture
P	Practical
T	Tutorial
H	Hours
CR	Credits
CIE	Continuous Internal Evaluation
ETE	End Term Evaluation
TW	Term Work
OR	Oral
PR	Project

DEPARTMENT OF COMPUTER ENGINEERING

Third Year B.Tech.– Computer Engineering: Semester-V

Course Code	Course Type	Course Name	Teaching Scheme(hrs/Week)							Evaluation Scheme					
			L	P	T	H	CR			CIE	ETE	TW	PR	OR	Total
							TH	PR/Tut	Total						
COPC508	PCC	Automata Theory	3	-	-	3	3	-	3	40	60	-	-	-	100
COPC509	PCC	Operating Systems	3	-	-	3	3	-	3	40	60	-	-	-	100
COPC509P	PCC	Operating Systems Lab	-	2	-	2	-	1	2	-	-	-	50	-	50
COPC510	PCC	Machine Learning	3	-	-	3	3	-	3	40	60	-	-	-	100
COPC510P	PCC	Machine Learning Lab	-	2	-	2	-	1	2	-	-	-	50	-	50
COPE501	PEC	Program Elective Course - I *	3	-	-	-	3	-	3	40	60	-	-	-	100
COPE501P	PEC	Program Elective Course - I * Lab	-	2	-	-	2	1	2	-	-		-	25	25
ALOE503	OE	Open Elective-III#	2	-	-	2	2	-	2	-	-	50	-	50	100
COMD503	MDM	Fundamentals of IoT	2	2	-	4	2	1	3	-	-	25	25	-	50
COIN504	ELC-INT	Internship- IV	4Weeks					2	2	-	-	25	-	-	25
Total			**16**	**8**	**-**	**24**	**16**	**6**	**22**	**160**	**240**	**100**	**125**	**75**	**700**

***-Select any one course from the given Program Elective Courses**		
Course Code	**Course Type**	**Program Elective-I**
COPE501A	PEC	Introduction to Data Science
COPE501AP		Introduction to Data Science Lab
COPE501B		Computer Networks
COPE501BP		Computer Networks Lab
COPE501C		Augmented and Virtual Reality
COPE501CP		Augmented and Virtual Reality Lab

#-Select any one course from the given Open Elective Courses		
Course Code	**Course Type**	**Open Elective-III**
ALOE503A	OEC	Introduction to Data Analytics
ALOE503B		Green Building
ALOE503C		Electric Vehicles and Charging Infrastructure
ALOE503D		Introduction to Stock Market
ALOE503E		Patent Law for engineers and Scientist

BoS Chairman

ZES's Zeal College of Engineering & Research Narhe, Pune-41 PU/PN/Engg./285/2007

Director
ZES's Zeal College of Engineering & Research
Narhe, Pune - 411041.

DEPARTMENT OF COMPUTER ENGINEERING

Third Year B.Tech.– Computer Engineering: Semester-VI

Course Code	Course Type	Course Name	Teaching Scheme (hrs/Week)							Evaluation Scheme					
			L	P	T	H	CR			CIE	ETE	TW	PR	O R	Total
							TH	PR/Tut	Total						
COPC611	PCC	Design and Analysis of Algorithms	3	-	-	3	3	-	3	40	60	-	-	-	100
COPC611P	PCC	Design and Analysis of Algorithms Lab	-	2	-	2	-	1	2	-	-	-	-	25	25
COPC612	PCC	Deep Learning	3	-	-	3	3	-	3	40	60			-	100
COPC612P	PCC	Deep Learning Lab	-	2	-	2	-	1	2	-	-	-	25	-	25
COPE602	PEC	Program Elective Course - II *	3	-	-	3	3	-	3	40	60	-		-	100
COPE602P	PEC	Program Elective Course - II * Lab	-	2	-	2	-	1	2	-	-	-	25	-	25
COPE603	PEC	Program Elective Course - III *	3	2	-	5	3	1	4	40	60	-	-	25	125
COMD604	MDM	Wireless Communication	2	-	-	2	2	-	2	40	60	-	-	-	100
COVS605	VSEC	Full-Stack Flask Lab	-	4	-	4	-	2	2	-	-	-	50	-	50
COIN504	ELC - INT	Internship - IV	4 Weeks					2	2	-	-	25		-	25
Total			**14**	**12**	**-**	**26**	**14**	**8**	**22**	**200**	**300**	**50**	**100**	**50**	**700**

*** - Select any one course from the given Program Elective Courses**		
Course Code	**Course Type**	**Program Elective-II**
COPE602A	PEC	Cloud Computing
COPE602AP		Cloud Computing Lab
COPE602B		MERN Stack Development
COPE602BP		MERN Stack Development Lab
COPE602C		Agile and DevOps
COPE602CP		Agile and DevOps Lab

*** - Select any one course from the given Program Elective Courses**		
Course Code	**Course Type**	**Program Elective-III**
COPE603A	PEC	Embedded Systems and RTOS
COPE603B		Ethical Hacking & Penetration Testing
COPE603C		5G & Next Generation Networks

BoS Chairman

ZES's Zeal College of Engineering & Research PU/PN/Engg./285/2007 Narhe, Pune-41

Director

ZES's Zeal College of Engineering & Research Narhe, Pune - 411041.

DEPARTMENT OF COMPUTER ENGINEERING

INDEX

Sr. No.	Course Code	Course Name	Page No.
Third Year B. Tech.: Semester – V			
1.	COPC508	Automata Theory	10-12
2.	COPC509	Operating Systems	13-16
3.	COPC510	Machine Learning	17-21
4.	COPE501	Program Elective Course - I *	23-32
5.	ALOE503	Open Elective-III#	33-49
6.	COMD503	Fundamentals of IoT	51-53
7.	COIN504	Internship- IV	-
Third Year B. Tech.: Semester – VI			
1.	COPC611	Design and Analysis of Algorithms	55-57
2.	COPC612	Deep Learning	58-62
3.	COPE602	Program Elective Course - II *	64-74
4.	COPE603	Program Elective Course - III *	75-84
5.	COMD604	Wireless Communication	85-86
6.	COVS605	Full-Stack Flask Lab	87-89
7.	COIN504	Internship – IV	-

SYLLABUS

SEMESTER - V

DEPARTMENT OF COMPUTER ENGINEERING

Program: Computer Engineering								**Semester: V**	
Course: Automata Theory								**Code:** COPC508	
Teaching Scheme (Hrs/week)				**Evaluation Scheme(Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	-	-	03	40	60	-	-	-	100

Prerequisites:

1. Discrete Mathematics (Sets, Relations, Functions, Logic),
2. Basic Concepts of Languages and Grammars

Course Objectives:

1. Understand formal language concepts, Chomsky hierarchy, and state abstraction for AI agents.
2. Design DFAs and NFAs for pattern recognition, game AI, and network security applications.
3. Apply regular expressions, Arden's theorem, and minimization for text searching and input validation.
4. Analyze CFGs, normal forms, and PDAs for parsing and structured data validation.
5. Evaluate Turing machines to model CPU execution, system architecture, and computability limits.
6. Assess undecidability, complexity classes, and NP-hardness in AI optimization and SAT solvers.

Course Outcomes: After completion of this course, students will be able to-

CO1	Explain the fundamental concepts of formal languages, automata, and the Chomsky hierarchy.
CO2	Design and analyze Deterministic Finite Automata (DFA), Non-Deterministic Finite Automata (NFA), and their applications in pattern recognition.
CO3	Apply regular expressions and regular language properties to model and solve text processing and language recognition problems.
CO4	Construct Context-Free Grammars (CFGs) and Pushdown Automata (PDAs) for syntax analysis and language processing.
CO5	Analyze Turing machines to model CPU execution and fundamental limits of computation.
CO6	Evaluate computational complexity (P, NP, NP-Complete) and NP-hardness in AI constraint solvers.

Course Contents:

Unit	**Description**	**Unit**
1.	**Fundamentals of Automata and Languages:** Introduction to Automata Theory, Alphabets, Strings, Languages, Operations on Languages, Chomsky Hierarchy Overview, Significance and practical relevance in lexical analysis and pattern matching, Analytical approaches using set theory and state-transition modeling**,** Finite-state abstraction in AI agents, event-driven reactive architectures, and automated agent behavior modeling.	07

	Case Study: Traffic light controller state modeling.	
2.	**Finite Automata** Deterministic Finite Automata (DFA), Non-Deterministic Finite Automata (NFA), NFA to DFA Conversion, Design of Finite Automata, Significance and practical relevance in text searching and digital logic, Analytical approaches using subset construction and state trace analysis using JFLAP,Finite State Machines (FSMs) for game AI, robot motion state control, and network firewall deep packet inspection**.** Case Study: Firewall packet filtering engine design	07
3.	**Regular Languages and Regular Expressions** Introduction to Regular Expressions, Conversions-RE to NFA,DFA,DFA to RE Conversion using ardens theorem , DFA Minimization, Pumping Lemma, Significance in web input validation and string searching, Analytical approaches using Hopcroft's minimization algorithm. Case study: RE in text search and replace.	07
4.	Context-Free Grammars and Pushdown Automata Context-Free Grammars (CFG), Parse Trees, Context free Language (CFL),Ambiguity in Grammars, Chomsky Normal Form, Greibach Normal Form, Applications of CFG,Pushdown Automata (PDA), PDA vs CFL, Significance in compiler parsing and structural validation. Case Study: XML/JSON structure validation using PDA.	07
5.	Turing Machines and Computability Turing Machine definition and design, Universal Turing Machine, Turning machine vs Finite automata, Significance in modeling computer architecture and CPU execution, Analytical approaches using tape-transition tracing and machine encoding. Case Study: Modeling basic CPU ALU execution.	07
6.	Decidability and Complexity Decidable vs. Undecidable Problems, Halting Problem, Complexity Classes: P, NP, and NP-Complete, Significance in identifying computationally impossible or intractable software problems, Computational intractability in optimization and scheduling algorithms, and the role of SAT solvers and NP-hardness bounds in constraint-based AI systems Case Study: Analyzing intractability in logistics route optimization (TSP).	07
	TOTAL	**42**

Text Books:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, 3rd Edition, Pearson Education.
2. Peter Linz, an Introduction to Formal Languages and Automata, 6th Edition, Jones & Bartlett Learning.
3. Michael Sipser, Introduction to the Theory of Computation, 3rd Edition, Cengage Learning.

Reference Books:

1. Harry R. Lewis and Christos H. Papadimitriou, Elements of the Theory of Computation, 2nd Edition, Pearson.
2. K.L.P. Mishra and N. Chandrasekaran, Theory of Computer Science: Automata, Languages and Computation, 3rd Edition, PHI Learning.
3. John C. Martin, Introduction to Languages and the Theory of Computation, 4th Edition, McGraw-Hill Education.
4. John Martin, Formal Languages and Automata Theory, McGraw-Hill.
5. Dexter C. Kozen, Automata and Computability, Springer

E-Resources:

1. Introduction to Automata Theory, Languages, and Computation (Companion Resources) Pearson Higher Education
2. Introduction to the Theory of Computation – Michael Sipser
   Cengage Learning
3. NPTEL – Theory of Computation (Video Lectures and Notes)
   NPTEL Official Portal
4. MIT OpenCourseWare – Automata, Computability and Complexity
   MIT OpenCourseWare
5. Stanford Online Resources on Automata and Computation
   Stanford Online
6. GeeksforGeeks – Theory of Computation Tutorials
   GeeksforGeeks
7. JFLAP Software Tool and Interactive Simulator (www.jflap.org)

Back to Structure

DEPARTMENT OF COMPUTER ENGINEERING

Program: Computer Engineering								**Semester: V**	
Course: Operating System								**Code:** COPC509	
Teaching Scheme (Hrs/week)				**Evaluation Scheme(Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	02	-	04	40	60	-	-	50	150

Prerequisites:

Programming Fundamentals, Computer Organization and Architecture, Discrete Mathematics, Software Engineering Basics

Course Objectives:

1. Understand the fundamentals of embedded systems and real-time operating systems (RTOS)with emphasis on Nucleus OS.
2. Explain the architecture, kernel services, task management, and scheduling mechanisms of Nucleus RTOS.
3. Apply memory management and inter-task communication techniques using semaphores, mutexes, and message queues.
4. Analyze file systems, device drivers, and networking services provided by Nucleus OS.
5. Develop embedded applications using Nucleus RTOS APIs and debugging tools.
6. Evaluate advanced featuress uchas multiprocessing, security, and IoT integration in embedded systems.

Course Outcomes: After completion of this course, students will be able to-

CO1	Explain OS architecture and kernel functionalities.
CO2	Implement process scheduling and synchronization techniques.
CO3	Analyze memory management mechanisms.
CO4	Manage file systems and storage resources.
CO5	Apply OS security and device management concepts.
CO6	Evaluate virtualization, containerization and modern OS technologies.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Unit I: Introduction to Operating Systems and Kernel Architecture** Evolution and Functions of Operating Systems, Types of Operating Systems: Batch, Multi programming, Multitasking, Real-Time, Distributed ,Operating System Structure and Services ,Monolithic, Layered, Microkernel, Hybrid Architectures ,Kernel Components and System Calls ,Booting Process and Kernel Initialization	7
2.	**Unit II: Process and Thread Management:** Processes and Process States , Process Control Block (PCB) ,Process Scheduling Algorithms ,Threads and Multithreading Models ,Inter-Process Communication (IPC)SynchronizationTechniques:Mutexes,Semaphores,Monitors,Deadlocks:Prevention, Avoidance, Detection, Recovery	7

3.	**Unit III: Memory Management:** Memory Hierarchy ,Contiguous and Non-Contiguous Memory AllocationPaging and Segmentation ,Virtual Memory Management ,Demand Paging and Page Replacement Algorithms ,Thrashing and Working Set Model ,Memory Management in Linux **Case Study: tools for memory management using AI**	7
4.	**Unit IV: File Systems and Storage Management:** File Concepts and Access Methods , Directory Structures ,File Allocation Methods, Integration of AI in file organization. File System Implementation ,Journaling File Systems ,Disk Scheduling Algorithms RAID Concepts and Storage Virtualization	7
5.	**Unit V: Device Management and System Security:** I/O System Organization ,Interrupts and DMA ,Device Drivers and Driver Architecture,Disk and Peripheral Management ,Operating System Security Principles,Authentication, Authorization, Access Control ,Malware Protection and Secure OS Design, AI driven innovations in Modern Device management.	7
6.	**Unit VI: Advanced Operating System Technologies:** Virtualization and Hypervisors ,Containers and Container Orchestration ,Cloud Operating Systems ,Mobile Operating Systems ,Real-Time Operating Systems (RTOS) , Kernel Tuning and Performance Optimization ,Emerging Trends: AI-Assisted Operating Systems, Edge and Fog Computing **AI Tools for operating system-Warmwind OS, AIOS, Claude Code**	7
	TOTAL	**42**
List of Practical's:(Perform any 8)		
1.	**First Fit, Best Fit and Worst Fit Memory Allocation** **Problem Statement:** Develop a C program to simulate dynamic memory allocation using First Fit, Best Fit and Worst Fit. Accept memory block sizes and process sizes, allocate each process using the selected strategy, and display allocation and remaining memory.	
2.	**Page Replacement – FIFO** **Problem Statement:** Develop a C program to simulate FIFO page replacement. Accept a page reference string and number of frames, display page hits/faults and calculate total faults and hit/fault ratios.	
3	**File Allocation – Sequential, Linked and Indexed** **Problem Statement** Develop a C program to simulate Sequential, Linked and Indexed file allocation. Accept starting block and number of blocks and display the allocation pattern.	

4.	**Disk Scheduling – FCFS** **Problem Statement** Develop a C program to implement FCFS disk scheduling. Accept disk requests and initial head position, display service order and calculate total head movement.
5	**Disk and Peripheral Management in Linux** **Problem Statement** Study disk and peripheral management using Linux utilities and a C program to display file-system capacity. Identify storage devices, partitions, mounted file systems, USB devices and device files.
6	**Access Control using UNIX/Linux Permissions** **Problem Statement** Create a file, assign different permissions to owner/group/others, verify them with ls -l and a C program using stat().
7	**Linux System Performance Monitoring and Analysis** **Problem Statement** Monitor CPU, memory, process, load and disk information on Linux. Observe the system under normal and increased workload and record changes.
8	**Simple RTOS Task using FreeRTOS** **Problem Statement** Develop a simple real-time application on ESP32 using FreeRTOS. Create concurrent tasks for LED blinking and serial status messages and demonstrate task scheduling, priority and delay.
9.	**Build and Run a Simple Docker Container** **Problem Statement** Create a simple containerized web application using an Nginx base image. Build the Docker image, run the container, verify it and manage its lifecycle.

Text Books:

1. Silberschatz,A.,Galvin,P.B.,&Gagne,G., *OperatingSystemConcepts*,10thEdition,Wiley India Pvt. Ltd., 2018.
2. Tanenbaum, A.S.,&Bos,H.,*ModernOperatingSystems*,4thEdition,PearsonEducation, 2015.
3. Stallings,W.,*OperatingSystems:InternalsandDesignPrinciples*,9thEdition,Pearson Education, 2018.

Reference Books:

1. Love,R.,*LinuxKernelDevelopment*,3rdEdition,Addison-WesleyProfessional,2010.
2. Bovet,D.P., &Cesati, M.,*UnderstandingtheLinuxKernel*, 3rdEdition, O'ReillyMedia, 2005.
3. Bach,M.J.,*TheDesignoftheUNIX Operating System*,PearsonEducation, 2003.
4. Mauerer,W.,*ProfessionalLinuxKernelArchitecture*,WileyIndiaPvt.Ltd.,2008.
5. Prasad,K. V.K. K.,Embedded and Real-Time Operating Systems, Dream tech Press, 2005.
6. Tanenbaum, A.S.,DistributedOperatingSystems,PearsonEducation,2009.

