Course Scheme and Detailed Syllabus of Minor in IT for 2024 Batch Onwards

For a minor degree in IT, students from other departments are required to complete18 credits by opting the courses as per the following.

Semester	Professional Core	Professional Elective
Semester 4	3 Credits	3 Credits
Semester 5	5 Credits	
Semester 6	4 Credits	-----
Semester 7	-----	3 Credits

FOURTH SEMESTER

S. No.	Course			Contact Hours				Credits
	Category	Course Code	Course Title	L	T	P	Total	
1	PCC	XXXXXXX	Software Engineering	3	0	0	3	3
2	PEC	XXXXXXX	Discipline Elective – 1	2	0	0	2	2
3	PEC	XXXXXXX	Discipline Elective – 1 Lab	0	0	2	2	1

FIFTH SEMESTER

S. No.	Course			Contact Hours				Credits
	Category	Course Code	Course Title	L	T	P	Total	
1	PCC	XXXXXXX	Operating Systems for Smart Technologies	3	1	0	4	4
2	PCC	XXXXXXX	Android & iOS Lab	0	0	2	2	1

SIXTH SEMESTER

S. No.	Course			Contact Hours				Credits
	Category	Course Code	Course Title	L	T	P	Total	
1	PCC	XXXXXXX	Full Stack and Open-Source Software Development	3	0	0	3	3
	PCC	XXXXXXX	Full Stack and Open-Source Software Development Lab	0	0	2	2	1
OR								
	PCC	XXXXXXX	Cloud Computing	3	0	0	3	3
	PCC	XXXXXXX	Cloud Computing Lab: AWS and Azure	0	0	2	2	1

SEVENTH SEMESTER

S. No.	Course			Contact Hours				Credits
	Category	Course Code	Course Title	L	T	P	Total	
1	PEC	XXXXXXX	Discipline Elective – 6	3	0	0	3	3

Bucket Wise Tentative List of Discipline Electives (To be updated time to time)

Discipline Elective – 1 and Discipline Elective – 1 Lab (offered in 4th Semester)

S. No.	Course Code	Course Title	L	T	P	Total	Credits
1		Data Analytics using R and Python	2	0	0	2	2
2		Data Analytics using R and Python Lab	0	0	2	2	1
3		Mobile Application Development	2	0	0	2	2
4		Mobile Application Development Lab	0	0	2	2	1
5		Fundamentals of Smart Systems and IoT	2	0	0	2	2
6		Fundamentals of Smart Systems and IoT Lab	0	0	2	2	1

Discipline Elective – 6 (offered in 7th Semester)

S. No.	Course Code	Course Title	L	T	P	Total	Credits
1		Machine Learning and Natural Language Processing	3	0	0	3	3
2		Ethical Hacking & Prevention	3	0	0	3	3
3		Introduction to Large Scale Database Systems	3	0	0	3	3
4		Introduction to DevOps	3	0	0	3	3
5		Industrial Automation and IOT	3	0	0	3	3

Software Engineering

COURSE CODE: 25B11CI414

COURSE CREDITS: 3

CORE/ELECTIVE: Core

L-T-P: 3-0-0

Pre-requisite: None

Course Objectives:

1. Provide students with a systematic understanding of software engineering principles and software process models used in the development of large-scale software systems.
2. Enable students to identify, analyze, and document functional and non-functional requirements and prepare a well-structured Software Requirement Specification (SRS).
3. Develop the ability to apply software design principles such as modularity, cohesion, and coupling, and to model software systems using UML diagrams.
4. Equip students with knowledge of software construction practices, coding standards, metrics, and software testing techniques to ensure software quality.
5. Encourage students to evaluate and improve software systems based on quality attributes, performance metrics, and testing outcomes, preparing them for professional software development roles.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Understand software engineering principles and various process models for project development.	Remembering
CO-2	Outline the functional and non-functional requirements of a software project for creating Software Requirement Specifications (SRS).	Understand
CO-3	Apply software design principles (modularity, cohesion, coupling) and create UML models.	Apply
CO-4	Analyze the performance of software system using different testing methodologies.	Analyze
CO-5	Evaluate the software in terms of general software quality attributes, code structure and try to optimize the overall software system.	Evaluate

Course Contents:

Unit	Contents	Lectures required
1	***Introduction to Software Engineering:*** Introduction to Software Engineering Principles, Software Process Models (Build and Fix Model, Waterfall Model, Incremental Process Model, Evolutionary- Prototype and Spiral Models, Agile Models (tools	7

	study). Project Planning, Project Scheduling: Network Diagram, Gant Chart, CPM and PERT.	
2	***Requirement Engineering:*** Types of Requirement, Requirement Elicitation, Analysis, Specification, SRS, Requirement Verification and Validation.	**4**
3	***Software Design:*** Use Case Diagram, State Diagram, Activity Diagram, Class Diagram, Sequence Diagram, Collaboration Diagram, Deployment Diagram, Component Diagram and Package Diagram, Design Modularity: Coupling Cohesion.	**7**
4	***Software Construction:*** Coding Standards and Guidelines, Code Checklist, Code Reviews, Code Refactoring, Code Optimization. Design Pattern, Modern Programming Environments (Code Search, Programming using Library Components and their APIs), Program Comprehension; Program Correctness, Defensive Programming.	**8**
5	***Software Metrics:*** Size-Oriented Metric, Function-Oriented Metric, Halstead's Software Metric, Information Flow Metric, Object-Oriented Metric, Class-Oriented Metric, COCOMO Model.	**7**
6	***Software Testing:*** White-Box Testing, Basis Path Testing, Control Structure Testing: Condition Testing, Data Flow Testing, Loop Testing, Black-Box Testing: Equivalence Class Partitioning, Boundary Value Analysis, Decision Table Testing, Cause Effect Graphing, Mutation Testing and Regression Testing, Formal Methods.	**9**
Total lectures		**42**

Project-Based Learning: Each student works on a different case study in Assignments. They utilize the concepts taught in lecture and develop project in a group of 3-4.