E-Resources:

1. Linux Kernel Documentation, Available at:
2. LinuxKernelDocumentation
3. The Linux Documentation Project (TLDP), Available at: The Linux Documentation Project (TLDP)
4. NPTEL–Operating Systems Course, Available at: NPTEL Operating Systems Course
5. MIT Open Course Ware–Operating System Engineering, Available at: MIT Open Courseware Operating System Engineering

GNU Operating System Documentation, Available at: GNU Documentation

Back to Structure

DEPARTMENT OF COMPUTER ENGINEERING

Program: B.Tech.(Computer Engineering)									**Semester:** V	
Course: Machine Learning									**Code:COPC510**	
Teaching Scheme				**Evaluation Scheme**						
Lecture	**Practical**	**Tutorial**	**Credit**		**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	02	00	04		40	60	-	-	50	150

Prerequisites:

1. Basic programming concepts using Python or any high-level programming language.
2. Basic knowledge of data structures, algorithms and problem solving.
3. Fundamentals of probability, statistics and linear algebra.
4. Basic understanding of databases and data handling is desirable.
5. Basic familiarity with NumPy, Pandas, Matplotlib and Jupyter Notebook/Google Colab is desirable.

Course Objectives:

1. To explain the fundamental concepts, learning paradigms, problem formulation and end-to-end workflow of Machine Learning.
2. To analyze, clean, preprocess and transform datasets using appropriate data preparation, exploratory analysis and feature engineering techniques.
3. To implement supervised and unsupervised Machine Learning algorithms for representative prediction, classification and pattern-discovery problems.
4. To evaluate, compare and optimize Machine Learning models using suitable performance measures, cross-validation and basic hyperparameter tuning.
5. To develop practical Machine Learning solutions for industry-oriented problems using Python and standard Machine Learning libraries.
6. To introduce model interpretation, deployment awareness, Responsible AI and ethical considerations in practical Machine Learning workflows.

Course Outcomes:

Upon successful completion of this course, students will be able to:

CO1	Explain fundamental Machine Learning concepts, learning paradigms, problem formulation and the standard industry ML workflow.
CO2	Analyze and apply data preprocessing, exploratory analysis and basic feature engineering techniques to prepare reliable ML datasets.
CO3	Implement and evaluate regression models for continuous-value prediction using suitable performance measures.
CO4	Implement, compare and interpret classification models for real-world predictive problems using appropriate evaluation metrics.
CO5	Apply clustering, dimensionality reduction and basic model optimization techniques to discover patterns and improve model selection.
CO6	Develop an end-to-end Machine Learning solution using Python and Scikit-learn, with basic deployment awareness, model interpretation and Responsible AI considerations.

DEPARTMENT OF COMPUTER ENGINEERING

Course Contents:		
Unit	**Description**	**Duration (Hrs.)**
1.	**Machine Learning Foundations and Industry Workflow:** Introduction to Artificial Intelligence (AI), Machine Learning (ML) and Data Science: basic concepts, relationship and differences; applications of ML in real-world systems; basic ML terminology – dataset, feature, label, model and prediction. Types of Machine Learning: supervised and unsupervised learning; brief overview of reinforcement learning. Basic ML workflow: problem definition, data collection, data preprocessing, model training and testing. Advantages and limitations of Machine Learning. **Case Study: Spam Email Detection.** (Implement an AI/ML-based spam email classification system using NLP techniques).	**07**
2.	**Data Preprocessing and Exploratory Data Analysis:** Introduction to structured and unstructured data. Data Preprocessing: handling missing values and basic outlier identification; categorical data encoding using Label Encoding and One-Hot Encoding; feature scaling using Min-Max Normalization and Standardization. Exploratory Data Analysis (EDA): summary statistics, data distribution and basic data visualization. **Case Study:** Customer Churn Prediction (Use AI Tools for Implementation).	**07**
3.	**Supervised Learning – Regression and Model Evaluation:** Regression problem formulation and applications. Simple Linear Regression and Multiple Linear Regression. Polynomial Regression – basic concept and use cases. Introduction to model evaluation using: MAE, MSE, RMSE and R^2; selecting metrics for business context. Generalization, Overfitting and under fitting; bias-variance concept. **Case Study:** House Price Prediction (Implementation Using AI-Assisted Tools such as Chat GPT, Google Gemini, and GitHub Copilot).	**07**
4.	**Supervised Learning: Classification** Introduction to Classification problems and difference from Regression. Logistic Regression: basic concept and sigmoid function. Decision Tree and Random Forest: basic concept and working. K-Nearest Neighbours (KNN): basic concept and selection of K. Introduction to Support Vector Machine (SVM): concept of margin and hyperplane. Model Evaluation: Confusion Matrix, Accuracy, Precision, Recall and F1-Score. Ensemble Learning: Bagging and Boosting, Decision tree and Random Forest algorithm: concept and working (introductory). **Case Study:** Healthcare Classification – Disease Risk Prediction (Use AI Tools for Implementation).	**08**

5.	**Unsupervised Learning and Model Optimization:** Introduction to Unsupervised Learning and its applications. Clustering: K-Means algorithm – concept and basic working. Hierarchical Clustering: Agglomerative approach and dendrograms. Difference between K-Means and Hierarchical Clustering. Dimensionality Reduction: need and applications; Principal Component Analysis (PCA) – basic concept and use cases with minimal mathematics. **Case Study: Customer Segmentation** (AI-Assisted Implementation using KNIME Analytics Platform).	**07**
6.	**Introduction to Neural Networks & Machine Learning Applications:** Introduction to Neural Networks: Biological and artificial neurons, perceptron, ANN architecture, single-layer and multi-layer neural networks, weights and biases. Activation Functions: Sigmoid, ReLU and Softmax. Machine Learning applications: Computer vision, NLP, recommendation systems, healthcare, finance and cybersecurity. **Case Study:** Machine Learning application development pipeline and implementation/demonstration of a simple neural-network-based application**.** (Use AI Tools for Implementation).	**06**
	TOTAL	**42**

Suggested List of Practical's / Laboratory Experiments:
Any 8 assignments and 1 mini project are mandatory.

Suggested Tools / Platforms
Python, Jupyter Notebook / Google Colab, NumPy, Pandas, Matplotlib and Scikit-learn

No.	**Practical / Problem Statement**	**Recommended Dataset**	**Target / Main Variables**	**Kaggle Link**
1.	Data loading, inspection, cleaning and EDA using Pandas	Titanic – Machine Learning from Disaster	Survived; passenger attributes	https://www.kaggle.com/competitions/titanic/data?select=train.csv
2.	Missing-value handling, encoding and feature scaling	Titanic – Machine Learning from Disaster	Age, Embarked, Sex, Pclass, Fare	https://www.kaggle.com/competitions/titanic/data?select=train.csv
3.	Linear Regression; MAE, MSE, RMSE and R^2	House Prices – Advanced Regression Techniques	Sale Price	https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data

4.	Logistic Regression; confusion matrix and performance measures	Titanic – Machine Learning from Disaster	Survived (0/1)	https://www.kaggle.com/competitions/titanic/data?select=train.csv
5.	KNN and Decision Tree classification; compare performance	Titanic – Machine Learning from Disaster	Survived (0/1)	https://www.kaggle.com/competitions/titanic/data?select=train.csv
6.	K-Means clustering and visualization	Mall Customer Segmentation Data	Annual Income, Spending Score	https://www.kaggle.com/datasets/vjchoudhary7/customer-segmentation-tutorial-in-python
7.	PCA for dimensionality reduction and visualization	Wine Quality / Wine dataset (Kaggle alternative recommended below)	Numerical chemical attributes	https://www.kaggle.com/datasets/yasserh/wine-quality-dataset
8.	Random Forest vs basic Decision Tree	Titanic – Machine Learning from Disaster	Survived (0/1)	https://www.kaggle.com/competitions/titanic/data?select=train.csv
9.	Cross-validation and basic hyperparameter tuning	Titanic – Machine Learning from Disaster	Survived (0/1)	https://www.kaggle.com/competitions/titanic/data?select=train.csv
10.	End-to-end mini project	Titanic – Machine Learning from Disaster	Predict passenger survival	https://www.kaggle.com/competitions/titanic/data?select=train.csv

Text Books:

1. Machine Learning with Python, Sebastian Raschka and Vahid Mirjalili, Packt Publishing, 2nd Edition, 2017.
2. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, Aurélien Géron, O'Reilly Media, 3rd Edition, 2022.
3. Introduction to Machine Learning with Python, Andreas C. Müller and Sarah Guido, O'Reilly

Media, 2016.
4. Python Machine Learning, Sebastian Raschka and Vahid Mirjalili, Packt Publishing, 3rd Edition, 2019.

Reference Books:

1. Pattern Recognition and Machine Learning, Christopher M. Bishop, Springer, 2006.
2. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Trevor Hastie, Robert Tibshirani and Jerome Friedman, Springer, 2nd Edition, 2009.
3. Machine Learning, Tom M. Mitchell, McGraw-Hill, 1997.
4. Deep Learning, Ian Goodfellow, Yoshua Bengio and Aaron Courville, MIT Press, 2016.
5. Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig, Pearson, 4th Edition, 2021.
6. Data Mining: Concepts and Techniques, Jiawei Han, Micheline Kamber and Jian Pei, Morgan Kaufmann, 3rd Edition, 2011.

E-Resources:

1. Scikit-learn Documentation and User Guide – https://scikit-learn.org/stable/
2. Google Machine Learning Education – https://developers.google.com/machine-learning
3. Kaggle Learn and Datasets – https://www.kaggle.com/learn
4. UCI Machine Learning Repository – https://archive.ics.uci.edu/
5. NPTEL – Machine Learning / related courses – https://nptel.ac.in/
6. Jupyter Documentation – https://docs.jupyter.org/
7. Google Colab – https://colab.research.google.com/
8. Stanford Online / academic ML resources – https://online.stanford.edu/

Back to Structure

PROGRAM ELECTIVE - I

(Select any one course from the given Program Elective Courses)

DEPARTMENT OF COMPUTER ENGINEERING

Program: B. Tech. (Computer Engineering)									**Semester: V**
Course: Introduction to Data Science									**Code: COPE501A**
Teaching Scheme (Hrs/week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	02	-	04	40	60	-	25	-	125

Prerequisites:

1. Data Structures
2. Python Programming
3. Database Management Systems

Course Objectives:

1. To introduce the fundamental concepts, scope, workflow, and applications of Data Science.
2. To understand data acquisition, data quality assessment, cleaning, and wrangling techniques.
3. To apply descriptive statistics, probability, sampling, and hypothesis testing for data analysis.
4. To perform exploratory data analysis and create effective visualizations using Python libraries.
5. To communicate analytical findings through reports, dashboards, and data storytelling.
6. To understand ethical, privacy, security, societal, and emerging issues in Data Science.

Course Outcomes: After completion of this course, students will be able to -

CO1	Explain Data Science concepts, workflow, data sources, applications, and the roles involved in a Data Science project.
CO2	Acquire, assess, clean, transform, and organize datasets using suitable Python-based data wrangling techniques.
CO3	Apply descriptive statistics, probability, sampling, correlation, and hypothesis testing to analyze datasets.
CO4	Perform exploratory data analysis to identify distributions, trends, relationships, and outliers in real-world data.
CO5	Create appropriate visualizations, dashboards, and analytical reports to communicate data-driven insights effectively.
CO6	Design an end-to-end Data Science solution while considering data ethics, privacy, security, and societal impact.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Fundamentals of Data Science:** Introduction to Data Science; evolution, scope and importance; Data Science, Data Analytics, Business Intelligence and Machine Learning—roles and distinctions; Data Science workflow and problem framing; types of data—structured, semi-structured and unstructured; data sources and data collection considerations; roles and responsibilities in a Data Science team; introduction to Python ecosystem; overview of NumPy, Pandas, Matplotlib and Jupyter Notebook. **Case Study:** Explore a public real-world dataset and identify its data types, attributes, sources, quality concerns, and possible analytical questions.	06

2.	**Data Acquisition, Data Quality and Data Wrangling**: Data acquisition from files, databases, common data formats—CSV, JSON, XML and tabular data; data profiling and quality dimensions—accuracy, completeness, consistency, validity and timeliness; data cleaning concepts; handling missing values and data types; data transformation; filtering, sorting, merging, joining, concatenation, reshaping, aggregation and grouping using Pandas. **Case Study:** Acquire and prepare a student performance dataset for analysis.	07
3.	**Statistical Foundations for Data Science**: Descriptive statistics—mean, median, mode, range, variance, standard deviation, quartiles and percentiles; probability concepts and conditional probability; random variables and commonly used probability distributions; sampling methods; population and sample. **Case Study:** Analyze customer purchase data using descriptive statistics, sampling, confidence intervals and correlation analysis**.**	08
4.	**Exploratory Data Analysis and Data Visualization:** Purpose and process of EDA; univariate, bivariate and multivariate analysis; distribution analysis; outlier identification using statistical and graphical methods; visualization principles and selection of appropriate charts; bar charts, line charts, histograms, boxplots, scatter plots, heat maps and area charts; Matplotlib and Seaborn for analytical visualization. **Case Study:** Perform EDA and communicate key insights from a customer or public-sector dataset.	07
5.	**Data Analytics, Reporting and Dashboards:** Analytical questions and KPI definition; aggregation and comparative analysis; pivot tables and summary views; analytical reporting structure; dashboard design principles; static and interactive dashboards using Python visualization tools. **Case Study:** Design a dashboard and analytical report for a real-world dataset.	07
6.	**Data Science Project Lifecycle and Ethics:** End-to-end Data Science project lifecycle from problem definition and data acquisition to analysis, introduction to cloud-based data science platforms and collaborative notebooks; data privacy, security, governance and responsible data use; bias and fairness in data collection and analysis; ethical use of data and AI; applications of Data Science in healthcare, finance, agriculture, education and smartcities. **Case Study:** Design an end-to-end Data Science solution for a real-world problem using a public dataset, without predictive model development.	07
	TOTAL	**42**

List of Experiment

No	**Practical / Problem Statement**
1	Data Acquisition and Dataset Profiling: Acquire a public dataset from an open-data repository such as Kaggle, UCI Machine Learning Repository, or Data.gov.in. Import CSV/JSON data using Pandas and perform initial profiling by inspecting dimensions, data types, unique values, summary information, missing values, and basic data quality observations.

2	Data Quality Assessment and Cleaning: Given a real-world dataset, identify missing values, duplicate records, inconsistent categorical entries, invalid values, and incorrect data types. Apply suitable cleaning operations and document the changes made to improve data quality.
3	Data Wrangling with Pandas: Perform filtering, sorting, selecting, merging/joining, concatenation, reshaping, grouping, aggregation, and pivot-table operations on multiple related datasets. Produce an analysis-ready table and explain the transformations performed.
4	Descriptive Statistical Analysis Using Python: Compute and interpret measures of central tendency and dispersion, quartiles, percentiles, and frequency summaries for a real-world dataset. Compare groups using appropriate descriptive statistics and communicate the findings.
5	Sampling and Hypothesis Testing: Draw suitable samples from a dataset and demonstrate sampling techniques. Construct confidence intervals and perform a basic hypothesis test using an appropriate statistical test. Interpret the p-value and statistical significance in the context of the problem.
6	Correlation and Relationship Analysis: Calculate covariance and correlation between numerical variables, generate appropriate plots, identify the strength and direction of relationships, and discuss the limitations of interpreting correlation as causation.
7	Exploratory Data Analysis and Outlier Detection: Perform univariate, bivariate and multivariate EDA using Pandas and Python visualization libraries. Identify distributions, trends, relationships, and outliers using boxplots, histograms, scatter plots, and suitable statistical methods.
8	Data Visualization and Storytelling: Create an analytical visualization set using Matplotlib and Seaborn containing suitable bar charts, line charts, histograms, boxplots, scatter plots, and heat maps. Apply visualization principles and prepare a short data story highlighting important insights.
9	Interactive Data Dashboard: Develop an interactive dashboard for a real-world dataset using Plotly and/or Streamlit. Include appropriate KPIs, filters, charts, summary tables, and explanatory text. Organize the dashboard for a defined audience and analytical purpose.
10	Mini Project – End-to-End Data Science Analysis: In a group of 2–3 students, complete an end-to-end Data Science project using a public dataset. The project must include problem definition, data acquisition, data quality assessment, cleaning and wrangling, statistical analysis, EDA, visualization/dashboard,

	interpretation, documentation, ethical considerations, limitations, and recommendations.

Text Books:

1. Wes McKinney, Python for Data Analysis, 3rd Edition, O'Reilly Media.
2. Cathy O'Neil and Rachel Schutt, Doing Data Science, O'Reilly Media.
3. Peter Bruce, Andrew Bruce and Peter Gedeck, Practical Statistics for Data Scientists, 2nd Edition, O'Reilly Media.
4. Foster Provost and Tom Fawcett, Data Science for Business, 2nd Edition, O'Reilly Media.

Reference Books:

1. Cole Nussbaumer Knaflic, Storytelling with Data, Wiley.
2. Kieran Healy, Data Visualization: A Practical Introduction, Princeton University Press.
3. David Spiegelhalter, The Art of Statistics, Basic Books.
4. Hadley Wickham and Garrett Grolemund, R for Data Science, O'Reilly Media.

E-Resources:

1. https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-fall-20111.
https://www.python.org/doc/
2. https://numpy.org/doc/
3. https://pandas.pydata.org/docs/
4. https://matplotlib.org/stable/contents.html
5. https://seaborn.pydata.org/
6. https://scipy.org/
7. https://archive.ics.uci.edu/
8. https://www.data.gov.in/

Back to Structure

DEPARTMENT OF COMPUTER ENGINEERING

Program: B.Tech. Computer Engineering								**Semester: V**	
Course: Computer Networks								**Code: COPE501B**	
Teaching Scheme (Hrs/week)				**Evaluation Scheme(Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	02	-	04	40	60	-	25	-	125

Prerequisites:

Course Objectives:

1. To understand the fundamental concepts, architectures, and components of computer networks.
2. To study the functions and protocol soft he Data Link, Network, Transport, and Application layers.
3. To analyze IP addressing, routing algorithms, congestion control, and network performance.
4. To understand network security mechanisms and secure communication protocols.
5. To explore emerging networking technologies such as cloud networking, Software Defined Networking (SDN)

Course Outcomes: After completion of this course, students will be able to-

CO1	Explain the fundamentals, architecture, and operation of computer networks.
CO2	Analyze the functions and protocols of the Data Link and Network layers.
CO3	Apply Transport and Application layer protocols for reliable network communication.
CO4	Evaluate IP addressing, routing algorithms, and network performance techniques.
CO5	Demonstrate network security mechanisms for secure data communication.
CO6	Illustrate modern networking technologies including cloud networking & SDN

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to Computer Networks:** Introduction to computer networks, network goals and applications, network topology, network types (LAN, MAN, WAN, PAN), reference models (OSI and TCP/IP) ,layered architecture, transmission media, switching techniques, multiplexing techniques, and network devices such as hubs, switches, routers, bridges, gateways, and repeaters.	07
2.	**Data Link Layer:** Data link layer functions, framing techniques, error detection and error correction methods, flow control, sliding window protocols, HDLC, Point-to-Point Protocol (PPP), Medium Access Control (MAC), Ethernet (IEEE 802.3), wireless LAN (IEEE802.11),VLANs, and switching concepts.	07
3.	**NetworkLayer:**IPv4 addressing, subnetting, CIDR, IPv6 addressing, Address Resolution Protocol (ARP), Reverse ARP (RARP), Internet Control Message Protocol (ICMP), routing algorithms, distance vector routing, link state routing, RIP, OSPF, BGP, Network Address Translation (NAT), and Dynamic Host Configuration Protocol(DHCP).	07

4.	**Transport Layer:** Transport layer services, User Datagram Protocol (UDP), Transmission Control Protocol (TCP), TCP segment structure, reliable data transfer, flow control, congestion control, three-way handshake, connection termination, socket programming concepts ,and Quality of Service (QoS).	07
5.	**Application Layer:** Domain Name System (DNS), Hypertext Transfer Protocol (HTTP and HTTPS),File Transfer Protocol (FTP),Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP3), Internet Message Access Protocol (IMAP), Telnet, Secure Shell (SSH),Dynamic Host Configuration Protocol (DHCP), Network Management using SNMP, and multimedia networking concepts.	07
6.	**Network Security:** Network security fundamentals, cryptography basics, firewalls, Virtual Private Networks (VPN), Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS) , Secure Socket Layer (SSL), Transport Layer Security (TLS),IPsec, wireless network security, cloud networking, Software Defined Networking (SDN),	07
	TOTAL	**42**

List of Practical:	
1	Study and configure different network topologies using Cisco Packet Tracer.
2	Configure IPv4 addressing and subnetting for a small network
3	Implement static and dynamic routing using RIP and OSPF.
4	Configure VLANs and Inter-VLAN routing in a switched network.
5	Configure Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) services.
6	Analyze network traffic and protocols using Wireshark.
7	Configure Network Address Translation (NAT) and Port Address Translation (PAT).
8	Configure Access Control Lists (ACLs) to control network traffic.
9	Develop client- server communication using TCP and UDP socket programming.
10	Configure and test FTP, HTTP, and Email (SMTP/POP3) services in a simulated network.
11	Configure a Virtual Private Network (VPN) and basic fire wall rules for secure communication. Design and implement a secure enterprise network using Cisco Packet Tracer, incorporating VLANs, routing, DHCP, ACLs, NAT, and network security features.

Text Books:

1. Computer Networks, 6th Edition, Pearson Education.
2. Computer Networking: A Top-Down Approach, 8th Edition, Pearson.
3. Data Communications and Networking, 5th Edition, McGraw-Hill Education.

Reference Books:

1. Data and Computer Communications, 11th Edition, Pearson.
2. Computer Networks, 5th Edition, Morgan Kaufmann.
3. TCP/IP Illustrated, Volume 1: TheProtocols,2nd Edition, Addison-Wesley.
4. Internet working with TCP/IP, Volume1, 6th Edition, Pearson.

E-Resources:

1. https://skillsforall.com/resources/lab-downloads
2. https://nptel.ac.in/courses/106105183
3. https://www.wireshark.org/docs/
4. https://docs.oracle.com/en/
5. https://www.netacad.com/courses/packet-tracer

Back to Structure

DEPARTMENT OF COMPUTER ENGINEERING

Program: T.Y B.Tech. Computer Engineering						**Semester: V**			
Course: Augmented and Virtual Reality						**Code:COPE603E**			
Teaching Scheme (Hrs/week)				**Evaluation Scheme(Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	02	-	03	40	60	-	25	-	125

Prerequisites:

Basic knowledge of Computer Graphics, Object-Oriented Programming(C#/Java) and Mobile Application Development.

Course Objectives:

1. Understand the fundamentals, concepts, and evolution of Augmented Reality (AR) and Virtual Reality (VR).
2. Learn the hardware, software, and development tools used in AR/VR systems.
3. Design and develop interactive AR and VR applications using modern development platforms.
4. Explore tracking, interaction, visualization, and rendering techniques in immersive environments.
5. Analyze real-world applications, challenges, and future trends in AR/VR technologies.

Course Outcomes: After completion of this course, students will be able to-

CO1	Explain the fundamental concepts, components, and technologies of AR and VR.
CO2	Differentiate between AR, VR, and Mixed Reality (MR) and identify suitable applications.
CO3	Develop simple AR/VR applications using appropriate development tools and SDKs.
CO4	Apply interaction, tracking, and rendering techniques to create immersive environments.
CO5	Analyze AR/VR applications in healthcare, education, gaming, manufacturing, and Industry 4.0.
CO6	Evaluate challenges, ethical issues, and future trends in immersive technologies.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	Introduction to AR and VR Introduction to Augmented Reality(AR), Introduction to Virtual Reality(VR), Difference between AR, VR, and Mixed Reality (MR), History and Evolution of AR/VR, Components of AR and VR Systems, Applications of AR and VR	07
2.	AR/VR Hardware and Software Display Technologies (HMD, Smart Glasses),Sensors and Tracking Devices, Camera as and Motion Controllers, AR/VR Development Platforms, Unity 3D and Unreal Engine Overview, Introduction to AR Core and AR Kit	07
3.	Computer Vision and Tracking Fundamentals of Computer Vision, Marker-Based Tracking, Marker less Tracking,FeatureDetectionandRecognition,MotionTracking,Environment Mapping	07
4.	AR/VR Development: Unity 3D Interface and Project Setup, Scene Creation and Object Manipulation,\Importing 3D Models and Assets, User Interaction Techniques, Gesture and Voice Interaction, Simple AR/VR Application Development.	07

5.	Rendering and Interaction: 3D Graphics Fundamentals, Rendering Pipeline , Lighting and Animation, User Interface Design in AR/VR, Audio Integration, Performance Optimization	07
6.	Applications and Future Trends: AR/VR in Healthcare, AR/VR in Education and Training, Gaming and Entertainment, Industrial and Smart Manufacturing Applications, Ethical Issues and Challenges, Future Trends (Meta verse, Digital Twins, Extended Reality)	07
	TOTAL	**42**

List of Practical:

1.	Installation and setup of Unity 3D and Android Studio.
2.	Importing and manipulating 3D objects.
3.	Creating a basic 3D scene in Unity.
4.	Developing a marker-based AR application using Vuforia.
5.	Developing a marker less AR application using AR Core/AR Foundation. Creating a simple VR environment using Unity XR.
6.	Implementing user interaction using controllers and gestures.
I	**Mini Project: Develop an AR or VR application for education, healthcare ,tourism, or gaming.**

Text Books:

1. Dieter Schmalstieg and To bias Hollerer, **Augmented Reality: Principles and Practice**,Addison-Wesley, 2016.
2. Steven M.LaValle, **Virtual Reality**, Cambridge University Press, 2017.
3. AlanB.Craig, **Understanding Augmented Reality**, MorganKaufmann,Elsevier.

TonyParisi, **Learning Virtual Reality**,O'ReillyMedia.

Reference Books:

1. Burdea,G.C.,andCoiffet,P.,**Virtual RealityTechnology**, Wiley.
2. JasonJerald,**The VR Book:Human-Centered Design for Virtual Reality**, ACM Books.
3. GrigoreC.Burdea, **Virtual Reality Systems and Applications**, Springer.
4. KellyS.HaleandKayM.Stanney,**Hand book of Virtual Environments**, CRCPress.

E-Resources:

1. Unity Learn–https://learn.unity.com
2. Unity Documentation–https://docs.unity3d.com
3. Google AR Core Documentation–https://developers.google.com/ar
4. Apple AR Kit Documentation–https://developer.apple.com/augmented-reality
5. Meta Quest Developer Hub–https://developer.oculus.com
6. Microsoft Mixed Reality Documentation–https://learn.microsoft.com/windows/mixed-reality
7. Coursera–AR&VR Courses–https://www.coursera.org
8. NPTEL–https://nptel.ac.in
9. SWAYAM–https://swayam.gov.in

Back to Structure

OPEN ELECTIVE - III

DEPARTMENT OF COMPUTER ENGINEERING

Program: B. Tech. (Computer Engineering)								**Semester:** V	
Course: Open Elective – III (Data Analytics)								**Code:** ALOE503A	
Teaching Scheme (Hrs/week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
02	-	-	02	-	-	50	50	-	100

Prerequisites:

Basic knowledge of Mathematics and Statistics, Fundamentals of Computer Science

Course Objectives:

1. To introduce the types and applications of data analytics across domains.
2. To apply descriptive statistics and visualization techniques to summarize data.
3. To apply inferential statistics and hypothesis testing for decision-making.
4. To analyze relationships in data using correlation, regression, and ANOVA.
5. To build predictive models using classification and clustering techniques.
6. To familiarize students with prescriptive analytics and Big Data concepts.

Course Outcomes: After completion of this course, students will be able to -

CO1	Explain the types of data analytics and their applications across domains.
CO2	Apply descriptive statistics and visualization techniques to summarize data.
CO3	Analyze data using inferential statistics and hypothesis testing methods.
CO4	Apply correlation, regression, and ANOVA techniques for data-driven decisions.
CO5	Design predictive models using classification and clustering approaches.
CO6	Evaluate prescriptive analytics approaches and Big Data platforms for applications.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to Data Analytics:** Types of analytics — descriptive, diagnostic, predictive, and prescriptive; Analytics vs. reporting; Key applications across domains: business, healthcare, finance, and marketing; Overview of analytics tools landscape: Excel, Python, R, and Power BI (comparative overview); KPIs and metrics for decision-making.	04
2.	**Descriptive Analytics:** Frequency distributions, outliers; Measures of central tendency and dispersion; Interquartile range; Skewness and kurtosis; Normal distribution and z-scores; Data visualization principles: bar chart, histogram, box plot, scatter plot, and heat map.	05
3.	**Inferential Statistics:** Population vs. sample, random sampling; Sampling distribution, standard error; Hypothesis testing: null and alternate hypothesis, z-test, t-test; One-tailed and two-tailed tests; Confidence intervals and level of confidence; Type I and Type II errors.	05

4.	**Correlation, Regression & ANOVA:** Correlation and scatter plots; Simple linear regression, least squares method; Multiple regression; Coefficient of determination (R^2); One-way and two-way ANOVA; Chi-square test of independence.	04
5.	**Predictive Analytics:** Introduction to predictive modeling; Classification techniques: logistic regression, decision trees; Clustering: k-means; Model evaluation: accuracy, confusion matrix; Time series analysis basics: moving averages, trend analysis.	04
6.	**Prescriptive & Big Data Analytics:** Prescriptive analytics: optimization and simulation for decision-making; Introduction to Big Data: the 5 V's, sources of big data; Overview of the Big Data ecosystem: Hadoop, Spark (conceptual only); Applications: business analytics, web analytics, and social media analytics.	04
	TOTAL	**26**

Term Work:

Sr. No.	Title	Proposed Tutorial Activities
1	Introduction to Data Analytics	Identify descriptive, diagnostic, predictive, and prescriptive analytics using real-world case studies. Compare Excel, Python, R, and Power BI for analytics tasks.
2	Descriptive Statistics – I	Calculate mean, median, mode, range, variance, and standard deviation using a given dataset in Excel/Python.
3	Descriptive Statistics – II	Detect outliers using IQR and visualize distributions using histograms and box plots.
4	Data Visualization	Create bar charts, scatter plots, heat maps, and box plots. Interpret insights from visualizations.
5	Sampling & Probability	Perform random sampling, calculate standard error, and differentiate between population and sample using datasets.
6	Hypothesis Testing	Perform z-test and t-test for given scenarios and interpret Type I and Type II errors.
7	Correlation & Regression	Calculate correlation coefficients, build simple linear regression models, and interpret R^2 values.
8	ANOVA & Chi-Square	Perform one-way ANOVA and Chi-square tests using statistical software and interpret results.

9	Predictive Analytics	Build basic classification models (Logistic Regression/Decision Tree) and perform K-Means clustering using Python.
10	Model Evaluation	Evaluate predictive models using confusion matrix and accuracy metrics.
11	Time Series Analytics	Apply moving averages and identify trends in time-series datasets.
12	Big Data & Prescriptive Analytics	Explore Hadoop and Spark concepts through case studies and discuss optimization-based decision-making.

Text Books:

1. Anil Maheshwari, "Data Analytics", McGraw Hill Education
2. Robert S. Witte and John S. Witte, "Statistics", Wiley Publications
3. U. Dinesh Kumar, "Business Analytics: The Science of Data-Driven Decision Making", Wiley India

Reference Books:

1. Seema Acharya and Subhashini Chellappan, "Big Data and Analytics", Wiley India
2. Allen B. Downey, "Think Stats: Exploratory Data Analysis in Python", Green Tea Press

E-Resources:

1. NPTEL Course on "Introduction to Data Analytics" (IIT Madras Track)
   Link: nptel.ac.in
2. NPTEL Course on "Data Analytics with Python" (IIT Roorkee Track)
   Link: https://nptel.ac.in/courses/106107220
3. NPTEL Course on "Business Intelligence & Analytics" (IIT Madras Track)
   Link: https://onlinecourses.nptel.ac.in/noc25_cs09/preview
4. NPTEL Course on "Business Analytics for Management Decision" (IIT Kharagpur Track)
   Link: https://nptel.ac.in/courses/110105089
5. Coursera: "Google Data Analytics Professional Certificate" Link: coursera.org
6. edX: "Data Science: Statistics and Machine Learning" by Harvard University
   Link: edx.org

Back to Structure

DEPARTMENT OF COMPUTER ENGINEERING

Program: B. Tech. (Computer Engineering)									**Semester: V**
Course: Introduction to Stock Market									**Code:** 24ALOE503D
Teaching Scheme (Hrs/week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
02	-	-	02	-	-	50	50	-	100

Prerequisites:

- Basic knowledge of economics and financial systems.
- Familiarity with percentage calculations and simple mathematics.
- Interest in investing and financial markets.

Course Objectives:

1. To understand the structure and functioning of financial and stock markets along with investor psychology.
2. To apply candlestick patterns and indicators for identifying market trends.
3. To understand the fundamentals of option trading and option pricing concepts.
4. To develop short-term and long-term investment strategies using fundamental and technical analysis.

Course Outcomes: After completion of this course, students will be able to -

CO1	Explain the functioning of financial markets using fundamental analysis	BL2
CO2	Analyze stock price movements using candlestick patterns for trading decisions.	BL4
CO3	Analyze stock price movements using technical indicators for trading decisions.	BL4
CO4	Describe the concepts of derivatives and option trading.	BL4
CO5	Apply the strategies for option trading.	BL3
CO6	Apply investment strategies based on market analysis.	BL3

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Fundamental Analysis and Investor Psychology** Introduction to Financial Markets, Primary and Secondary Markets, Market Participants, Stock Exchanges, SEBI and Regulatory Framework, Investment vs Trading, Understanding Business Models, Reading Financial Statements (Balance Sheet, Profit & Loss Account, Cash Flow Statement), Fundamental Ratios (EPS, PE Ratio, PB Ratio, ROE, ROCE, Debt-Equity Ratio), Sector Analysis, Economic Indicators, Investor Psychology, Market Emotions (Fear & Greed), Behavioral Biases, Risk Management Principles.	06
2.	**Technical Analysis using Candlestick Charts** Introduction to Technical Analysis, Dow Theory, Support and Resistance, Trend Analysis, Price Action Concepts Candlestick Anatomy : Single Candlestick Patterns (Doji, Hammer, Shooting Star, Marubozu, Spinning Top), Double Candlestick Patterns (Bullish & Bearish Engulfing, Harami, Piercing Pattern, Dark Cloud Cover), Triple Candlestick	04

	Patterns (Morning Star, Evening Star, Three White Soldiers, Three Black Crows), Gap Analysis, Trend Reversal and Continuation Patterns.	
3.	**Technical Analysis using Indicators** Technical Indicators : Moving Averages (SMA, EMA), MACD, RSI, Stochastic Oscillator, Bollinger Bands, VWAP, ADX, ATR, Volume Analysis, Fibonacci Retracement, Pivot Points, Trend Confirmation, Indicator-Based Trading Strategies, Combining Multiple Indicators, Entry and Exit Rules, Backtesting Concepts.	04
4.	**Introduction to Derivatives and FNO** Introduction to Derivatives, Futures and Options, Types of Options (Call and Put), Option Terminology (Strike Price, Expiry, Premium, Lot Size, Open Interest), Intrinsic and Time Value, ITM, ATM, OTM Options, Option Chain Analysis, Greeks (Delta, Gamma, Theta, Vega, Rho - Basic Concepts), Option Pricing Factors, Option Expiry and Settlement Process.	04
5.	**Introduction to Option Trading Strategies** Option Buying vs Option Selling, Risk and Reward Analysis, Long Call, Long Put, Covered Call, Cash Secured Put, Bull Call Spread, Bear Put Spread, Bull Put Spread, Bear Call Spread, Iron Condor (Basic Concept), Straddle and Strangle (Introduction), Position Sizing, Stop Loss Techniques, Capital Management, Risk Management in Options Trading.	04
6.	**Short-Term and Long-Term Trading & Investment Strategies** Swing Trading, Positional Trading, Intraday Trading (Overview), Momentum Trading, Breakout Trading, Trend Following Strategies, Portfolio Building, Value Investing, Growth Investing, Dividend Investing, SIP in Equity, Asset Allocation, Diversification, Trading Plan Development, Trading Journal, Performance Evaluation, Common Mistakes Made by Traders and Investors, Ethics in Financial Markets.	04
	Total	26

Term work: Any 10

1. Fundamental Analysis of a Company
Problem Statement:
Select one listed company and study its business, financial statements and important financial ratios. Based on the analysis, comment on the company's financial strength.