The course emphasized on the skill development for employability in software industry by engaging students on Software Development methodologies. Various activities are carried out to enhance the student's software development skills. Some of them are study of various software process models and their applicability, progress tracking, size estimation techniques, software testing strategies, etc.

Suggested Text Book(s):

1. Roger S. Pressman, "Software Engineering: A Practitioner's Approach", Fifth Edition-TMH International.
2. Sommerville, "Software Engineering", Seventh Edition - Addison Wesley.

Suggested Reference Book(s):

1. Grady Booch, James Rumbaugh, Ivar Jacobson, The Unified Modeling Language User Guide, Addison Wesley, Reading, Massachusetts, May 2005
2. Richard Thayer, "Software Engineering Project Management", Second Edition -Wiley-IEEE Computer Society Press.

3. B. Bezier, “Software Testing Techniques”, Second Edition- International Thomson Computer Press.
4. Pankaj Jalote, “An Integrated Approach to Software Engineering” Third addition, Springer Press

Other useful resource(s):

1.Link to NPTEL course contents:https://nptel.ac.in/courses/106105182

EvaluationScheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	2	2	2	1	1	1	2	2	2	**1.73**
CO-2	2	3	3	3	3	1	1	1	2	2	1	**2.00**
CO-3	2	2	2	2	3	1	1	1	2	2	1	**1.73**
CO-4	2	3	3	3	2	1	1	1	2	3	2	**2.09**
CO-5	1	2	2	1	3	2	1	1	1	1	1	**1.45**
Average	**1.8**	**2.4**	**2.4**	**2.2**	**2.6**	**1.2**	**1.0**	**1.0**	**1.8**	**2.0**	**1.4**	

Data Analytics Using R and Python

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: Elective

L-T-P: 2-0-0

Pre-requisite: Basic understanding of mathematics, including elementary statistics (mean, median, probability)

Course Objectives:

1. Gain a clear understanding of the basic concepts of data analytics and how data is used to solve real-world problems.
2. Learn to use R and Python for writing programs to clean, organize, and analyze data.
3. Develop skills to explore datasets and apply basic statistical techniques to draw meaningful conclusions.
4. Create clear and informative visualizations to represent data using popular R and Python libraries.
5. Work confidently with essential data analytics libraries such as NumPy, pandas, and tidyverse.
6. Apply analytical and predictive techniques to solve practical problems using real datasets.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the fundamental concepts of data analytics.	Understand
CO-2	Demonstrate the concepts of R & Python for data analytics.	Understand
CO-3	Apply advanced methods and their quantitative analysis for real-world problems.	Apply
CO-4	Apply statistical methods for hypothesis testing and inference problems.	Apply
CO-5	Analyze, visualize and interpret the results for useful insights.	Analyze

Course Contents:

Unit	Contents	Lectures required
1	Data Definitions and Analytical Programming Techniques: Introduction to Data Analytics, Elements, Variables, and Data categorization, Levels of Measurement, Introduction to analytical programming languages R & Python, and Installing Software & Setting up, Lists & Dictionaries, Functions & Packages, Data frame, Import and Export data, Data Preprocessing.	**10**
2	Parametric &Non Parametric Tests: Hypothesis Testing, Assumption Testing, T-Test, Power Analysis, ANOVA, Fitting ANOVA Model in Python & R, Wilcoxon Tests, Mann-Whitney U Test	**6**

3	Correlation & Association Analysis: Pearson Correlation, Spearman Correlation, Kendall Tau Correlation, Affinity Analysis & Market Basket Analysis APriori Algorithm, Association Rules, Frequent Pattern Analysis	**7**
4	Data Analysis Techniques: Introduction to Machine Learning, Applications of ML Library in R & Python for Supervised & Unsupervised Learning, Basic Neural Network, Transfer Function Models, Multivariate Time Series Analysis.	**5**
Total lectures		**28**

Suggested Text Book(s):

1. Navlani, A., Fandango, A. and Idris, I. (2021). Python Data Analysis – Third Ed. Packt Publishing Ltd.
2. Jake vanderPlas, Python Data Science Handbook – Essential Tools for Working with Data, O'Really
3. Media, 2022
4. David J. Pine, Introduction to Python for Science and Engineering, CRC Press, 2019.
5. Manoj Kumar Srivastava and Namita Srivastava, Statistical Inference – Testing of Hypotheses, PHI, 2014.
6. Kabacoff, Robert I. R in action: data analysis and graphics with R. Simon and Schuster, 2015.
7. Haider, M. (2015). Getting Started with Data Science: Making Sense of Data with Analytics. IBM Press.

Suggested Reference Book(s):

1. Doing Data Science, Straight Talk from The Frontline, Cathy O'Neil and Rachel Schutt, O'Reilly (2014).
2. Robert Johansson, Numerical Python – Scientific Computing and Data Science Applications with NumPy, SciPy and Matplotlib, Apress, 2019
3. Robert Sedgewick, Kevin Wayne, Robert Dondero, Introduction to Programming in Python: An Inter-disciplinary Approach, Pearson India Education Services Pvt. Ltd., 2016
4. Nelli, F., Python Data Analytics: with Pandas, NumPy and Matplotlib, Apress, 2018.
5. Wickham, H., &Grolemund, G. (2016). R for data science: import, tidy, transform, visualize, and model data. " O'Reilly Media, Inc.".

Other useful resource(s):

1. Link to NPTEL course contents:https://nptel.ac.in/courses/106107220

EvaluationScheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2

3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	2	2	2	1	1	1	2	2	2	**1.73**
CO-2	2	3	3	3	3	1	1	1	2	2	1	**2**
CO-3	2	2	2	2	3	1	1	1	2	2	1	**1.73**
CO-4	2	3	3	3	2	1	1	1	2	3	2	**2.09**
CO-5	1	2	2	1	3	2	1	1	1	1	1	**1.45**
Average	**1.8**	**2.4**	**2.4**	**2.2**	**2.6**	**1.2**	**1**	**1**	**1.8**	**2**	**1.4**	

Data Analytics using R and Python Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Programming Language Fundamentals

Course Objectives:

1. An understanding of fundamental concepts of data analytics and its importance in real-world applications.
2. Proficiency with R and Python programming languages for data analysis.
3. Skills with data collection, cleaning, preprocessing, and transformation techniques.
4. Experience with exploratory data analysis and effective data visualization methods.
5. Knowledge with basic statistical methods and machine learning algorithms used in analytics.
6. Ability with interpreting analytical results for data-driven decision making.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Apply fundamental concepts of data analytics to understand and analyze real-world datasets.	Apply
CO2	Use R and Python programming languages to perform data manipulation, cleaning, and preprocessing tasks.	Apply
CO3	Perform exploratory data analysis and data visualization to extract meaningful insights from data.	Apply
CO4	Implement basic statistical techniques and machine learning algorithms for data analysis and prediction.	Apply

List of Experiments

S. No	Description	Hours
1	Program to install, import, and use basic R and Python libraries (NumPy, Pandas, Matplotlib, ggplot2). Program to create and manipulate vectors, lists, matrices, and data frames in R.	**2**
2	Program to perform basic data types and control structures in Python. Program to read data from CSV/Excel files and display summary statistics.	**2**
3	Program to handle missing values (NA/NULL) using mean, median, or deletion methods.	**2**
4	Program to compute descriptive statistics (mean, median, mode, variance, SD). Program to perform data normalization and standardization.	**2**

5	Program to generate data visualizations: • Bar chart • Histogram • Box plot • Scatter plot	**2**
6	Program to detect and remove outliers using IQR/Z-score. Program to perform correlation and covariance analysis.	**2**
7	Program to filter, sort, and group data using Pandas and dplyr. Program to perform data aggregation and pivot operations.	**2**
8	Program to merge and join multiple datasets.	**2**
9	Program to apply string manipulation and date-time operations.	**2**
10	Program to perform feature selection and feature scaling.	**2**
11	Program to perform probability distribution analysis.	**2**
12	Program to implement hypothesis testing (t-test, z-test, chi-square test).	**2**
13	Project	**2**
14	Project	**2**
Total Lab hours		**28**

Suggested/Resources:

1. Wes McKinney, Python for Data Analysis, O'Reilly Media.
2. Hadley Wickham & Garrett Grolemund, R for Data Science, O'Reilly Media.
3. Anil Maheshwari, Data Analytics Made Accessible, McGraw Hill.
4. Gareth James et al., An Introduction to Statistical Learning, Springer.
5. Link to topics related to course:
   i. https://www.bigbookofr.com/
   ii. https://rstudio-education.github.io/hopr/

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects, etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	3	1	1	1	1	1	2	**1.73**
CO-2	3	3	3	3	3	2	1	2	2	1	2	**2.27**
CO-3	3	3	2	3	3	2	1	2	2	1	2	**2.18**
CO-4	3	3	3	3	3	2	1	2	2	1	2	**2.27**
Average	**3**	**2.75**	**2.5**	**2.75**	**3**	**1.75**	**1**	**1.75**	**1.75**	**1**	**2**	

Mobile Application Development

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: Basic Programming (C/C++/Java), OOP Concepts

Course Objectives:

1. To understand mobile application development concepts, platforms, and architecture.
2. To develop basic Android applications using activities, intents, and UI components.
3. To design user-friendly interfaces using layouts and widgets.
4. To implement event-driven programming in mobile applications.
5. To integrate database and device features in mobile applications.
6. Understand basic mobile application security practices, including permissions and secure data handling.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the mobile application development lifecycle, Android architecture, and core components	Understand
CO-2	Apply concepts of activities, intents, and event handling to develop simple apps	Apply
CO-3	Design user interfaces using layouts, widgets, and navigation components	Create
CO-4	Analyze and implement advanced UI components, data handling techniques, and basic application security practices	Analyze
CO-5	Develop mobile applications integrating databases, device services, and secure data handling mechanisms	Create

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Mobile App Development: Mobile platforms overview (Android), app development lifecycle, Android architecture, setting up development environment (Android Studio, SDK), project structure, Activities, Activity lifecycle, Intents (explicit & implicit), manifest file, passing data between activities	**5**
2	Event Handling and Interaction: Event-driven programming, handling button clicks, listeners (onClick, onLongClick), input	**4**

	handling, simple app development using event listeners	
3	User Interface Design – I: Views and ViewGroups, Layouts (Linear, Relative, Constraint), TextView, EditText, ImageView, ListView, GridView, Spinner, XML-based UI design, Adapter classes	**6**
4	User Interface Design – II: Advanced UI components: RadioButton, CheckBox, ToggleButton, ProgressBar, Menus, Navigation (Tabs, Navigation Drawer basics), Date & Time Picker, handling gestures (swipe, pinch, zoom)	**5**
5	Advanced UI and Device Features: Customizing ListView, GridView, RecyclerView (intro), filtering data, sending SMS/email (conceptual), introduction to location-based services and Google Maps API	**4**
6	Mobile Database (SQLite): Introduction to SQLite, database creation, CRUD operations, content values, simple app with database integration, data persistence. Application security fundamentals: Android permissions model, runtime permissions, secure storage of user data, data privacy principles, secure coding practices.	**4**
Total lectures		**28**

Suggested Text Book(s):

1. Wei-Meng Lee, Beginning Android Programming, Wiley
2. Bill Phillips et al., Android Programming: The Big Nerd Ranch Guide, Big Nerd Ranch

Suggested Reference Book(s):

1. Reto Meier, Professional Android Development, Wiley
2. Jerome (J.F.) DiMarzio, Android: A Programmer's Guide, McGraw-Hill

Other useful resource(s):