2. Investor Psychology and Market Emotions
Problem Statement:
Study a stock-market situation affected by fear, greed or panic. Identify the behavioral biases involved and explain how investors could manage such situations.

3. Candlestick Pattern Identification

Problem Statement:
Select a stock and identify different candlestick patterns from its historical price chart. Explain whether

each pattern indicates a possible buy, sell or trend-reversal signal.

4. Support and Resistance Analysis
Problem Statement:
Select a stock and identify its support, resistance and trend using a historical price chart. Explain how these levels can help in making trading decisions.

5. Technical Indicator Analysis
Problem Statement:
Select a stock and apply SMA/EMA, RSI and MACD to its historical price data. Identify the signals given by these indicators and comment on the stock's trend.

6. Simple Trading Strategy and Backtesting
Problem Statement:
Develop a simple trading strategy using two technical indicators. Apply the strategy to historical data and check whether the strategy generates profitable or loss-making trades.

7. Futures and Options Chain Analysis
Problem Statement:
Select a stock or index and study its Futures and Options chain. Identify the strike price, premium, expiry, open interest and ITM/ATM/OTM options, and explain their significance.

8. Option Trading Strategies
Problem Statement:
Study any four basic option trading strategies and explain their market conditions, maximum profit, maximum loss and break-even point using simple examples.

9. Portfolio Construction
Problem Statement:
Create a hypothetical portfolio of 5–8 stocks for a selected investor profile. Allocate the available virtual capital and explain the choice of stocks based on risk, return and diversification.

10. Paper Trading and Performance Analysis
Problem Statement:
Perform paper trading using virtual money for selected stocks over a specified period. Maintain a simple trading journal and calculate the overall profit/loss and performance.

Reference Books:

1. Murphy, John J. – *Technical Analysis of the Financial Markets*, Pearson.
2. Benjamin Graham – *The Intelligent Investor*, Harper Business.
3. Martin J. Pring – *Technical Analysis Explained*, McGraw-Hill.
4. Sheldon Natenberg – *Option Volatility and Pricing*, McGraw-Hill.
5. Aswath Damodaran – *Investment Valuation*, Wiley.
6. Alexander Elder – *Trading for a Living*, Wiley.

E-Resources:

1. National Stock Exchange (NSE) Learning Portal – https://www.nseindia.com ….. learn tab
2. Bombay Stock Exchange (BSE) Investor Education – https://www.bseindia.com
3. SEBI Investor Education – https://investor.sebi.gov.in
4. NISM (National Institute of Securities Markets) – https://www.nism.ac.in
5. Zerodha Varsity – https://zerodha.com/varsity
6. Investopedia – https://www.investopedia.com
7. TradingView – https://www.tradingview.com
8. Moneycontrol Learning – https://www.moneycontrol.com/education

Back to Structure

DEPARTMENT OF COMPUTER ENGINEERING

Program: B. Tech. (Computer Engineering)									**Semester:** V
Course: Open Elective – III (Green Building)									**Code:** ALOE503B
Teaching Scheme (Hrs/week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
02	-	-	02	-	-	50	50	-	100

Prerequisites:

Basic environmental engineering, building materials & construction, engineering chemistry/ environmental science, basic knowledge of energy and sustainability concepts

Course Objectives:

1. To equip students with the principles of sustainable building design, resource-efficient construction, and green building practices.
2. To provide students with knowledge of energy-efficient technologies, water and waste management, sustainable materials, and green building rating systems.

Course Outcomes: After completion of this course, students will be able to -

CO1	Explain the concepts, principles, benefits, and environmental impacts associated with green buildings and sustainable development.
CO2	Assess building orientation, climate-responsive planning, landscaping, green roofs, and green walls for improving environmental performance.
CO3	Evaluate building energy consumption and recommend energy-efficient strategies, HVAC optimization, renewable energy systems, and conservation measures.
CO4	Design suitable water conservation, rainwater harvesting, greywater recycling, waste management, and sustainable drainage solutions for buildings.
CO5	Select appropriate sustainable building materials and assess indoor environmental quality based on thermal comfort, indoor air quality, lighting, and acoustics.
CO6	Compare major green building rating systems such as IGBC, GRIHA, and LEED and apply their principles to assess building sustainability.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to Green Buildings:** Definition, concept, and need of green buildings, Sustainable development principles, Environmental impacts of conventional buildings, Benefits of green buildings.	05
2.	**Sustainable Planning & Architecture:** Sustainable planning, Orientation and climate-responsive design, Landscape planning, Green roofs and walls.	05

3.	**Energy Efficiency in Buildings:** Energy consumption in buildings, Energy-efficient building design, HVAC optimization, Renewable energy systems (solar, wind), Energy audit and conservation techniques.	04
4.	**Water & Waste Management:** Water conservation techniques, Rainwater harvesting systems, Greywater recycling, Waste management in buildings, Sustainable drainage systems.	04
5.	**Sustainable Materials & Indoor Environmental Quality:** Eco-friendly building materials, Low embodied energy materials, Recycled and locally available materials, Indoor air quality (IAQ), Thermal comfort, lighting, and acoustics.	04
6.	**Green Building Rating Systems & Regulations:** Overview of rating systems: Indian Green Building Council (IGBC) Green Rating for Integrated Habitat Assessment (GRIHA) Leadership in Energy and Environmental Design (LEED)	04
	TOTAL	**26**

Teamwork : (Any 10)

1. Study any IGBC/GRIHA certified building and identify the sustainable features adopted.
2. Study how sustainable development principles have been implemented in any residential, commercial, or institutional building.
3. Study a construction project and identify its environmental impacts. Suggest green alternatives to reduce these impacts.
4. Analyze building designed using passive architectural principles.
5. Study the site planning of a green campus, township, or commercial complex.
6. Study an existing building incorporating green roofs or vertical gardens.
7. Study an energy-efficient building and identify passive design strategies adopted.
8. Study a building utilizing renewable energy systems and evaluate their benefits.
9. Analyze an energy audit report (or published case study) and summarize the recommended energy-saving measures.
10. Study water conservation measures implemented in any green building.
11. Study a residential or institutional building with rainwater harvesting and grey water reuse systems.
12. Study the waste management practices of a smart city, residential township, or institutional campus.
13. Study a building that has extensively used sustainable building materials.
14. Perform a literature review on the life cycle assessment of one construction material.
15. Study an office, hospital, educational building, or green building focusing on indoor environmental quality improvements.

16. Compare two certified buildings, each evaluated under different rating systems (e.g., IGBC vs LEED or GRIHA vs IGBC).
17. Study the certification journey of an IGBC or GRIHA certified building.

Text Books:

1. K.S. Jagadeesh, "Alternative Building Materials and Technologies", New Age International.
2. C.J. Kibert, "Sustainable Construction: Green Building Design and Delivery", Wiley.
3. Charles J. Kibert, "Green Buildings: Principles and Practices", Island Press.

Environmental Legislations & Standards:

Indian Regulations

1. Environment (Protection) Act, 1986
2. Energy Conservation Act, 2001
3. National Building Code (NBC) of India
4. Eco-Niwas Samhita (Energy Conservation Building Code – Residential)

Standards & Guidelines

1. Energy Conservation Building Code (ECBC)
2. GRIHA & IGBC rating guidelines
3. LEED certification standards
4. Bureau of Energy Efficiency (BEE) norms

Reference Books:

1. O.H. Koenigsberger, "Manual of Tropical Housing and Building", Orient BlackSwan
2. G.Z. Brown and Mark DeKay, "Sun, Wind & Light", Wiley
3. Laurie Baker, "Building with Earth", Centre for Science and Environment
4. Indian Green Building Council, "IGBC Manuals and Guidelines", CII – Indian Green Building Council

E-Resources:

1. Indian Green Building Council - https://igbc.in
2. Green Rating for Integrated Habitat Assessment - https://grihaindia.org
3. U.S. Green Building Council - https://usgbc.org
4. NPTEL Green Building courses - https://nptel.ac.in
5. Ministry of Environment, Forest and Climate Change - https://moef.gov.in

Back to Structure

DEPARTMENT OF COMPUTER ENGINEERING

Program: B. Tech. (Computer Engineering)								**Semester:** V	
Course: Open Elective – III (Electrical Mobility)								**Code:** ALOE503C	
Teaching Scheme (Hrs/week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
02	-	-	02	-	-	50	50	-	100

Prerequisites:

Basic Electrical and Electronics Engineering.

Course Objectives:

1. To apply fundamental principles of electric mobility to interpret EV architectures, components and applications.
2. To utilize energy storage and battery management concepts to evaluate battery performance, charging, and safety.
3. To implement basic concepts of EV powertrains, electric drives, regenerative braking, and charging infrastructure.
4. To integrate smart charging, vehicle-to-grid, IoT, and emerging technologies for sustainable electric mobility.

Course Outcomes: After completion of this course, students will be able to -

CO1	Explain the concepts, architecture, and applications of Electric and Hybrid Electric Vehicles.
CO2	Identify the suitable energy storage systems used in electric vehicles.
CO3	Illustrate battery charging methods, battery management systems, and SoC estimation techniques.
CO4	Interpret hybrid powertrain configurations, operating modes, and regenerative braking.
CO5	Demonstrate the use of electric drives, charging levels, charging standards, and EV charging equipment.
CO6	Describe EV charging technologies and grid integration concepts such as G2V, V2G, V2H, V2B, and V2V.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to Electric Mobility:** Need for electric mobility, evolution of EVs, advantages and challenges, environmental and economic aspects, EV vs conventional vehicles, types of EVs: BEV, HEV, PHEV and fuel-cell vehicles, basic EV architecture and major components.	04
2.	**Energy Storage Systems:** Energy storage requirements, battery terminology and specifications, Li-ion batteries and their types, basic battery characteristics, battery alternatives: solid-state, Aluminium-air, fuel cell and supercapacitor, battery–supercapacitor hybrid system, selection of energy storage system.	04

3.	**Battery Charging and Management:** EV charging basics, AC and DC charging, charging methods and levels, charging time and factors affecting charging, Battery Management System (BMS), functions of BMS, State of Charge (SoC), State of Health (SoH), State of Power (SoP), cell balancing, battery thermal management and safety.	05
4.	**EV Powertrain and Electric Drives:** EV powertrain architecture, energy flow in EVs, vehicle performance and basic tractive effort, electric motors used in EVs: BLDC and PMSM, motor selection, basic drive control, regenerative braking and its benefits.	04
5.	**EV Charging Infrastructure:** EV charging infrastructure, charging levels, AC/DC chargers, EVSE, charging connectors and standards: CCS, CHAdeMO, SAE J1772, Bharat AC/DC, basic charging station components, installation and safety considerations.	04
6.	**Smart and Emerging Electric Mobility:** Grid-to-Vehicle (G2V), Vehicle-to-Grid (V2G), Vehicle-to-Home (V2H), Vehicle-to-Building (V2B), Vehicle-to-Vehicle (V2V), smart charging, wireless charging, battery swapping, energy management, IoT and digital technologies in EVs, future trends and challenges.	05
	TOTAL	**26**

Teamwork :

1. Study the architecture of an Electric Vehicle (EV) and identify the functions of major components such as battery, BMS, motor, inverter, charger, controller and sensors.
2. Compare 3–5 EVs available in India based on battery capacity, motor power, driving range, charging time, efficiency, cost and other key specifications.
3. Study and compare different energy storage technologies such as Li-ion batteries, solid-state batteries, fuel cells and supercapacitors based on energy density, charging time, life, cost, safety and applications.
4. Visit an EV charging station and study the charger type, AC/DC charging, charging level, connector/standard, power rating, and charging time and safety provisions.
5. Prepare a basic layout of an EV charging station showing charging points, power supply, protection system, parking area, cable routing, safety provisions and user facilities.
6. Study smart EV charging and Vehicle-to-Grid (V2G) concepts, including G2V, V2G, V2H and V2B, and identify their applications in smart grids and buildings.
7. Collect and analyze EV performance or charging data using Excel/Python/MATLAB and study parameters such as battery capacity, range, charging time and energy consumption.
8. Design a conceptual IoT-based EV monitoring system showing sensors, communication network, cloud platform and user interface for monitoring SoC, temperature, charging status and energy consumption.
9. Study regenerative braking in EVs and prepare an energy-flow diagram. Analyze energy recovery under different braking conditions using numerical calculations or simulation.
10. Study an emerging EV technology such as battery swapping, wireless charging, solid-state batteries, autonomous EVs, solar charging or second-life batteries and evaluate its technical, economic and environmental benefits.

Text Books:

1. James Larminie and John Lowry, "Electric Vehicle", John Wiley & Sons, 2012.
2. Ronald K. Jurgen, "Electric and Hybrid-Electric Vehicles", SAE International.
3. K. T. Chau, "Energy Systems for Electric and Hybrid Vehicles", The Institution of Engineering and Technology.
4. D. A. J. Rand, R. Woods, and R. M. Dell, "Batteries for Electric Vehicles", Research Studies Press Ltd., New York, John Wiley & Sons.
5. "Electric & Hybrid Vehicles – Design Fundamentals", CRC Press.

Reference Books:

1. Mehrdad Ehsani, Yimin Gao, and Ali Emadi, "Modern Electrical Hybrid Electric and Fuel Cell Vehicles: Fundamentals, Theory and Design", CRC Press, 2009.
2. Junwei Lu and Jahangir Hossain et al. (Eds.), "Vehicle-to-Grid: Linking Electric Vehicles to the Smart Grid", IET Digital Library.
3. Tom Denton, "Automobile Electrical and Electronic Systems", SAE International Publications.
4. C. Mi, M. A. Masrur, and D. W. Gao, "Hybrid Electric Vehicles: Principles and Applications with Practical Perspectives", John Wiley & Sons, 2011.
5. Mark Warner, "The Electric Vehicle Conversion Handbook", HP Books, 2011.

E-Resources:

1. https://www.theiet.org/resources/books/transport/vehicle2grid.cfm
2. https://www.sae.org/publications/books/content/pt-143.set/
3. http://nptel.ac.in/courses/108103009/

Back to Structure

DEPARTMENT OF COMPUTER ENGINEERING

Program: B. Tech. (Computer Engineering)								**Semester:** V	
Course: OE: Patent Law for Engineers and Scientist								**Code:** ALOE503E	
Teaching Scheme (Hrs/week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
02	-	-	02	-	-	50	50	-	100

Course Objectives: Students will be able To:

1. Understand the importance of Intellectual Property Rights (IPR) and patents.
2. Identify patentable and non-patentable inventions based on patentability criteria.
3. Conduct prior-art and patent searches using public patent databases.
4. Draft a basic patent specification, including claims and abstract.
5. Understand and apply the complete patent filing, examination, and granting process in India.
6. Analyze engineering case studies involving patents, innovation, and commercialization.