1. Android Developer Documentation: https://developer.android.com
2. NPTEL Courses on Mobile Computing / Android Development: https://nptel.ac.in/courses/106106147

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered up to T-1
2	T-2	25	1.5 Hours	Syllabus covered up to T-2
3	T-3	35	2 Hours	Entire Syllabus

4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	2	1	1	1	2	1	2	**1.5**
CO-2	3	3	2	2	3	1	1	1	2	1	2	**1.9**
CO-3	3	2	3	2	3	1	1	2	3	1	2	**2.0**
CO-4	2	3	2	3	3	1	2	1	2	2	2	**2.1**
CO-5	3	2	3	2	3	1	2	2	3	2	3	**2.4**
Average	**2.8**	**2.4**	**2.2**	**2**	**2.8**	**1**	**1.4**	**1.4**	**2.4**	**1.4**	**2.2**	

Mobile Application Development Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Basic Programming, OOP Concepts

Course Objectives:

1. To provide hands-on experience in developing Android applications.
2. To implement activities, intents, and event-driven programming.
3. To design interactive user interfaces using layouts and widgets.
4. To develop applications using mobile databases (SQLite).
5. To integrate device features and services into mobile applications.
6. Implement basic mobile application security practices, including permission handling and secure data storage.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Develop basic Android applications using activities and intents	Apply
CO-2	Implement event handling and user interaction in mobile apps	Apply
CO-3	Design user interfaces using layouts, views, and adapters	Create
CO-4	Develop applications with advanced UI components, device features, and secure handling of permissions	Apply
CO-5	Implement database-driven applications using SQLite and basic secure data handling practices	Apply

List of Experiments

S. No	Description	Hours
1	Installation and setup of Android Studio, SDK, emulator; creating the first Android application (Hello World)	**2**
2	Study of Android project structure, Activity lifecycle demonstration using a sample app	**2**
3	Developing an application using Intents (explicit and implicit) and passing data between activities	**2**
4	Event handling: Button clicks, listeners (onClick), simple calculator application	**2**
5	Designing UI using Layouts (Linear, Relative, Constraint) with XML	**2**
6	Implementing UI elements: TextView, EditText, and ImageView with user input handling	**2**

7	Implementing ListView/GridView using Adapter classes	**2**
8	Creating an application using a Spinner and handling item selection events	**2**
9	Implementing advanced UI controls: RadioButton, CheckBox, ToggleButton, ProgressBar	**2**
10	Working with Menus and Navigation (Options menu / Context menu)	**2**
11	Designing an app with DatePicker and TimePicker	**2**
12	Customizing ListView/introduction to RecyclerView	**2**
13	SQLite database: Create database, insert and retrieve data (simple CRUD app).Implement basic secure storage practices (e.g., avoiding hardcoded data, using internal storage/SharedPreferences securely)	**2**
14	Mini Application: Develop a small Android app integrating UI + database (e.g., student record/notes app). Include runtime permission handling (e.g., storage/location) and basic secure data handling	**2**
Total Lab hours		**28**

Suggested/Resources:

1. Wei-Meng Lee, Beginning Android Programming, Wiley
2. Bill Phillips et al., Android Programming: The Big Nerd Ranch Guide

Other useful resource(s):

1. Android Developer Documentation: https://developer.android.com

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects, etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	3	1	1	1	2	1	2	**1.7**
CO-2	3	2	2	2	3	1	1	1	2	1	2	**1.8**
CO-3	3	2	3	2	3	1	1	2	3	1	2	**2.1**
CO-4	3	2	3	2	3	1	2	2	2	1	2	**2.1**
CO-5	3	3	3	2	3	1	2	2	2	2	3	**2.4**
Average	**2.5**	**2.2**	**2.6**	**1.8**	**3**	**1**	**1.4**	**1.6**	**2.2**	**1.2**	**2.2**	

Fundamentals of Smart System and IoT

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: Elective

L-T-P: 2-0-0

Pre-requisite: Basic Programming Knowledge, Basic Digital Electronics, Basic Computer Networking

Course Objectives:

1. Develop a foundational understanding of the Internet of Things (IoT), including smart sensors, communication paradigms, and their role in smart applications such as smart homes, smart cities, healthcare, and industrial automation.
2. Explain the architecture, communication protocols, and core components of IoT systems, along with the operational principles of Arduino microcontroller-based platforms.
3. Apply sensor interfacing techniques and Arduino programming skills to design, implement, and control basic IoT-enabled smart devices.
4. Analyze the performance characteristics, limitations, and suitability of different sensors and communication modules for various IoT-based smart system applications.
5. Demonstrate the design and working of real-world IoT applications in domains such as smart homes, healthcare, agriculture, and intelligent transportation systems.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Define the basic concepts of the Internet of Things (IoT), smart sensors, and their applications in various smart systems.	Understand
CO-2	Explain the architecture, protocols, and components of IoT systems and describe the working of Arduino microcontroller-based setups.	Understand
CO-3	Apply the knowledge of different sensors and Arduino programming to interface and control smart devices.	Apply
CO-4	Analyze the characteristics and performance of various sensors and communication modules used in IoT-based smart systems.	Apply
CO-5	Illustrate the design and operation of IoT applications in fields such as smart homes, healthcare, agriculture, and transportation.	Apply

Course Contents:

Unit	Contents	Lectures required

1	Introduction to Smart Sensors and IoT: Introduction to IoT: Definition, evolution, enabling technologies. Smart Sensors: concept, role in IoT. Measuring and monitoring environmental conditions. Types of smart systems and application domains (home, healthcare, agriculture, transport)	**3**
2	IoT Architecture, Protocols, and Arduino Microcontroller IoT system architecture (device, gateway, cloud, application layers). Bread board and its internal connections. IoT protocol stack: Physical, MAC, Network, Transport, Application layers. Overview of common IoT protocols. Arduino overview: board components, microcontroller basics. Arduino development environment and basic operation.	**5**
3	Sensors and Their Characteristics: Sensor classification and working principles. Capacitive, resistive, SAW sensors. Sensors for temperature, pressure, humidity, gas. Static characteristics: sensitivity, accuracy, resolution. Dynamic characteristics: response time, bandwidth. Smart sensor architecture and interfacing challenges.	**5**
4	Design of Smart Systems: Embedded system design using Arduino. LED, tricolor LED, LCD, seven-segment display concepts. Push button interfacing. Pulse Width Modulation (PWM). Environmental sensors (DHT11).	**5**
5	Wireless Smart Systems: ESP8266-01 Wi-Fi module. IoT system components for cloud connectivity. ThingSpeak platform overview. MQTT protocol fundamentals. Introduction to low-level programming (AVR-GCC concepts).	**5**
6	IoT Applications: Smart Home systems: components, benefits, challenges. Smart Cities and Intelligent Transportation Systems. Smart Healthcare and wearable systems. Smart agriculture and precision farming. Smart factories and Industry 4.0. Societal, ethical, and privacy challenges	**5**
Total lectures		**28**

Suggested Text Book(s):

1. Advances in Modern Sensors; Physics, design, simulation and applications by Sinha, G, R, IOP (Institute of Physics Publishing), 2020
2. Internet of Things: Architecture and Design Principles, Raj Kamal, McGrawHill. 2017
3. JanHo"ller,VlasiosTsiatsis,CatherineMulligan,Stamatis,Karnouskos,StefanAvesand,David Boyle, "From Machine-to-Machine to the Internet of Things -Introduction to a New Age of Intelligence",Elsevier,2014.

Suggested Reference Book(s):

1. ArshdeepBahga, Vijay Madisetti, "Internet of Things – A hands-on approach", Universities Press, 2015
2. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton, Jerome Henry, "IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things", CISCO Press, 2017.

Other useful resource(s):

1. https://spoken-tutorial.org/tutorial-search/?search_foss=Arduino&search_language=English
2. https://www.ibm.com/smarterplanet/us/en/?ca=v_smarterplanet

EvaluationScheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	2	2	2	1	1	1	2	2	2	**1.73**
CO-2	2	3	3	3	3	1	1	1	2	2	1	**2**
CO-3	2	2	2	2	3	1	1	1	2	2	1	**1.73**
CO-4	2	3	3	3	2	1	1	1	2	3	2	**2.09**
CO-5	1	2	2	1	3	2	1	1	1	1	1	**1.45**
Average	**1.8**	**2.4**	**2.4**	**2.2**	**2.6**	**1.2**	**1**	**1**	**1.8**	**2**	**1.4**	

Fundamentals of Smart System and IoT Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Basic Programming Knowledge, Basic Digital Electronics, Basic Computer Networking

Course Objectives:

1. Introduce students to the fundamental concepts of the Internet of Things (IoT) through hands-on interaction with embedded hardware, sensors, and actuators.
2. Enable students to acquire practical skills in interfacing sensors, displays, and actuators using microcontroller platforms such as Arduino and ESP32.
3. Familiarize students with basic networking concepts and cloud connectivity by implementing Wi-Fi–enabled data transmission and cloud-based data visualization.
4. Develop an understanding of edge computing concepts through Linux-based single-board computers by configuring Raspberry Pi for data acquisition and local storage.
5. Prepare students to design and analyze simple end-to-end IoT systems by integrating sensing, communication, data processing, and visualization components.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	To identify and explain basic components of IoT systems	Understand
CO2	To interface sensors, displays, and actuators using microcontroller platform	Apply
CO3	To implement wired and wireless IoT systems for sensor data transmission	Apply
CO4	To configure edge-based IoT systems using Raspberry Pi	Apply

List of Experiments

S. No	Description	Hours
1	Arduino IDE Setup and Digital I/O Test: Blink LED and Serial Monitor Output	**2**
2	Push Button and LED Control on Arduino: Toggle/Counter with Debouncing	**2**
3	Analog Input using Potentiometer/LDR: Sensor Value Reading and Mapping in Arduino	**2**
4	LCD Interfacing with Arduino: Display Sensor Readings (Potentiometer/LDR/DHT11)	**2**
5	PWM with Arduino: LED Brightness Control using Potentiometer (AnalogRead to AnalogWrite)	**2**

6	DHT11 Sensor Interfacing with Arduino: Temperature/Humidity Readings and Serial Logging	**2**
7	ESP8266 Wi-Fi Interfacing: Connect to Wi-Fi, Obtain IP, and Test Connectivity (Ping/Basic Web Request)	**2**
8	IoT Data Upload using ESP8266 and DHT11: Send Sensor Data to ThingSpeak using HTTP GET/POST	**2**
9	Cloud Dashboard on ThingSpeak: Visualize DHT11 Data, Field Configuration, and Update-Rate Analysis	**2**
10	Raspberry Pi Setup and GPIO with Python: Blink LED and Read Push Button (RPi.GPIO / gpiozero)	**2**
11	Raspberry Pi as IoT Gateway: Read DHT11/DS18B20 and Store Data in CSV/SQLite with Timestamp	**2**
12	Mini IoT Project (ESP8266/Raspberry Pi): Sensor Node and Cloud/Local Dashboard (Smart Home/Agriculture/Health)	**6**
Total Lab hours		**28**

Suggested/Resources:

1. Advances in Modern Sensors; Physics, design, simulation and applications by Sinha, G, R, IOP (Institute of Physics Publishing), 2020
2. Internet of Things: Architecture and Design Principles, Raj Kamal, McGrawHill. 2017
3. Jan Ho"ller, VlasiosTsiatsis, Catherine Mulligan, Stamatis, Karnouskos, Stefan Avesand, David Boyle, "From Machine-to-Machine to the Internet of Things -Introduction to a New Age of Intelligence", Elsevier, 2014.
4. ArshdeepBahga, Vijay Madisetti, "Internet of Things – A hands-on approach", Universities Press, 2015
5. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton, Jerome Henry, "IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things", CISCO Press, 2017.
6. Link to topics related to course:

iii. https://spoken-tutorial.org/tutorial-search/?search_foss=Arduino&search_language=English
iv. https://www.ibm.com/smarterplanet/us/en/?ca=v_smarterplanet
v. https://www.arduino.cc/en/Tutorial/HomePage
vi. https://www.raspberrypi.org/documentation/
vii. "InternetofThings:AHands-onApproach",byArshdeepBahgaandVijayMadisetti

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	3	1	1	1	1	1	2	**1.73**
CO-2	3	3	3	3	3	2	1	2	2	1	2	**2.27**
CO-3	3	3	2	3	3	2	1	2	2	1	2	**2.18**
CO-4	3	3	3	3	3	2	1	2	2	1	2	**2.27**
Average	**3**	**2.75**	**2.5**	**2.75**	**3**	**1.75**	**1**	**1.75**	**1.75**	**1**	**2**	

Operating Systems for Smart Technologies

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: Core

L-T-P: 3-1-0

Pre-requisite: Basics of Computers and Programming languages

Course Objectives:

1. To introduce the design principles and functionalities of modern operating systems, with emphasis on process management, scheduling, and inter-process communication.
2. To explain concepts of memory management, including paging, virtual memory, and page replacement algorithms.
3. To familiarize students with the architecture and features of popular mobile and smart device operating systems like Android, iOS, and Linux.
4. Have good understanding of file system structures and data management techniques across Linux, Windows, Android, and iOS platforms.
5. To provide insights into mobile OS forensics, data acquisition techniques, and security aspects of smart device platforms.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Understand core OS design concepts including spooling, buffering, and scheduling	Remember, Understand
CO-2	Analyze memory management and page replacement techniques	Analyze, Evaluate
CO-3	Compare features and architectures of Android, iOS, Linux, and mobile OSs	Analyze
CO-4	Understand file management across operating systems and perform basic forensic tasks	Understand, Analyze
CO-5	Analyze and recover data from mobile platforms and compare mobile OS security models	Apply

Course Contents:

Unit	Contents	Lectures required

1	Operating System Basics: Introduction to OS design, Spooling and Buffering, Single and multiprocessor scheduling, Process operations, Threads, Inter-process Communication (IPC), Precedence graphs, Critical section problem, Semaphores	**6**
2	Deadlock and Memory Management: Deadlock – characterization, detection, prevention, recovery; Concepts of memory management, Paging, Segmentation Virtual Memory and Cache: Concepts of virtual memory, Cache Memory Organization, Page replacement algorithms	**4** **4**
3	Android OS and File Systems: Introduction to Android OS, Features of Android, Architecture and Applications, Android Emulator, File Management in Linux, Windows, Android and iOS	**6**
4	iOS System and File Architecture: iOS systems overview, Design architecture of iOS, File system architecture of iOS, Difference between Android and iOS	**6**
5	Linux and Other Smart OS: Introduction to Linux OS, Unix and Shell, Mobile OS, IoT OS, Handheld OS, Monolithic and Microkernel architectures on Android, iOS, and Linux	**6**
6	Data Acquisition and Recovery: Data acquisition from iOS, Android, and Windows OS; Acquiring data from backups (iOS, Linux, Windows); iOS data analysis and recovery Mobile OS Forensics and Open OS: Examination of iOS, Android, and Windows phone forensics; Overview of Android and iOS security; Introduction to Open OS and applications	**5** **5**
Total lectures		**42**

Suggested Text Book(s):

1. Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Operating System Concepts, 10th Edition, Wiley, 2018. ISBN: 9781119456339
2. Andrew S. Tanenbaum, Herbert Bos, Modern Operating Systems, 4th Edition, Pearson, 2015. ISBN: 9780133591620
3. Neil Smyth, Android Studio Development Essentials – Android 6 Edition, Payload Media, 2016. ISBN: 9781530161678
4. Jesse Varsalone, Mac OS X, iPod, and iPhone Forensic Analysis DVD Toolkit, Syngress, 2008. ISBN: 9781597492973
5. Brian Carrier, File System Forensic Analysis, Addison-Wesley, 2005. ISBN: 9780321268174

Suggested Reference Book(s):

1. Jonathan Levin, Mac OS X and iOS Internals: To the Apple's Core, Wiley, 2012. ISBN: 9781118057650
2. RamezElmasri, A. Carrick, David Levine, Operating Systems: A Spiral Approach, McGraw-Hill,

2009. ISBN: 9780073385567
3. Pavel Yosifovich et al., Windows Internals, 7th Edition, Microsoft Press, 2017. ISBN: 9780735684188
4. Chris Simmonds, Mastering Embedded Linux Programming, 2nd Edition, Packt Publishing, 2017. ISBN: 9781787288853

Other useful resource(s):

1. Link to NPTEL course contents (Operating System): https://nptel.ac.in/courses/106108101/
2. NPTEL – Introduction to Android Programming: https://nptel.ac.in/courses/106106147/
3. NPTEL – Mobile Computing: https://nptel.ac.in/courses/106106147/
4. Android Developer Documentation: https://developer.android.com/docs
5. Apple Developer Documentation: https://developer.apple.com/documentation/

EvaluationScheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2) -10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	3	1	1	1	2	2	2	**1.91**
CO-2	3	3	2	3	3	1	1	1	2	2	2	**2.09**
CO-3	2	2	3	2	3	2	1	2	2	2	2	**2.09**
CO-4	2	3	3	3	3	2	1	2.	2	3	2	**2.36**
CO-5	2	3	3	2	3	3	2	2	2	2	3	**2.45**
Average	**2.4**	**2.6**	**2.6**	**2.4**	**3.0**	**1.8**	**1.2**	**1.6**	**2.0**	**2.2**	**2.2**	

Android & iOS Lab

COURSE CODE:

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: Basic knowledge of OOP, Java

1. To introduce students to the architecture, development environments, and core concepts of Android and iOS platforms
2. To enable students to design and implement user interfaces using basic and advanced widgets in Android and iOS.
3. To build functional mobile applications such as calculators, portals, and quizzes using proper layouts and storage mechanisms.
4. To implement authentication, security, and advanced features like voice control and timers in real-time mobile applications.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Demonstrate understanding of Android and iOS operating systems and development environments.	Apply
CO2	Develop mobile interfaces using widgets, layouts, and event handling mechanisms in Android and iOS.	Apply
CO3	Build and deploy mobile applications with user authentication, data handling, and UI responsiveness.	Evaluate, Create
CO4	Integrate advanced features like timers, quizzes, voice control, and admin panels into cross-platform mobile apps.	Evaluate, Create

List of Experiments

S. No	Description	Hours
1	Introduction to android & iOS operating system and study of basic widgets.	**4**
2	Study of Android & iOS lifecycle and demonstration of it.	**4**
3	Design the Calculator App in Android & iOS using different layouts.	**4**
4	Build a student portal showing students enrolled in courses and list of all students available in the courses in Android & iOS platforms.	**4**
5	Extend the application including the admin panel for providing authentication and security.	**4**
6	Build an application in Android & iOS platform for conduction online examination and Quiz having stopwatch facility.	**4**

7	Design a game in Android and iOS platform controlling the panel using voice commands.	**4**
Total Lab hours		**28**

Suggested/Resources:

1. Android Programming: The Big Nerd Ranch Guideby Bill Phillips & Chris Stewart
2. iOS Programming: The Big Nerd Ranch Guide by Christian Keur& Aaron Hillegass
3. Professional Android by Reto Meier
4. Swift Programming: The Big Nerd Ranch Guide by Matthew Mathias

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the ProgrammeOutcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	3	1	1	1	1	1	2	**1.73**
CO-2	3	3	3	3	3	2	1	2	2	1	2	**2.27**
CO-3	3	3	2	3	3	2	1	2	2	1	2	**2.18**
CO-4	3	3	3	3	3	2	1	2	2	1	2	**2.27**
Average	**3**	**2.75**	**2.5**	**2.75**	**3**	**1.75**	**1**	**1.75**	**1.75**	**1**	**2**	

Full Stack and Open-Source Software Development

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: CORE

L-T-P: 3-0-0

Pre-requisite: Basic knowledge of Programming and Web Technologies (HTML, CSS, JavaScript)

Course Objectives:

1. To understand the fundamentals of modern web application development.
2. To develop frontend applications using React and JavaScript.
3. To build backend services using Node.js, Express, and MongoDB.
4. To learn testing, deployment, and security practices in full stack applications.
5. To explore modern tools such as Redux, GraphQL, React Native, and containerization.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the architecture and components of modern full stack web applications.	Understand
CO-2	Develop interactive frontend applications using React, components, states, and event handling.	Apply
CO-3	Design and implement backend services using Node.js, Express, and MongoDB.	Apply
CO-4	Analyze and implement testing, authentication, and deployment strategies for web applications.	Analyze
CO-5	Develop scalable applications using Redux, GraphQL, containers, and orchestration concepts.	Create

Course Contents:

Unit	Contents	Lectures required
1	Fundamentals of Web apps, Introduction to React, JavaScript, component states, event handlers, debugging React apps	5
2	Rendering data collection and modules, forms, getting data from server, altering data in server, adding styles to React apps	5
3	Introduction of server programming with Node.js and express, deploying app to internet, Introduction of MongoDB, saving data to MongoDB, Validation and ESLint	6

4	Structure of Backend Application, introduction to testing, testing the backend, user administration tasks and control, token validation	5
5	Frontend login, props children and prototype, Testing react apps, End-to-End testing – Playwright, Cypress	5
6	Flux architecture and Redux, Reducers, Server communication with redux application, React Query, UseReducer and Context	5
7	React router, custom hooks, Styles, Webpacks, class components, Bloglists	5
8	Introduction to GraphQL server, Cache login and updates, Fragments and subscriptions, TypeScripts, React Native, Containers, Building and configuring environments, basics of Orchestration	6
Total Lectures		**42**

Suggested Text Book(s):

1. Marijn Haverbeke, *Eloquent JavaScript*, No Starch Press, 3rd Edition, 2018
2. Kyle Simpson, *You Don't Know JS (Book Series)*, O'Reilly Media, 2nd Edition, 2020
3. Chris Northwood, *The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer*, Apress, 1st Edition, 2018

Suggested Reference Book(s):

1. Brad Traversy, *Modern JavaScript from the Beginning*, Packt Publishing, 2020
2. Ethan Brown, *Web Development with Node and Express*, O'Reilly Media, 2nd Edition, 2019
3. Robin Nixon, *Learning PHP, MySQL & JavaScript*, O'Reilly Media, 5th Edition, 2018

Other useful resource(s):

1. Full Stack Web Development Courses (Coursera)
2. https://www.freecodecamp.org/learn/
3. https://www.theodinproject.com/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.2**
Average	**2.8**	**2.4**	**2.4**	**2.0**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.2**	

Full Stack and Open-Source Software Development Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: Basic knowledge of HTML, CSS, JavaScript

Course Objectives:

1. To understand practical usage of React, JavaScript, Node.js and MongoDB in full stack development.
2. To develop frontend web interfaces using React.
3. To implement backend services using Node.js and Express.
4. To manage and store application data using MongoDB databases.
5. To develop authentication and CRUD based web applications.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Develop web applications using React and JavaScript.	Apply
CO2	Build backend servers using Node.js and Express.	Apply
CO3	Design applications with database connectivity using MongoDB.	Apply
CO4	Implement CRUD operations using Node.js and Express.	Apply
CO5	Develop authentication-based applications with login mechanisms.	Apply