Course Outcomes: After completion of this course, students will be able to -

CO1	**Apply** the provisions of the **Indian Patent Law** to engineering inventions. **[BTL3]**
CO2	**Perform** prior-art and patent searches using standard patent databases. **[BTL3]**
CO3	**Analyze** the patentability, filing, examination, and grant process under the Indian Patent Act. **[BTL4]**
CO4	**Evaluate** patent infringement, licensing, technology transfer, and commercialization issues. **[BTL5]**
CO5	**Draft** a patent specification, including claims and abstract, in compliance with patent law. **[BTL6]**
CO6	**Develop** a patent application for an innovative engineering solution following legal and procedural requirements. **[BTL6]**

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to Intellectual Property Rights and Patents** Introduction to Intellectual Property Rights (IPR), significance of IPR in engineering, innovation and entrepreneurship, types of IPR, concept and objectives of patents, patentability criteria (novelty, inventive step and industrial applicability), patentable and non-patentable inventions under the Indian Patent Act, 1970, patents versus research publications and products, role of patents in technology commercialization and startup ecosystem. **Case Study:** Patentability analysis of a simple engineering innovation.	05

2.	**Indian Patent Law and Prior-Art Search** Overview of the Indian Patent Act, 1970 and its amendments, objectives and scope of patent law, rights and duties of inventors, overview of WIPO, Patent Cooperation Treaty (PCT) and international patent systems, prior-art search techniques, patent classification systems, patent search using Google Patents, InPASS, Espacenet, and WIPO PATENTSCOPE, novelty assessment and patent landscape analysis. **Case Study**: Prior-art search and novelty analysis of an engineering invention.	04
3.	**Patent Drafting** Structure and components of a patent specification including title, field of invention, background, prior art, problem statement, objectives, summary of invention, brief description of drawings, detailed description, claims, abstract, and drawings. Principles of claim drafting, dependent and independent claims, patent drafting guidelines, common drafting errors, and patent documentation for engineering innovations. Case Study: Drafting of a patent specification for a smart engineering product.	05
4.	**Patent Filing, Examination and Grant** Patent filing procedure in India, provisional and complete specification, online filing process, statutory forms, publication of patent application, request for examination, First Examination Report (FER), response to objections, hearing process, patent grant, renewal, restoration, opposition, and maintenance of patents under the Indian Patent Act. **Case Study:** Patent filing and grant timeline of an Indian engineering patent.	04
5.	**Patent Commercialization and Legal Issues** Patent ownership, licensing, assignment, technology transfer, patent valuation, startup and university patents, patent infringement, remedies and dispute resolution, freedom-to-operate (FTO), patent analytics, commercialization strategies, and ethical issues in patent protection. **Case Study:** Commercialization of a patented engineering innovation.	04

6.	**Emerging Technologies and Patent Case Studies** Patents in Artificial Intelligence, Machine Learning, Internet of Things (IoT), Robotics, Semiconductor Technology, Biotechnology, Software, Green Technologies, and Industry 4.0, AI-generated inventions, recent developments in patent law, international patent trends, patent portfolio management, and innovation-driven entrepreneurship. **Case Study:** Analysis of landmark patent disputes such as Apple vs Samsung or an Indian engineering innovation, along with preparation of a basic patent draft for an engineering solution.	04
	TOTAL	**26 Hrs.**

Term Work:

Students shall complete the following practical, activity-oriented assignments based on a selected engineering innovation/project:

1. Identify suitable IP protection for an engineering innovation and assess basic patentability criteria.
2. Conduct patent searches using Google Patents, IP India, WIPO PATENTSCOPE and Espacenet and analyze the novelty of the selected innovation.
3. Identify a copyrightable work such as software, technical content or digital work and prepare/undertake the copyright registration process.
4. Develop a product/start-up brand, conduct trademark search, identify the appropriate class and prepare a basic trademark application.
5. Identify protectable visual features of an engineering product, conduct design search and prepare a basic design registration application.
6. Prepare a basic patent specification including title, background, objectives, description, drawings, abstract and claims for an engineering innovation.
7. Prepare a mock provisional/complete patent application and study the filing, publication, examination, FER, response and grant process in India.
8. Study the PCT route and prepare a mock PCT application based on the selected engineering invention.
9. Analyze patents in a selected emerging technology area and identify major patent holders, technology trends and basic FTO considerations.
10. Develop an IP portfolio for the selected innovation covering patent, copyright, trademark/design and propose a basic commercialization strategy.

Text Books:

1. P. Narayanan, *Patent Law*, Eastern Law House.
2. Prabuddha Ganguli, *Intellectual Property Rights: Unleashing the Knowledge Economy*, McGraw-Hill.
N. S. Gopalakrishnan and T. G. Agitha, *Principles of Intellectual Property*, Eastern Book Company.
The Patents Act, 1970 (as amended) and The Patents Rules, 2003 (as amended), Gov.of India.

Reference Books:

1. David Pressman, *Patent It Yourself*, NOLO Publications.
2. WIPO, *WIPO Intellectual Property Handbook: Policy, Law and Use*, World Intellectual Property Organization.
3. Indian Patent Office, *Manual of Patent Office Practice and Procedure*, Office of the Controller General of Patents, Designs & Trade Marks (CGPDTM), Government of India.
4. Alexander I. Poltorak and Paul J. Lerner, *Essentials of Intellectual Property*, John Wiley & Sons.

E-Resources:

1. **Patent Law for Engineers and Scientists** – NPTEL (IIT Madras): https://nptel.ac.in/courses/110106081
2. **Intellectual Property** – NPTEL (IIT Madras): https://nptel.ac.in/courses/109106137

Back to Structure

DEPARTMENT OF COMPUTER ENGINEERING

Program: B.Tech. Computer Engineering								**Semester: V**	
Course: Fundamentals of IoT								**Code:COMD503**	
Teaching Scheme(Hrs/week)				**Evaluation Scheme(Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
02	02	-	2	-	-	25	-	25	50

Prerequisites:

1. C/C++ or Python basics
2. Basic Computer Networks
3. Basic electronics
4. Basic Internet concepts

Course Objectives:

1. Understand the concepts, architecture, components, enabling technologies, and design methodology of the Internet of Things (IoT).
2. Understand and implement the role of sensors, actuators, microcontrollers, embedded systems, and edge devices in IoT solutions.
3. Apply IoT networking and communication technologies and select suitable protocols based on application requirements such as range, power, bandwidth, and reliability.
4. Develop IoT solutions for data acquisition, edge processing, cloud integration, visualization, and basic analytics.
5. Understand IoT security, privacy, device identity, authentication, secure communication, firmware security, and device management practices.
6. Design and evaluate simple end-to-end IoT solutions for smart home, agriculture, healthcare, industrial, energy, environmental, and other real-world applications

Course Outcomes: After completion of this course, students will be able to-

CO1	Explain the architecture, components, workflow, and enabling technologies of IoT systems.
CO2	Interface and program sensors, actuators, microcontrollers, and edge devices for IoT applications.
CO3	Apply and compare IoT connectivity and communication protocols for different application requirements.
CO4	Develop IoT applications involving data acquisition, edge processing, cloud integration, visualization, and basic analytics.
CO5	Analyze IoT security, privacy, authentication, secure communication, firmware, and device-management requirements.
CO6	Design an end-to-end IoT solution for a real-world smart application and evaluate its scalability, reliability, and intelligent features.

Course Contents:

Unit	**Description**	**Duration**

		(Hrs.)
1.	**Fundamentals of Internet of Things:** Introduction to IoT, need and evolution, definition and characteristics, components of IoT, sensors, actuators and connected devices, IoT ecosystem and workflow, IoT architecture and layers, applications, benefits and challenges, basic IoT design methodology.	5
2.	**IoT Devices, Sensors and Actuators:** Fundamentals of embedded systems, microcontrollers and microprocessors, Arduino and ESP32 development boards, Raspberry Pi as an edge device, digital and analog signals, common sensors and actuators, basic programming and sensor/actuator interfacing.	5
3.	**IoT Connectivity, Networking and Communication Protocols:** Basic networking concepts, IP addressing and IPv6 overview, wired and wireless connectivity, Wi-Fi, BLE, Zigbee/Thread and LoRaWAN concepts, IoT gateways, MQTT, HTTP/REST and CoAP, publish-subscribe and request-response models.	5
4.	**IoT Data, Edge Computing and Cloud Integration:** IoT data lifecycle, data acquisition and transmission, basic data processing and filtering, edge computing and device-edge-cloud architecture, cloud fundamentals, device-to-cloud communication, IoT cloud platforms, data storage, visualization and dashboards, introduction to IoT analytics and Digital Twin concepts.	5
5.	**IoT Security, Privacy and Device Management:** Need for IoT security, attack surfaces, common threats and vulnerabilities, security requirements, device identity, authentication and authorization, access control, encryption and secure communication, privacy, security-by-design and basic threat modeling, device provisioning and monitoring.	5
6.	**IoT Applications and Intelligent Connected Systems:** End-to-end IoT system design, Smart Home, Smart Agriculture, Smart Healthcare and Smart City applications, Industrial IoT (IIoT), remote monitoring and predictive maintenance, Digital Twin applications, Edge AI/TinyML concepts, scalability and reliability, current IoT trends, design of a simple IoT solution.	5
	TOTAL	**30**

Practical List: Any 8 practicals (Mini project mandatory) **Platform / Tools: Arduino / ESP32, Raspberry Pi + Python**	
1	Introduction to IoT Device and Sensor Interfacing: Interface an LED, push button and basic sensor with an Arduino/ESP32 and develop a simple sensor-based application..
2	Analog and Digital Sensor Interfacing: Interface sensors such as LDR, temperature and humidity sensor with Arduino/ESP32 and display the sensor readings.
3	Actuator Control using Arduino/ESP32: Interface an actuator such as a DC motor, servo motor or relay and control it based on sensor input.
4	Raspberry Pi as an IoT Edge Device: Configure Raspberry Pi, interface a sensor, read sensor data using Python and display/store the readings locally.
5	IoT Communication using MQTT: Establish publish-subscribe communication between an Arduino/ESP32 and Raspberry Pi using an MQTT broker and exchange sensor data.

6	IoT Device-to-Cloud Data Transmission: Send sensor data from Arduino/ESP32 or Raspberry Pi to an IoT cloud platform and create a basic dashboard for data visualization.
7	Edge Computing and Local Data Processing: Use Raspberry Pi to collect sensor data, perform basic filtering/processing, generate alerts based on threshold values, and visualize the processed data.
8	IoT Security Implementation: Implement basic device authentication and secure communication for an IoT system and study common IoT security threats and protection techniques.
9	IoT-Based Remote Monitoring System: Develop a system using sensors and Raspberry Pi/ESP32 to remotely monitor parameters such as temperature, humidity, light or motion, with alerts when predefined limits are exceeded.
10	Intelligent IoT System with Anomaly Detection: Collect sensor data using Arduino/ESP32 or Raspberry Pi and implement a basic rule-based or simple ML-based method to identify abnormal sensor readings.
I	**Mini Project:** Design and implement an end-to-end IoT system that collects environmental data using sensors, processes the data at the edge, sends it to the cloud, and provides monitoring and control capabilities.

Text Books:

1. Arshdeep Bahga and Vijay Madisetti, **Internet of Things –A Hands-on Approach**, UniversitiesPress, Latest Edition.
2. RajKamal, **Internet of Things :Architecture and Design Principles** ,McGraw-HillEducation.
3. DavidHanes, GonzaloSalgueiro,PatrickGrossetete,RobertBartonand JeromeHenry, **IoT Fundamentals: Networking Technologies, Protocols and Use Cases for the Internet of Things**, Cisco Press.
4. SrinivasaK.G.,SiddeshG.M.,andHanumanthaRaju R.,**Internet of Things**,CengageLearning.

Reference Books:

1. OlivierHersent,DavidBoswarthickandOmarElloumi,TheInternetofThings:KeyApplications and Protocols, Wiley.
2. AdrianMcEwenand HakimCassimally,DesigningtheInternetofThings,Wiley.
3. SamuelGreengard, TheInternetofThings,MITPress.
4. CunoPfister, GettingStartedwiththeInternetofThings, O'ReillyMedia.
5. MaciejKranz, BuildingtheInternetofThings,Wiley.

E-Resources:

1. https://www.iotforall.com/ebooks/an-introduction-to-iot
2. https://www.qorvo.com/design-hub/ebooks/internet-of-things-for-dummies
3. https://www.udemy.com/course/a-complete-course-on-an-iot-system-design-anddevelopment/
4. https://www.coursera.org/learn/iot
5. https://nptel.ac.in/courses/108/108/108108098/
6. https://nptel.ac.in/courses/106/105/106105166/

Back to Structure

SYLLABUS

SEMESTER - VI

DEPARTMENT OF COMPUTER ENGINEERING

Program: B. Tech. (Computer Engineering)								**Semester:** VI	
Course: Design and Analysis of Algorithms								**Code:** COPC611	
Teaching Scheme (Hrs/week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	02	-	04	40	60	-	25	-	125

Prerequisites:

1. Data Structures
2. Programming fundamentals and problem solving
3. fundamental programming in C/C++/Java

Course Objectives:

1. To explain fundamental concepts of algorithms, algorithmic problem solving, correctness, and complexity analysis
2. To apply divide-and-conquer and greedy techniques to solve computational problems.
3. To analyze dynamic programming, hashing, and graph algorithms for efficient problem solving. To evaluate backtracking and constraint-satisfaction techniques for solving combinatorial problems.

Course Outcomes: After completion of this course, students will be able to -

CO1	To analyze dynamic programming, hashing, and graph algorithms for efficient problem solving. To evaluate backtracking and constraint-satisfaction techniques for solving combinatorial problems.
CO2	Apply divide-and-conquer and greedy techniques to solve searching, sorting, scheduling and optimization problems.
CO3	Analyze dynamic programming and hashing techniques for solving optimization and efficient data-retrieval problems.
CO4	Analyze graph algorithms for traversal, shortest-path and network optimization problems.
CO5	Evaluate backtracking and constraint-satisfaction techniques for solving combinatorial search problems.
CO6	Create suitable solutions using branch-and-bound techniques and classify computational problems using P, NP, NP-Hard and NP-Complete concepts.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to Algorithmic Problem Solving** Introduction and characteristics of algorithms; algorithm specification and pseudo code; algorithm correctness and loop invariants; performance analysis; time and space complexity; best-case, average-case and worst-case analysis; asymptotic notations -Big-O, Ω and Θ; searching techniques -linear search and binary search; sorting techniques including insertion sort, selection sort, bubble sort, heap sort introduction to algorithm visualization and AI-assisted code	06

	analysis.	
2.	**Divide and Conquer and Greedy Method** Divide and conquer strategy, recurrence relations; binary search; merge sort; quick sort; Strassen's matrix multiplication, analysis of divide-and-conquer algorithms; greedy method and greedy-choice property, optimal substructure, fractional knapsack; job sequencing with deadlines, minimum spanning trees using Prim's and Kruskal's algorithms, comparison and application of greedy strategies to optimization problems.	07
3.	**Dynamic Programming and Hashing** Introduction to dynamic programming, overlapping sub problems and optimal substructure, top-down and bottom-up approaches, recurrence formulation, 0/1 knapsack problem, subset sum problem comparison of dynamic programming with greedy techniques; hash tables; hash functions, collision resolution techniques including chaining, double hashing, analysis of hashing performance and applications in efficient data retrieval.	08
4.	**Graph Algorithms** Introduction to graphs and graph terminology, directed and undirected graphs; weighted and unweight graphs; graph representations using adjacency matrix and adjacency list; graph traversal using Breadth First Search and Depth First Search, shortest-path algorithms including Dijkstra's algorithm, Bellman-Ford algorithm and Floyd-Warshall algorithm	07
5.	**Backtracking and Constraint Satisfaction** General method and terminology of backtracking, state-space tree, recursive implementation, promising and non-promising nodes; constraint satisfaction problems; N-Queens problem; Hamiltonian cycle problem; Sum of Subsets problem; Sudoku solving, backtracking for combinatorial search problems	07
6.	**Branch and Bound, Complexity Theory** General method and terminology of branch and bound; state-space tree; control abstraction; bounding functions; Least Cost Search; FIFO and Least Cost Branch and Bound; applications to 0/1 knapsack, Travelling Salesperson Problem; comparison of backtracking and branch-and-bound techniques; introduction to computational complexity; polynomial and non-polynomial problems; complexity classes P, NP, NP-Hard and NP-Complete	07
	TOTAL	**42**

No	**Practical / Problem Statement**
1	Implementation and Performance Analysis of Linear Search and Binary Search
2	Implementation and Comparative Analysis of Sorting Algorithms: Insertion Sort, Selection Sort, Bubble Sort, Heap Sort, Counting Sort, Radix Sort and Bucket Sort
3	implementation of Divide and Conquer Algorithms: Merge Sort, Quick Sort and Binary

	Search
4	Implementation of Greedy Algorithms: Fractional Knapsack, Job Sequencing and Minimum Spanning Tree using Prim's and Kruskal's Algorithms
5	Implementation of Dynamic Programming Algorithms: 0/1 Knapsack and Subset Sum
6	Implementation of Hashing Techniques with Chaining and Open Addressing.
7	Implementation and Analysis of Graph Algorithms: BFS, DFS, Dijkstra, Bellman-Ford
8	Implementation of Backtracking Algorithms: N-Queens, Hamiltonian Cycle and Sum of Subsets
9	implementation of Branch and Bound Algorithms for 0/1 Knapsack and Travelling Salesperson Problem
10	AI-Assisted Algorithm Design and Optimization: Use of AI tools to develop, analyze, optimize and compare solutions for a selected computationally difficult problem

Text Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, 4th Edition, MIT Press
2. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, Computer Algorithms, Computer Science Press..
3. Seymour Lipschutz, George S. Mohanty and others, Data Structures, Schaum's Outline Series, McGraw-Hill

Reference Books:

1. Anany Levitin**,** *Introduction to the Design & Analysis of Algorithms*, 3rd Edition, Pearson.
2. Robert Sedgewick and Kevin Wayne**,** *Algorithms*, 4th Edition, Addison-Wesle
3. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman**,** *The Design and Analysis of Computer Algorithms*, Addison-Wesley.

E-Resources:

2. https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-fall-2011
3. https://www.geeksforgeeks.org/fundamentals-of-algorithms
4. https://www.w3schools.com/dsa

Back to Structure

DEPARTMENT OF COMPUTER ENGINEERING

Program: B.Tech.(Computer Engineering)									**Semester:** VI	
Course: Deep Learning									**Code:COPC612**	
Teaching Scheme				**Evaluation Scheme**						
Lecture	**Practical**	**Tutorial**	**Credit**		**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	02	-	04		40	60	25	-	25	150

Prerequisites:

1. Basics of Python Programming
2. Fundamentals of Machine Learning
3. Knowledge of Linear Algebra, Probability and Statistics
4. Basic understanding of Data Structures and Algorithms

Course Objectives:

1. To understand the fundamentals of deep learning and artificial neural networks.
2. To learn different deep learning architectures such as ANN, CNN and RNN.
3. To study optimization techniques, activation functions and training mechanisms used in deep neural networks.
4. To implement deep learning models using modern frameworks.
5. To apply deep learning techniques for real-world applications such as image processing, NLP and speech recognition.
6. To develop analytical and problem-solving skills for AI-based systems.

Course Outcomes:

On completion of the course, learner will be able to-

CO1	Understand the fundamentals of deep learning and neural networks
CO2	Apply activation functions, optimization techniques and back propagation algorithms
CO3	Design and implement Artificial Neural Networks for prediction and classification tasks.
CO4	Develop Convolutional Neural Network models for image processing applications
CO5	Build Recurrent Neural Networks for sequential and time-series data analysis.
CO6	Implement deep learning solutions using Tensor Flow and PyTorch frameworks for real-world applications

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to Deep Learning:** Introduction to Artificial Intelligence, Machine Learning and Deep Learning, Biological Neuron, Artificial Neuron, Perceptron Model, Activation Functions, Types of Neural Networks, Introduction to Deep Learning Applications.	06

2.	**Neural Network Fundamentals:** Feed Forward Neural Networks, Multilayer Perceptron, Forward Propagation, Back propagation Algorithm, Gradient Descent, Loss Functions, Optimization Techniques, Hyper parameter Tuning, Regularization Techniques	07
3.	**Convolutional Neural Networks (CNN):** Introduction to CNN, Convolution Operation, Pooling, Padding, CNN Architectures, Image Classification, Object Detection, Transfer Learning, CNN Applications in Computer Vision. Duration:	07
4.	**Recurrent Neural Networks (RNN):** Sequential Data Processing, RNN Architecture, Vanishing Gradient Problem, Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Applications in NLP and Time-Series Forecasting.	07
5.	**Deep Learning Frameworks and Advanced Concepts:** Introduction to Tensor Flow and PyTorch, GPU Computing, Model Evaluation Metrics, Auto encoders, Generative Adversarial Networks (GAN), Reinforcement Learning Basics	08
6.	**Deep Learning Applications and Case Studies:** Natural Language Processing, Speech Recognition, Recommendation Systems, Healthcare Applications, Autonomous Systems, Industry Case Studies and Mini Project Development.	07
	TOTAL	42

List of Practical's :

Practical Platforms: Python Programming Environment using Jupyter Notebook/Google Colab with Tensor Flow, Keras, and PyTorch Libraries on GPU-enabled Systems.

1.	Install, configure and verify TensorFlow and PyTorch deep learning frameworks in a Python environment (Anaconda/Jupyter Notebook/Google Colab) with GPU support. **Problem Statement:** Set up a complete deep learning development environment by installing TensorFlow (with Keras) and PyTorch using pip/conda, configure GPU acceleration (CUDA/cuDNN drivers or Colab's built- in GPU runtime),and write a verification script that checks the installed versions, confirms GPU availability (tf.config.list_physical_devices('GPU') and torch.cuda.is_available()), and runs a simple tensor operation on both frameworks to confirm the setup is functional. **Tools/Platform:**Python3.x,JupyterNotebook/GoogleColab,TensorFlow,Keras,PyTorch,CUDA Toolkit (if local GPU). **Expected Outcome:** Successfully configured environment with both frameworks working correctly and GPU acceleration verified and available for subsequent experiments.
2.	Implement a Single-Layer Perceptron from scratch to understand the fundamental building block of a neural network. **Problem Statement:** Step/threshold activation function, and the perceptron learning rule (weight update: $w = w + \eta(y_target - y_predicted)x$) to train the model until convergence. Plot the decision boundary before and after training, and demonstrate that the model fails to converge on a non-linearly-separable problem (e.g.,XOR), highlighting the perceptron's limitation. **Tools/Platform:** Python, NumPy, Matplotlib.

	Expected Outcome: A working perceptron that correctly classifies AND/OR gate inputs, with a visualized decision boundary, and an understanding of why a single perceptron cannot solve the XOR problem.
3.	Design and implement a Multilayer Perceptron(MLP) using Keras/Tensor Flow or PyTorch to solve a non-linearly separable classification problem. **Problem Statement:** Build an MLP with one or more hidden layers to solve the XOR problem (or a real-world classification dataset such as the Iris dataset or a binary classification dataset). Design the network architecture (input layer, hidden layer(s) with appropriate neurons, output layer), select an appropriate loss function (categorical/binary cross-entropy) and optimizer (SGD/Adam), train the model using back propagation, and evaluate its performance using accuracy, confusion matrix, and loss/accuracy curves across epochs. **Tools/Platform:** Python, TensorFlow/Keras or PyTorch, NumPy, Matplotlib, Scikit-learn (for dataset and metrics). **Expected Outcome:** A trained MLP model that successfully classifies the given dataset (solving XOR, or achieving good accuracy on Iris/other dataset), with visualized training/validation performance curves.
4.	To study various activation functions used in neural networks and implement/visualize their behaviour and effect on model training. **Problem Statement:** Implement and plot the mathematical curves of common activation functions — Sigmoid, Tanh, ReLU, Leaky ReLU, and Softmax — along with their derivatives, using Python. Then build a simple neural network(e.g., MLP on MNIST/Iris) and train multiple versions of it, each using a different activation function in the hidden layers, to compare their impact on convergence speed, training accuracy, and problems such as vanishing gradients (observed with Sigmoid/Tanh) versus faster convergence (observed with ReLU). **Tools/Platform:** Python, NumPy, Matplotlib, TensorFlow/ Keras or PyTorch. **Expected Outcome:** Graphical plots of each activation function and its derivative, along with a comparative analysis (table/graph) of model accuracy and training speed for each activation function.
5.	Design and implement a Convolutional Neural Network for classifying images into multiple categories. **Problem Statement:** Build a CNN architecture consisting of convolutional layers (feature extraction using filters/kernels), pooling layers(down-sampling using Max/Average Pooling), and fully connected dense layers (classification), to classify images from a standard dataset such as CIFAR-10 (10 classes: airplane,car,bird,cat,etc.).Preprocessthedataset(normalization,one-hot encoding of labels),train the CNN using an appropriate optimizer and loss function, and evaluate the model's performance using accuracy, loss curves, and a confusion matrix. Optionally apply data augmentation to improve generalization.

	Tools/Platform: Python, TensorFlow/Keras or PyTorch, CIFAR-10 dataset, Matplotlib, Google Colab (GPU runtime recommended). **Expected Outcome:** A trained CNN model that classifies test images with reasonable accuracy (typically 70%+ on CIFAR-10 with a basic CNN), along with visualized sample predictions and performance metrics.
6.	Implement a CNN model for recognizing handwritten digits using the MNIST dataset. **Problem Statement:** Using the MNIST dataset (60,000 training images and 10,000 test images of handwrittendigits0-9,each 28×28 gray scale),design and train a CNN with convolutional layers, pooling layers, dropout(for regularization),and a Softmax output layer to classify each image into one of 10 digit classes. Evaluate the trained model on the test set using accuracy and a confusion matrix, and test the model's prediction on a few custom/sample handwritten digit images to demonstrate real-world applicability. **Tools/Platform:** Python, TensorFlow/Keras or PyTorch, MNIST dataset (built into both frameworks), Matplotlib. **Expected Outcome:** A CNN model achieving high accuracy (typically 98%+) on the MNIST test set, with visualized correct and misclassified digit predictions.
7.	Mini project based on real-world deep learning application. **Tools/Platform:** Python, Tensor Flow/Keras or PyTorch, relevant public dataset(Kaggle/UCI repository),Matplotlib/Seaborn for visualization, Google Colab(GPU). **Expected Outcome:** A complete, Working deep learning application with a documented report demonstrating the full pipeline from problem identification to model evaluation, showcasing practical mastery of the course concepts.

Text Books:

1. Ian Goodfellow, Yoshua Bengio and Aaron Courville, Deep Learning, 1st Edition, 2016, MIT Press.
2. François Chollet, Deep Learning with Python, 2nd Edition, 2021, Manning Publications.
3. Charu C. Aggarwal, Neural Networks and Deep Learning: A Textbook, 1st Edition, 2018, Springer.
4. Simon Haykin, Neural Networks and Learning Machines, 3rd Edition, 2008, Pearson.

Reference Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, 4th Edition, 2022, MIT Press.
2. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, Computer Algorithms, 2nd Edition, 1998, Computer Science Press.
3. Seymour Lipschutz, George S. Mohanty and others, Data Structures, Schaum's Outline Series, 2nd Edition, 1996, McGraw-Hill.

E-Resources:

1. **TensorFlow/Keras:** Official documentation, tutorials, guides, and API references.
   TensorFlow Official Documentation
2. **PyTorch:** Official documentation, tutorials, guides, and API references.

PyTorch Official Documentation

3. **Keras:** Official Keras documentation, examples, and developer guides.
   Keras Official Documentation
4. **Google Colab:** Cloud-based environment for running Python and deep-learning experiments.
   Google Colab

Back to Structure

PROGRAM ELECTIVE - II

(Select any one course from the given Program Elective Courses)

\

DEPARTMENT OF COMPUTER ENGINEERING

Program: B.Tech.(Computer Engineering)								**Semester: VI**	
Course: Cloud Computing								**Code:COPE602A**	
Teaching Scheme (Hrs/week)				**Evaluation Scheme(Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	02	1	04	40	60	-	-	25	125

Prerequisites:

1. Computer Networks and Internet fundamentals
2. Basic understanding of software development and version.

Course Objectives:

1. To understand the fundamental concepts of cloud computing, including deployment models and virtualization.
2. To apply cloud infrastructure and services using platforms such as AWS, Microsoft Azure, and Google Cloud.
3. To analyze cloud-native technologies, data migration strategies, cloud architectures, scalability, security, governance, and cost optimization.

Course Outcomes: After completion of this course, students will be able to-

CO1	Explain the fundamental concepts, service models, deployment models, architecture, virtualization, and containerization used in cloud computing.
CO2	Apply virtualization techniques, cloud infrastructure, containers, and cloud services to develop scalable cloud solutions.
CO3	Analyze cloud services, APIs, load balancing, auto-scaling, serverless computing, and cloud-native architectures using AWS, Azure, and GCP**.**
CO4	Analyze cloud-native application development, microservices, Kubernetes, DevOps, CI/CD, and 6R-based data migration strategies.
CO5	Evaluate cloud security, IAM, encryption, governance, compliance, SLA, cost optimization, disaster recovery, Zero Trust, and multi-cloud solutions.
CO6	Create AI-enabled cloud applications using LLMs, prompt engineering, RAG, vector databases, AI agents, tool/function calling, and API integration.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Cloud Computing Fundamentals** Evolution and characteristics of cloud computing Cloud service models: IaaS, PaaS, SaaS ,Deployment models: Public, Private, Hybrid and Community Cloud ,Cloud architecture and essential components ,Virtualization and hypervisors, Containers and containerization concepts Benefits, challenges and applications of cloud computing, Public, Private, Hybrid and Community	04

2.	**Cloud Infrastructure and Virtualization** Virtualization Fundamentals: the concept of virtualization to cloud computing environments, the role of hypervisors and virtual machines. Type 1 (Bare-Metal) and Type 2 (Hosted) hypervisors with suitable examples and use cases. Virtualization Techniques: Full Virtualization, Para-Virtualization and Hardware-Assisted Virtualization types of virtualization –Network, Storage server	05
3.	**Cloud Infrastructure and Services:** virtual machines, AWS**:**EC2, S3, VPC, EBS, RDS and Cloud Watch Microsoft Azure Services**:** Virtual Machines, Blob Storage, Virtual Network, Azure Functions and Azure Monitor Google Cloud Platform (GCP): Compute Engine, Cloud Storage, VPC, Cloud Functions and Cloud Monitoring, containers and Docker, cloud APIs, load balancing, auto-scaling and server less computing.	05
4.	**Cloud-Native Computing and Data Migration:** Microservices, Kubernetes, DevOps and CI/CD, cloud application architecture, containerization and container orchestration, data migration strategies, 6R Migration Techniques, migration planning, ETL and data transfer, database migration, cloud-native databases and distributed storage, backup and recovery, disaster recovery and business continuity, scalability, availability and performance analysis.	05
5.	**Cloud Security Governance** Cloud security policies and organizational policies ,IAM, authentication, encryption, Cloud Governance, policies and compliance, SLA ,SLA life cycle ,cost management and optimization, monitoring, Cloud governance framework and governance models ,Data privacy and protection in cloud computing Data sovereignty, data residency and data localization , GDPR (General Data Protection Regulation) ,HIPAA (Health Insurance Portability and Accountability Act) – healthcare data, multi-cloud, edge computing and development of secure Cloud-AI solutions. Zero Trust Architecture	05
6.	**AI Integrated Cloud Computing and Emerging Technologies** AI and Cloud Computing, AI-as-a-Service (AIaaS), cloud-based ML platforms, Generative AI and LLMs, prompt engineering and RAG, MLOps and AI model deployment, Edge/Fog Computing and Edge AI, Serverless and event-driven AI, Cloud-IoT integration, AIOps, Digital Twins, confidential computing, blockchain/Web3 and quantum computing in the cloud, Responsible AI, and industry-oriented applications.	06
	TOTAL	**30**
No	**Practical / Problem Statement**	
1	**Cloud Service Exploration:** Explore AWS/Azure/GCP cloud platforms and identify the major compute, storage, networking, database and serverless services	
2	**Virtual Machine Deployment:** Create, configure and deploy a virtual machine on AWS	

	EC2/Azure VM and access it remotely.
3	**Cloud Storage Management:** Create a cloud storage bucket, upload/download objects, configure access permissions and implement basic lifecycle management using AWS S3/Azure Blob Storage.
4	**Virtual Network Configuration:** Design and configure a Virtual Private Cloud (VPC) with subnets, routing tables, security groups and internet connectivity.
5	**Cloud Application Deployment:** Deploy a web application on a cloud platform and configure it for public access and basic scalability.
6	**Load Balancing and Auto Scaling:** Configure a load balancer and auto-scaling mechanism to distribute application traffic and maintain service availability.
7	**Cloud Data Migration:** Migrate a sample dataset/application from a local environment to a cloud platform and evaluate the migration process.
8	**Cloud Security and IAM:** Create users, roles and access policies using AWS IAM/Azure IAM and implement the principle of least privilege.
9	**Serverless Computing:** Develop and deploy an event-driven function using AWS Lambda/Azure Functions and integrate it with a cloud service.
10	**Cloud CI/CD Pipeline:** Implement an automated CI/CD pipeline using GitHub Actions/Jenkins for building, testing and deploying a cloud application.

Text Books:

1. Rajkumar Buyya, James Broberg and Andrzej M. Goscinski, Cloud Computing: Principles and Paradigms, Wiley.
2. Thomas Erl, Ricardo Puttini and Zaigham Mahmood, Cloud Computing: Concepts, Technology & Architecture, Pearson.
3. Kai Hwang, Geoffrey C. Fox and Jack J. Dongarra, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Morgan Kaufmann.
4. George Reese, Cloud Application Architectures: Building Applications and Infrastructure in the Cloud, O'Reilly Media

Reference Books:

1. Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, Shivananda Poojara and Satish Narayana Srirama, *Mastering Cloud Computing: Foundations and Applications Programming*, 2nd Edition, Morgan Kaufmann/Elsevier, 2026.
2. Thomas Erl, Zaigham Mahmood and Ricardo Puttini, *Cloud Computing: Concepts, Technology & Architecture*, Prentice Hall.
3. Jonathan Bartlett, *Cloud Native Applications with Docker and Kubernetes*, Apress, 2022.

E-Resources:
1. https://learn.microsoft.com/azure/?utm_source=chatgpt.com\ 2. https://cloud.google.com/docs 3. https://docs.docker.com/?utm_source=chatgpt.com

DEPARTMENT OF COMPUTER ENGINEERING

Program: B.Tech.(Computer Engineering)									**Semester:** VI	
Course: MERN Stack Development									**Code:COPE602B**	
Teaching Scheme				**Evaluation Scheme**						
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**		**ETE**	**TW**	**OR**	**PR**	**Total**
03	02	1	04	40	60	-	-	-	25	125

Prerequisites:

1. Basic programming, HTML/CSS.
2. JavaScript fundamentals, and basic database concepts.

Course Objectives:

1. To Understand full-stack web application architecture using the MERN stack.
2. To Develop responsive front-end applications using React.
3. To Build RESTful back-end services using Node.js and Express.js.
4. To Design and manage databases using MongoDB.
5. To Integrate front-end, back-end, and database components into complete applications..

Course Outcomes:
On completion of the course, learners will be able to –

CO1	Understand architecture and technologies involved in modern full-stack web development.
CO2	Develop interactive and responsive front-end applications using React.
CO3	Design and implement RESTful APIs using Node.js and Express.js.
CO4	Design, query, and integrate MongoDB databases with MERN application
CO5	Develop secure, authenticated, database-driven full-stack applications.
CO6	Test, deploy, document, and maintain production-oriented MERN applications.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	Web Development and JavaScript Foundations Introduction to full-stack web development HTML5: semantic elements, forms, tables, multimedia, CSS3: selectors, box model, Flexbox, Grid, responsive design,JavaScript fundamentals,Variables, data types, operators, functions,Arrays, objects, DOM manipulation ,Events and event handling,Modules and asynchronous JavaScript Promises and async/await,Introduction to JSON and REST concepts Browser developer tools and debugging	4

2.	React.js and Front-End Development Introduction to React and SPA architecture,React project structure and development environment JSX,Components and component composition Props and state,Event handling,Conditional rendering and lists Forms and controlled components,React Hooks: useState, useEffect, useRef,Component lifecycle concepts,Routing using React Router Context API,State management concepts,API communication using Fetch/Axios,Loading, error, and empty states,Responsive UI development	5
3.	Node.js and Express.js Introduction to Node.js and server-side JavaScript,Node.js runtime and architecture,npm and package management,Modules and file-system operations,Event-driven and asynchronous programming Creating HTTP servers,Introduction to Express.js,Routing Middleware,Request and response objects,RESTful API development HTTP methods and status codes,Request parameters, query parameters, and request bodies,Error handling,Input validation,API testing using Postman/REST clients	5
4.	MongoDB and Database Integration Introduction to NoSQL databases,MongoDB architecture and terminology Databases, collections, and documents,BSON and MongoDB data types CRUD operations,MongoDB queries and operators,Sorting, limiting, and pagination,Indexing,Data modeling and schema design,Embedded documents vs. references,Aggregation framework,MongoDB Atlas,Introduction to Mongoose,Schemas and models,Validation Relationships and population,Connecting MongoDB with Node.js/Express	5
5.	Full-Stack MERN Application Development MERN stack architecture,Integrating React with Express and Node.js Connecting React to REST APIs,CRUD operations in a MERN application,Authentication and authorization concepts,User registration and login,Password hashing,JSON Web Tokens (JWT),Protected routes Role-based authorization,Cookies, sessions, and token-based authentication,Form validation,File/image upload concepts,Search, filtering, sorting, and pagination	5
6.	Full-Stack MERN Application Development MERN stack architecture,Integrating React with Express and Node.js Connecting React to REST APIs,CRUD operations in a MERN application,Authentication and authorization concepts,User registration and login,Password hashing,JSON Web Tokens (JWT),Protected routes Role-based authorization,Cookies, sessions, and token-based authentication,Form validation,File/image upload concepts,Search, filtering, sorting, and pagination **case study:** Use an AI coding assistant to generate a React component and manually	6

	review/improve the code.	
	TOTAL	**30**
Sr.No.	**List of Practical's**	
1.	Develop a MERN-based web application to manage student information. The system should allow administrators to add, view, update, delete, search, and filter student records. Include fields such as student ID, name, department, semester, email, and contact details.	
2.	Develop a MERN-based e-commerce application for selling books. The system should provide product listing, search, filtering, shopping cart, user authentication, order management, and admin product management.	
3.	Develop a MERN-based job portal connecting job seekers and recruiters. Job seekers should be able to create profiles, search and filter jobs, and submit applications, while recruiters can post jobs and manage applications.	
4.	Develop a web application for managing doctor appointments. Patients should be able to register, view doctors, check available slots, and book appointments. Doctors/admins should be able to manage schedules, appointments, and patient information.	
5.	Develop a MERN-based grievance management system where students can submit complaints, upload supporting documents, track complaint status, and receive updates. Administrators should be able to assign complaints, update their status, and generate reports.	
6	To do list web application using MERN stack.	
7	Notes Maker application using MERN stack.	

Text Books:

1. Learning React: Modern Patterns for Developing React Apps — Alex Banks & Eve Porcello, O'Reilly Media.
2. Node.js Design Patterns — Mario Casciaro & Luciano Mammino, Packt Publishing.
3. Express in Action: Writing, Building, and Testing Node.js Applications — Evan M. Hahn, Manning Publications.

Reference Books:

1. JavaScript: The Definitive Guide — David Flanagan, O'Reilly Media.
2. Eloquent JavaScript — Marijn Haverbeke, No Starch Press.
3. You Don't Know JS Yet — Kyle Simpson, O'Reilly Media.
4. Web Development with Node and Express — Ethan Brown, O'Reilly Media.

E-Resources:

1. React Documentation — https://react.dev/
2. Node.js Documentation — https://nodejs.org/docs/latest/api/
3. Express.js Documentation — https://expressjs.com/
4. MongoDB Documentation — https://www.mongodb.com/docs/

Back to Structure

DEPARTMENT OF COMPUTER ENGINEERING

Program: B.Tech.(Computer Engineering)										**Semester: VI**
Course: Agile and DevOps										**Code:COPE602C**
Teaching Scheme				**Evaluation Scheme**						
Lecture	**Practical**	**Tutorial**	**Credit**		**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	02	1	04		40	60	-		25	125

Prerequisites:

1. Fundamentals of Software Engineering and Software Development Life Cycle (SDLC).
2. Basic Programming skills (Python/Java) and understanding of Data Structures.
3. Basics of Operating Systems and Linux Commands.
4. Fundamentals of Version Control Systems (Git)and Networking Concepts.

Course Objectives:

1. To understand Agile philosophy, values, principles and various Agile frameworks for iterative software development.
2. To learn Scrum and Kanban methodologies for effectives print-based project management.
3. To understand DevOp sculture, principles and the Continuous Integration/Continuous Delivery (CI/CD) pipeline.
4. To gain hands-on expo sure to modern DevOp stools for version control, build, test, and deployment automation.
5. To understand containerization, orchestration and Infrastructure as Code(IaC) practices used in industry.
6. To develop skills in monitoring, logging and securing DevOps pipelines for real-world software projects.

Course Outcomes:

On completion of the course, learner will be able to-

CO1	Understand Agile philosophy, principles and various Agile frameworks such as Scrum, Kanban, XP and SAFe.
CO2	Apply Scrum roles, artifacts and ceremonies to plan, estimate and manage software projects.
CO3	Implement version control workflows and CI pipelines using Git, Jenkins and Git Hub Actions.
CO4	Build, containerize and orchestrate applications using Docker and Kubernetes.
CO5	ApplyInfrastructureasCodeandconfigurationmanagementusingTerraformandAnsible.
CO6	Monitor, log, analyze and secure DevOps pipelines using Prometheus, Grafana, ELK stack and DevSecOps practices.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**

1.	**Introduction to Agile and Scrum Framework:** Traditional SDLC (Waterfall) vs Agile approach, overview of Agile frameworks (Scrum, Kanban, XP, Lean, SAFe), Scrum roles (Product Owner, Scrum Master, Development Team), Scrum artifacts (Product Backlog, Sprint Backlog, Increment), Scrum ceremonies (Sprint Planning, Daily Stand-up, Sprint Review, Sprint Retrospective), User Stories and Acceptance Criteria, Estimation techniques (Story Points, Planning Poker, T-shirt sizing).	**05**
2.	**Agile Project Management Practices and Tools:** Kanban boards and WIP limits, Burndown and Burnup charts, Velocity and Release Planning, overview of Agile project tracking tools – Jira and Trello, and the relationship between Agile and DevOps culture.	**04**
3.	**Version Control and Continuous Integration:** Git fundamentals – repositories, branching, merging and conflict resolution, Git workflows and collaboration using GitHub/GitLab, Pull Requests and Code Review, Continuous Integration concepts, Jenkins architecture and Pipeline as Code (Jenkinsfile), and GitHub Actions workflows.	**05**
4.	**Containerization and Orchestration:** Virtualization vs Containerization, Docker architecture, Docker images and containers, Dockerfile, Docker Compose for multi-container applications, Docker Hub and private container registries, Introduction to Kubernetes architecture.	**05**
5.	**Continuous Delivery, Deployment and Infrastructure as Code:** Continuous Delivery vs Continuous Deployment, deployment strategies - Blue-Green Deployment, Infrastructure as Code (IaC) concepts, Terraform basics - providers, resources, state management, overview of cloud-native DevOps pipelines.	**06**
6.	**DevOps Monitoring, Security and Case Studies:** Monitoring and logging fundamentals, metrics collection and visualization using Prometheus and Grafana, centralized logging using ELK Stack, DevSecOps principles and shift-left security, code quality and security scanning using SonarQube and Trivy, Site Reliability Engineering (SRE) basics, and industry case studies of DevOps adoption.	**05**
	TOTAL	**30 Hrs.**

List of Practical's(All Practical's are Mandatory)

Practical Platform: Linux/Windows Environment with Git, Docker Desktop, Minikube/Kubernetes Cluster, Jenkins, and Cloud Free-Tier Account (AWS/Azure/GCP).

1.	Install and configure Git on a Linux/Windows system. Create a local repository and perform version control operations such as branching, merging, and rebasing. Study the Agile Manifesto, the Twelve Agile Principles, and the Scrum framework, and prepare a comparative report between the Agile and Waterfall software development models.
2.	Create a Scrum project using Jira or Trello by defining product vision, user stories, product backlog, sprint backlog, and sprint planning activities.
3.	Create and manage a Kanban board for a mini software project using Work-In-Progress

	(WIP) limits and project tracking metrics.
4.	Configure Jenkins to automate the build process of a sample application whenever source code changes are pushed to GitHub.
5.	Develop a Continuous Integration and Continuous Deployment pipeline using GitHub Actions for automated build and testing.
6.	Create a Docker image for a sample web application using Docker file and execute it inside a Docker container.
7.	Deploy a multi-container web application consisting of application and database services using Docker Compose.
I	**Mini Project: End-to-End DevOps Pipeline**: Design and implement a complete DevOps pipeline that automates source code management, continuous integration, continuous deployment, containerization, orchestration, monitoring, and logging for a sample web application.

Text Books:

1. Jez Humble and David Farley, Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, 1st Edition, 2010, Addison-Wesley Professional.
2. Len Bass, Ingo Weber and Liming Zhu, DevOps: A Software Architect's Perspective, 1st Edition, 2015, Addison-Wesley Professional.
3. Ken Schwaber, Agile Project Management with Scrum, 1st Edition, 2004, Microsoft Press.
4. Mike Cohn, Succeeding with Agile: Software Development Using Scrum, 1st Edition, 2009, Addison-Wesley Professional.

Reference Books:

1. Gene Kim, Jez Humble, Patrick Debois, John Willis and Nicole Forsgren, The DevOps Handbook, 2nd Edition, 2021, IT Revolution Press.
2. Nigel Poulton and Pushkar Joglekar, The Kubernetes Book: Navigate the World of Kubernetes with Expertise, 2nd Edition, 2024, Independently Published/Packt.
3. Sean P. Kane and Karl Matthias, Docker: Up & Running: Shipping Reliable Containers in Production, 3rd Edition, 2023, O'Reilly Media.
4. Yevgeniy Brikman, Terraform: Up & Running, 3rd Edition, 2022, O'Reilly Media.

E-Resources:

1. https://www.atlassian.com/agile
2. https://www.jenkins.io/doc/
3. https://docs.docker.com/
4. https://kubernetes.io/docs/home/
5. https://www.terraform.io/docs
6. https://www.coursera.org/professional-certificates/devops-and-software-engineering

https://www.redhat.com/en/topics/devops

Back to Structure

PROGRAM ELECTIVE - III

(Select any one course from the given Program Elective Courses)

DEPARTMENT OF COMPUTER ENGINEERING

Program: B.Tech.(Computer Engineering)										**Semester: VI**
Course: Embedded System and RTOS										**Code:COPE603A**
Teaching Scheme				**Evaluation Scheme**						
Lecture	**Practical**	**Tutorial**	**Credit**	**IE**	**MTE**	**ETE**	**TW**	**OR**	**PR**	**Total**
3	2	-	4	40	-	60	-	25	-	125

Prerequisites:
Digital Logic, Analog Circuits.

Course Objectives:

1. To understand the fundamentals of embedded systems and their role in modern technology.
2. To understand the implementation of the various embedded components using the embedded C program.
3. To understand the fundamentals of ARM-based systems, including architecture and its units like registers, debug interface, stack, MPU, Interrupts etc.
4. To Use the various instruction stereogram the ARM controller.
5. To Understand the embedded system's real-time operating system and its application in IoT

Course Outcomes: On completion of the course, learner will be able to–

CO1	To Understand and Analyze various types of Embedded systems & its fundamental concepts including their architecture, components, and operational principles.
CO2	To Understand the basic hardware components and their selection method based on the characteristics and attributes of an embedded system
CO3	To Describe &Apply the architectural features and instructions of 32-bit microcontroller ARM Cortex M3.
CO4	To Design &Apply the knowledge gained for Programming ARM CortexM3 for different applications.
CO5	To Analyze & evaluate the need of real time operating system for embedded system applications.
CO6	To Analyse & evaluate the need of real time operating system for embedded system applications.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to Embedded Systems:** Architecture of Embedded System, Design Methodology, Design Metrics, General Purpose Processor, System On chip. Embedded system design an development, Life-Cycle Models, Development tools.	07

	System design specifications Functional design, Architectural design, Prototyping	
2.	**ARM Architecture**: ARM Design Philosophy, Registers, PSR, Pipeline, Interrupts and Vector Table, Architecture Revision, ARM Processor Families ARM architectural support for operating System, Memory subsystem architecture, Designing a cache system, Memory allocation, Communication protocols.	07
3.	**ARM Programming Model:** Instruction Set: Data Processing Instructions, Branch, Load, Store Instructions, PSR Instructions, Conditional Instructions. Register Usage, Other Branch Instructions, Data Processing Instructions, Single-Register and Multi Register Load-Store Instructions, Stack, Software Interrupt Instructions Scheduling, Register Allocation, Conditional Execution and Loops.	07
4.	**Introduction to MOS Technology**: Basic MOS Transistor action: Enhancement and Depletion Modes. Basic electrical properties of MOS, Threshold voltage and Body Effect. Design of MOS inverters with different loads, Basic Logic Gates with CMOS: INVERTER, N AND, N OR, AOI and OAI gates. Transmission gate logic circuits, Bi-CMOS inverter	07
5.	**Combinational Logic:** Manchester, Carry select and Carry Skip adders, Crossbar and barrel shifters, Multiplexer. Sequential Logic: Design of Dynamic Register Element, Dynamic RAM Cell, Static RAM Cell. D flip flop using Transmission gates. NOR and NAND based ROM Memory Design.	07
6.	**Introduction to HDLs:** Basic Concepts of Verilog, Data Types, System Tasks and Compiler Directives. Behavioural Modelling: Structured Procedures, Procedural Assignments, Timing control, Conditional statements, Sequential and Parallel Blocks, Generate Blocks. Switch level Modelling. Tasks, Functions, Procedural Continuous Assignments, Design of Mealy and Moore state models using Verilog.	07
	TOTAL	**42**

List of Practical's :

1.	Define Embedded system and explain its architecture
2.	Study various tools used in development of embedded system
3.	Analyze various memory allocation algorithms
4.	Explain in detail ARM architecture
5.	Study instruction set required for developing ARM programming model
6.	Explain use of basic logic gates in designing a transmission circuit

7.	Distinguish between combinational and Sequential logic
8.	Explain in detail various types of of flip-flops
9.	Describe in detail concept of Verilog
10.	Case study Embedded system

Text Books:

1. Embedded Systems-Architecture Programming and Design–RajKamal,2nd ed.,2008,TMH.
2. ShibuKV,"Introduction to Embedded Systems",TataMcGrawHillEducation
3. AndrewNSloss,DominicSystemandChrisWright,"ARMSystemDevelopersGuide",Elsevier,Morgan Kaufman publisher, 1st Edition, 2008

Reference Books:

1. Embedded Microcomputer Systems, Real Time Interfacing–Jonathan W.Valvano–Brookes/ Cole, 1999, Thomas Learning.
2. "IntroductiontoVLSISystems:ALogic,CircuitandSystemPerspective",Ming-BOLin,CRC Press, 2011.

Back to Structure

DEPARTMENT OF COMPUTER ENGINEERING

Program: B.Tech.(Computer Engineering)							**Semester: VI**		
Course: Ethical Hacking & Penetration Testing							**Code:**COPE603B		
Teaching Scheme(Hrs./week)				**Evaluation Scheme(Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	02	-	04	40	60	-	25	-	125

Prerequisites:

1. Computer Networks
2. Operating Systems
3. Information and Cyber Security Fundamentals
4. Programming Fundamentals

Course Objectives:

1. To understand the fundamentals of ethical hacking, cyber threats.
2. To identify security vulnerabilities through reconnaissance, scanning, and vulnerability assessment techniques.
3. To analyse system, network, web application, wireless, and cloud security vulnerabilities.
4. To apply ethical hacking tools and penetration testing methodologies in a controlled environment.
5. To evaluate security risks and recommend appropriate security controls and mitigation strategies.
6. To prepare professional penetration testing reports by following legal, ethical, and industry best practices.

Course Outcomes: After completion of this course, students will be able to-

CO1	Explain the concepts of ethical hacking, cyber threats, and penetration testing methodologies.
CO2	Perform reconnaissance, scanning, and vulnerability assessment using standard security tools.
CO3	Analyse system, network, web application, wireless, and cloud security vulnerabilities.
CO4	Apply penetration testing methodologies and ethical hacking tools in a controlled Environment.
CO5	Evaluate identified security risks and recommend appropriate mitigation strategies.
CO6	Prepare professional penetration testing reports while following legal and ethical practices.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Foundations of Ethical Hacking and Cyber Security** Introduction to Ethical Hacking, Information Security Fundamentals, Types of Hackers, Cyber Threat Landscape, Types of Cyber Threats and Attacks, Hacking Methodology, Information Gathering, Foot printing and Reconnaissance.	06
2.	**Scanning and Vulnerability Assessment** Network Scanning, Port Scanning Techniques, Service Enumeration, Banner Grabbing, Vulnerability Assessment, Vulnerability Scanners, Risk Assessment	08
3.	**System and Network Security Assessment** Authentication and Password Security, Password Attack Concepts, Windows Security Features, Linux Security Features, Network Hardening, Firewall IDS/IPS, Security Auditing	08

4.	**Web Application Security Testing** WebApplicationArchitecture,SQLInjectionConcepts,Cross-SiteScripting (XSS), Authentication and Session Management, Secure Coding Practices	06
5.	**Malware, Wireless and Cloud Security** Introduction to Malware, Malware Types, Wireless Network Fundamentals, Wi-Fi Security Standards, Wireless Vulnerabilities, Cloud Security Basics, Cloud Threats, Mobile Security Basics, Security Best Practices	08
6.	**Penetration Testing Process and Emerging Trends** Penetration Testing Life Cycle, Reporting and Documentation, CVSS Basics, Honeypots, Threat Intelligence, AIinCyber Security, Emerging Trends in Ethical Hacking	06
	Total	42Hrs

List of Practical's: Perform any 8 practicals and 1 mini project

1.	Information Gathering and Reconnaissance: Perform passive reconnaissance and footprinting of a permitted/controlled target using tools such as WHOIS, DNS lookup, and basic OSINT techniques.
2.	Network and Port Scanning: Perform network discovery, host identification, and port scanning in a controlled lab environment using Nmap and analyze the results.
3.	Service Enumeration and Banner Grabbing: Identify running services, versions, and banners on authorized systems and analyze potential security risks.
4.	Vulnerability Assessment: Perform a vulnerability assessment of a deliberately vulnerable virtual machine using Nessus/OpenVAS and prepare a basic risk summary.
5.	System and Network Security Assessment: Examine Windows/Linux security configurations and perform basic security auditing, password-policy checks, firewall configuration, and system hardening.
6.	Web Application Security Testing: Identify and demonstrate common web vulnerabilities such as SQL Injection and XSS using a deliberately vulnerable application such as DVWA.
7.	Authentication and Session Security Testing: Analyze login mechanisms, password security, session management, and common authentication weaknesses in a controlled web application.
8.	Wireless and Network Security Assessment: Analyze Wi-Fi security configurations, identify insecure settings, and recommend appropriate security practices using a controlled wireless/lab environment.
9.	Penetration Testing and Security Reporting: Perform a basic penetration-testing exercise on an authorized lab system, classify findings using CVSS, and prepare a professional penetration-testing report.
I	Mini Project: Security Assessment and Penetration Testing of a Controlled Web/Network Environment

Text Books:

1. MichaelT.Simpson,KentBackmanandJamesCorley,Hands-OnEthicalHacking and Network Defense, Cengage Learning.
2. RafayBaloch,Ethical Hackingand PenetrationTestingGuide,CRC Press.

3. GeorgiaWeidman,PenetrationTesting:AHands-OnIntroductiontoHacking,No Starch Press.
4. WilliamEasttom,ComputerSecurityFundamentals, Pearson.
5. EC-Council,EthicalHackingEssentials.

Reference Books:

JonErickson,Hacking: TheArtof Exploitation.
KevinBeaver,HackingforDummies.
ChrisMcNab, Network SecurityAssessment.
OWASPWeb SecurityTestingGuide.
NISTSP800-115:TechnicalGuidetoInformationSecurityTestingandAssessment.

E-Resources:

1. **https://www.oreilly.com/library/view/penetration-testing-a/9781593275648/**
2. **https://www.routledge.com/Ethical-Hacking-and-Penetration-Testing-Guide/Baloch/p/book/9781138500457**
3. **https://owasp.org/www-project-web-security-testing-guide/**

Back to Structure

DEPARTMENT OF COMPUTER ENGINEERING

Program: B. Tech. (Computer Engineering)								**Semester:** VI	
Course: 5G & Next-Generation Networks								**Code:** COPE603C	
Teaching Scheme (Hrs/week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	02	-	04	40	60	-	-	25	125

Prerequisites:

1. Computer Networks and Internet fundamentals
2. Basic understanding of software development and version control

Course Objectives:

1. Understand the evolution, architecture, and key requirements of 5G networks.
2. Explain 5G New Radio, spectrum, massive MIMO, beamforming, and other communication technologies.
3. Understand 5G core network architecture, virtualization, SDN, NFV, and network slicing
4. Analyze 5G applications, IoT integration, edge computing, and security.
5. Understand the concepts and technologies of Next Generation Networks (NGN).
6. Explore emerging technologies and the evolution of 5G toward 6G networks.

Course Outcomes: After completion of this course, students will be able to -

CO1	Explain the fundamental concepts, service models, deployment models, architecture, virtualization, and containerization used in cloud computing.
CO2	Apply virtualization techniques, cloud infrastructure, containers, and cloud services to develop scalable cloud solutions.
CO3	Analyze cloud services, APIs, load balancing, auto-scaling, serverless computing, and cloud-native architectures using AWS, Azure, and GCP**.**
CO4	Analyze cloud-native application development, microservices, Kubernetes, DevOps, CI/CD, and 6R-based data migration strategies.
CO5	Evaluate cloud security, IAM, encryption, governance, compliance, SLA, cost optimization, disaster recovery, Zero Trust, and multi-cloud solutions.
CO6	Create AI-enabled cloud applications using LLMs, prompt engineering, RAG, vector databases, AI agents, tool/function calling, and API integration.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction and Evolution of 5G Networks:** Introduction to wireless and mobile communication, Evolution of mobile communication, Limitations of existing mobile networks, Need for 5G networks, 5G vision and requirements, High data rate and low latency, 5G use cases, Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communication (URLLC), Massive Machine-Type Communication (mMTC)	07
2.	**5G Architecture and New Radio:** 5G system architecture, 5G Radio Access Network, gNB, 5G Core Network,	07

	Control Plane and User Plane, Introduction to 5G New Radio, 5G frequency ranges, Sub-6 GHz and millimeter-wave communication, OFDM and 5G numerology, Massive MIMO, Beamforming and beam management, Cloud RAN and Open RAN.	
3.	**5G Core Network and Virtualization:** Introduction to 5G Core Network, Service-Based Architecture, AMF, SMF, UPF, UDM, AUSF, PCF, NRF, Registration and authentication, PDU session establishment, Software Defined Networking, Network Function Virtualization, Virtual Network Functions, Network slicing, eMBB, URLLC and mMTC network slices.	07
4.	**5G Applications, IoT, Edge Computing and Security** 5G and Internet of Things: Massive IoT, Industrial IoT, Machine-to-Machine Communication, 5G and Cloud Computing, Mobile/Multi-access Edge Computing, Edge Computing vs. Cloud Computing, 5G Applications, 5G Security Requirements, Privacy and Data Protection, Security Threats: DoS/DDoS, Eavesdropping and Spoofing, Security Challenges in Network Slicing.	07
5.	**Next Generation Networks: Architecture and Technologies** Introduction to Next Generation Networks (NGN): Need, Evolution, Characteristics and Challenges, NGN Functional Architecture, Key Development Areas: Terminal, Access, Backhaul, Core Transport and Service Creation, Network Control and Management, IP-Based and All-IP Networks, SDN and Network Virtualization in NGN, NGN Standardization: ITU, ETSI, 3GPP and NGMN	07
6.	**Beyond 5G and Emerging 6G Networks** Beyond 5G and 5G-Advanced, Introduction to 6G: Vision, Requirements and Key Features, AI-Native and Intelligent Networks, Terahertz Communication and Intelligent Reflecting Surfaces, Cell-Free Massive MIMO and Integrated Sensing and Communication, Non-Terrestrial Networks and Edge Intelligence, Emerging Applications: Extended Reality (XR), Digital Twins and Holographic Communication, Energy Efficiency.	07
	TOTAL	**42**
No	**Practical / Problem Statement**	
1	**5G Network Performance Analysis**: Develop a Python program to calculate and compare throughput, latency, packet loss, and bandwidth for different network scenarios representing 4G and 5G.	
2	**OFDM Signal Simulation:** Implement an OFDM transmitter and receiver using Python. Generate subcarriers, perform IFFT/FFT, add noise, and analyze the received signal.	
3	**MIMO Performance Analysis:** Simulate a basic 2×2 MIMO wireless communication system and analyze the effect of the number of antennas on data transmission performance.	
4	**5G Beamforming Simulation:** Implement a simple beamforming algorithm and analyze signal strength at different user locations with and without beamforming.	
5	**5G Network Slicing:** Design and simulate three virtual network slices for eMBB, URLLC, and mMTC. Allocate different bandwidth and latency requirements to each slice and analyze	

	their performance.
6	**SDN Network Implementation:** Create a simple Software Defined Network with multiple hosts and a switch. Configure flow rules through an SDN controller and demonstrate communication between hosts.
7	**Network Traffic Analysis:** Capture network packets using Wireshark and identify protocols, source/destination addresses, packet size, transmission time, and traffic characteristics.
8	**5G-Enabled IoT Data Processing:** Generate data from multiple simulated IoT devices and develop a Python program to collect, process, and visualize the data for a 5G-enabled IoT scenario.

Text Books:

1. Erik Dahlman, Stefan Parkvall and Johan Sköld, 5G NR: The Next Generation Wireless Access Technology, Academic Press.
2. Jonathan Rodriguez, Fundamentals of 5G Mobile Networks, Wiley.
3. Jingming Li Salina and Pascal Salina, Next Generation Networks: Perspectives and Potentials, John Wiley & Sons.

Reference Books:

1. Jonathan Rodriguez, 5G: An Introduction to the Next Generation Networks, Wiley.
2. W. Saad, M. Bennis and M. Chen, A Vision of 6G Wireless Systems, IEEE.
3. Steven Noble, Building Modern Networks: Create and Manage Cutting-Edge Networks and Services, Packt Publishing.

E-Resources:

1. 3GPP – 5G and Next Generation Networks
2. ITU – 5G and IMT-2020
3. NGMN Alliance – 5G and Future Networks
4. ETSI – 5G and Network Technologies
5. NPTEL – Wireless Communication and 5G Courses

Back to Structure

DEPARTMENT OF COMPUTER ENGINEERING

Program: B.Tech. Computer Engineering									**Semester:** VI
Course: Wireless Communication									**Code:COMD604**
Teaching Scheme:				**Evaluation Scheme**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03			04	40	60				100

Prerequisites: Digital and analog Communication, Signals & Systems, Electromagnetic Theory, Probability & Random Processes.

Course Objectives:

1. The objective of the course is to introduce the Concepts of basic wireless mobile communication systems
2. To learn and understand the basic principles of Telecommunications witching, traffic, and networks.
3. To learn and understand basic concepts of cellular system, wireless propagation, and the techniques used to maximize the capacity of a cellular network.

Course Outcomes: After completion of this course, students will able to-

CO1	Explain the evolution of wireless communication and identify the key milestones In the development to wireless technologies.
CO2	Explain the principles of radio wave propagation and the free-space propagation Model.
CO3	Apply handoff strategies and roaming concept tonsure seamless mobile Communication.
CO4	Compare the characteristics, advantages, limitations, and applications of different Multiple access techniques.
CO5	Compare the combining techniques— Selection Combining(SC),Equal Gain Combining (EGC), and Maximal Ratio Combining (MRC)—based on their performance and implementation complexity.
CO6	Compare the characteristics, advantages, limitations, and applications of various Emerging wireless communication technologies.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to Wireless Communication**: Evolution of wireless communication, Wireless communication systems and applications, Generations of mobile communication (1G–5G), Wireless spectrum and regulatory aspects, Challenges in wireless communication.	5
2.	**Radio Wave Propagation**: Free-space propagation model, Friis transmission equation, Reflection, diffraction, and scattering, Path loss models, Shadowing and log-normal fading, Large-scale Propagation models.	5
3.	**Cellular Communication Systems:** Cellular architecture, Cell splitting and sectoring, Frequency reuse, Handoff	5

	strategies, Roaming concepts, Capacity improvement techniques, Co-channel Interference, Signal attenuate on, Network congestion, Security and privacy concerns.	
4.	**Multiple Access Techniques:** Introduction, comparisons of multiple access strategies like TDMA , CDMA, FDMA, OFDM, CSMA Protocol, SC-FDMA, NOMA (Non-Orthogonal Multiple Access).	5
5.	**Diversity Techniques:** Diversity concepts, Time diversity, Frequency diversity, Space diversity, combining techniques: Selection Combining(SC),Equal Gain Combining(EGC),and Maximal Ratio Combining (MRC).	5
6.	**Recent Trends:** Introduction to Wi-Fi, Wi-MAX, ZigBee Networks, Software Defined Radio, UWB Radio, Wireless Ad hoc Network, Mobile Portability, MIMO, Security issues, and Challenges in a Wireless network.	5
	TOTAL	30

Text Books:

1. TheodoreS.Rappaport,"WirelessCommunication,",PrenticeHall
2. VijayGarg,"WirelessCommunicationsand Networking", Elsevier.
3. KamiloFeher, "Wirelessdigitalcommunication",PHI
4. WilliamC. Y.Lee,"MobileCommunicationsEngineeringGrawHillPublications.
5. RajPandya,"MobileandpersonalCommunicationsystemandservices", IEEEpress (PHI).
6. T.L.Singh,"WirelessCommunication",TMH

Reference Books:

1. UpenaDalal,—"WirelessCommunication"OxfordHigherEducation.
2. DavidTse,PramodViswanath—"FundamentalsofWirelessCommunication", CAMBRIDGE University Press.
3. Gary.J.Mullet,—"IntroductiontoWirelessTelecommunicationSystemsand Networks", DELMAR CENGAGE Learning
4. WilliamC.Y.Lee,"MobileCellularTelecommunications:AnalogandDigital Systems", Tata McGraw Hill Edition.
5. Vijay.K.Garg,"WirelessNetworkEvolution:2Gto3G",PearsonEducation

E-Resources:

1. Students may use SCILAB, MATLAB,NETSIM,NS2
2. NPTLVideos-http://www.digimat.in/nptel/courses/video/117102062/L01.html
3. MIT open course website
4. Virtual Labs-https://iitb.vlabs.co.in/

Back to Structure

DEPARTMENT OF COMPUTER ENGINEERING

Program: B.Tech.(Computer Engineering)										**Semester:** VI
Course: Full-Stack Flask Lab										**Code:COVS605**
Teaching Scheme				**Evaluation Scheme**						
Lecture	**Practical**	**Tutorial**	**Credit**		**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
	04	-	02		-	-	25	-	50	75

Prerequisites:

1) Basics of Python Programming
2) Object-Oriented Programming Concepts
3) HTML5andCSS3
4) JavaScript Fundamentals
5) SQL and Relational Databases
6) Basic Data Structures
7) Computer Networks and Internet Fundamentals
8) Operating System Basics
9) Basic Linux/Windows Command Line
10) Introduction to Web Technologies

Course Objectives:

After completing this laboratory, students will be able to:

CO1	Understand the architecture and workflow of Flask-based web applications.
CO2	Develop responsive front-end interfaces using HTML, CSS, Bootstrap, and JavaScript integrated with Flask.
CO3	Design and implement database-driven web applications using Flask and SQL databases.
CO4	Build secure full-stack applications with user authentication, authorization, session management, and REST APIs.
CO5	Deploy and test scalable Flask applications using modern development tools and version control.
CO6	Design complete real-world web applications by integrating front-end, back-end, databases, APIs, and cloud deployment.

Course Outcomes:

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

CO1	Develop dynamic web applications using Flask framework and Python.
CO2	Design responsive user interfaces integrated with Flask templates and static resources.
CO3	Implement CRUD operations using Flask with relational databases.

CO4	Develop secure authentication and authorization mechanisms in Flask applications.
CO5	Integrate RESTful APIs, third-party services, and cloud deployment techniques.
CO6	Design, develop, test, and deploy complete full-stack web applications for real-world problems.

Course Contents:

Unit	Description	Duration (Hrs.)
1.	**Flask Installation and First Dynamic Web Application:** Zeal College wants a welcome portal for first-year students. Develop a Flask application that displays dynamic content including student name, department, and current date using Flask routing and templates.	**04**
2.	**Student Registration Portal:** Develop a web application that allows students to register for technical workshops By accepting user inputs through HTML forms and displaying submitted information.	**04**
3.	**Employee Management System using SQLite:** An organization wants to maintain Employee records. Develop a Flask application to perform Create, Read, Update, and Delete (CRUD) operations using SQLite.	**04**
4.	**College Library Management System:** Design a Flask-based Library Management System that allows librarians to add, search, update, issue, and return books using database operations.	**04**
5.	**Online Feedback Management System:** Develop a web portal where students can submit course feedback. Administrators should be able to view, update, and analyze submitted feedback.	**04**
6.	**User Authentication System** Develop a secure login system for an institute portal with registration, login, logout, password hashing, and session management.	**04**
7.	**Hospital Appointment Booking System:** Develop a Hospital Appointment Management System where patients can register, book appointments, and doctors can manage appointment schedules.	**04**
8.	**E-Commerce Product Management System:** Develop a Flask application where an administrator can manage products, categories, inventory, and customers can browse products.	**04**
9.	**REST API for Student Information System:** Develop RESTful APIs for managing student records. The application should support GET, POST, PUT, DELETE requests and JSON responses.	**04**
10.	**Online Food Ordering System:** Develop a restaurant ordering portal where customers can browse menus, place orders, calculate bills, and receive order confirmations.	**04**
11.	**Flask Application Deployment using Docker and Cloud:** Containerize a Flask application using Docker and deploy it on a cloud platform with environment variables and production configuration.	**04**
	Smart Campus Management System: Design and develop a complete Full Stack Flask application for a Smart Campus.	

12.	The system should include: • Student Registration • Faculty Login • Attendance Management • Notice Board • Course Registration • Assignment Submission • Admin Dashboard • Reports • REST API Integration • Authentication • Responsive UI • Database Management • Cloud Deployment	**04**
	TOTAL	**48 Hrs.**

TextBooks:

1. MiguelGrinberg, **FlaskWeb Development: Developing Web Applications with Python**, 2nd Edition, O'Reilly Media, 2018.
2. EricMatthes,**Python Crash Course**, 3rdEdition, NoStarchPress, 2023.
3. ClausFuhrer,**Python and Flask for Beginners**,PacktPublishing.
4. CoreyM.Schafer,**Practical Flask Development**,IndependentlyPublished.

Reference Books:

1. MiguelGrinberg,**FlaskMega-Tutorial**.
2. Antonio Melé,**Django5ByExample** (for comparisonofPythonwebframeworks).
3. DavidBeazley&BrianJones,**PythonCookbook**, 3rdEdition, O'Reilly.
4. AlSweigart,**AutomatetheBoringStuffwithPython**,2ndEdition.

E-Resources:

1. FlaskDocumentation:https://flask.palletsprojects.com/
2. PythonDocumentation:https://docs.python.org/3/
3. Jinja2Documentation:https://jinja.palletsprojects.com/
4. SQLAlchemyDocumentation:https://docs.sqlalchemy.org/
5. BootstrapDocumentation:https://getbootstrap.com/
6. WTFormsDocumentation:https://wtforms.readthedocs.io/
7. Flask-LoginDocumentation: https://flask-login.readthedocs.io/
8. DockerDocumentation:https://docs.docker.com/

Back to Structure