List of Experiments:

S. No	Description	Hours
1	Study the React, JavaScript, MongoDB, and Node.js along with their usage and application areas	4
2	Create a form and validate the contents of the form using React and Node.js	4
3	Create a Node.JS server that serves static HTML and CSS files to the user without using Express	4
4	Create a NodeJS server using Express that stores data from a form as a JSON file and displays it in another page. The redirect page should be prepared using Handlebars	4
5	Create a NodeJS server using Express that creates, reads, updates and deletes students' details and stores them in MongoDB database. The information about the	4

	user should be obtained from a HTML form	
6	Create a Todo application using ReactJS. Store the data to a JSON file using a simple Node.JS server and retrieve the information from the same during page reloads	4
7	Create a simple Sign up and Login mechanism and authenticate the user using cookies. The user information can be stored in either MongoDB or MySQL and the server should be built using Node.JS and Express Framework	4
Total Lab Hours		**28**

Suggested/Resources:

1. Alex Banks & Eve Porcello, Learning React, O'Reilly Media.
2. Ethan Brown, Web Development with Node and Express, O'Reilly Media.
3. Robin Wieruch, The Road to React.
4. Adam Freeman, Pro MERN Stack.
5. React Documentation – https://react.dev
6. MongoDB Documentation – https://mongodb.com

Other useful resource(s):

1. Full Stack Web Development Courses (Coursera)
2. https://www.freecodecamp.org/learn/
3. https://www.theodinproject.com/

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	3	1	3	1	1	2	2	1	2	**1.9**
CO-2	3	3	3	1	3	1	1	2	2	1	2	**2.0**
CO-3	3	2	2	1	3	1	1	1	1	1	2	**1.6**
CO-4	3	2	2	2	2	1	1	1	1	1	2	**1.6**
CO-5	3	2	3	2	3	1	1	1	2	2	2	**2.0**
Average	**3.0**	**2.2**	**2.6**	**1.4**	**2.8**	**1.0**	**1.0**	**1.4**	**1.6**	**1.2**	**2.0**	

Cloud Computing

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: CORE

L-T-P: 3-0-0

Pre-requisite: Nil

Course Objectives:

1. To understand the principles, architecture, and economic aspects of cloud computing.
2. To analyze virtualization, resource management, and cloud service models.
3. To apply cloud platforms and tools for application deployment.
4. To evaluate cloud security, governance, and performance issues.
5. To design scalable and efficient cloud-based solutions.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain cloud computing concepts, architecture, and service models.	Understand
CO-2	Apply virtualization and cloud platforms for application deployment	Apply
CO-3	Analyze resource management, scalability, and distributed systems in cloud	Analyze
CO-4	Evaluate cloud security, performance, and governance issues	Evaluate
CO-5	Design cloud-native applications and scalable systems	Create

Course Contents:

Unit	Contents	Lectures required
1	**Introduction & Cloud Fundamentals**: Definition, characteristics, evolution of cloud computing; NIST service models (IaaS, PaaS, SaaS); deployment models; cloud cube model; cloud economics (CapEx vs OpEx, ROI); benefits and challenges.	5
2	**Cloud Architecture & Services:** Layered architecture; data center infrastructure; SOA; web services and REST APIs; microservices; multi-tenancy; service lifecycle and orchestration	5
3	**Virtualization & Containerization:** Virtualization concepts; hypervisors; virtual machines vs containers; Docker basics; resource abstraction and pooling	6
4	**Resource Management & Capacity Planning:** Resource provisioning; scheduling; load balancing; elasticity and scalability; SLA and QoS; performance monitoring	3
5	**Cloud Platforms & PaaS**: PaaS architecture; application development; DevOps and CI/CD pipelines; application lifecycle management **Cloud Infrastructure & IaaS**: Infrastructure design; orchestration;	7

	automation tools; infrastructure monitoring	
6	**Public Cloud Platforms:** AWS (EC2, S3, Lambda); Microsoft Azure; Google Cloud Platform; comparative analysis; serverless computing (FaaS) **Data Management in Cloud:** Distributed storage; replication; consistency models; NoSQL databases; data lifecycle management	**10**
7	**Cloud Security & Governance:** Security architecture; IAM; encryption; privacy; compliance; risk management	**3**
8	**Advanced Topics and Emerging Trends** Edge computing, AI and Machine Learning in the cloud, AWS SageMaker, Azure ML Studio, Blockchain and cloud computing, Future trends in cloud-native platforms	**3**
Total lectures		**42**

Suggested Text Book(s):

1. Kai Hwang, Geoffrey Fox, and Jack Dongarra, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Morgan Kaufmann, 2012. ISBN: 978-0123858801.
2. Rajkumar Buyya, James Broberg, and Andrzej M. Goscinski (eds.), Cloud Computing: Principles and Paradigms, Wiley, 2011. ISBN: 978-0470887998.
3. Thomas Erl, Cloud Computing: Concepts, Technology & Architecture, Prentice Hall, 2013. ISBN: 978-0133387520.

Suggested Reference Book(s):

1. Barrie Sosinsky, Cloud Computing Bible, Wiley, 2010. ISBN: 978-0470903568.
2. Tim Mather, Subra Kumaraswamy, and Shahed Latif, Cloud Security and Privacy, O'Reilly Media, 2009. ISBN: 978-0596802769.
3. George Reese, Cloud Application Architectures, O'Reilly Media, 2009. ISBN: 978-0596156367.
4. Martin Kleppmann, Designing Data-Intensive Applications, O'Reilly Media, 2017. ISBN: 978-1449373320.
5. AWS Skill Builder
6. AWS Certified Cloud Practitioner Study Guide
7. Microsoft Azure Fundamentals AZ-900 Guide

Other useful resource(s):

1. Link to NPTEL course contents: https://onlinecourses.nptel.ac.in/noc26_cs55/preview
2. Link to NPTEL course contents: https://onlinecourses.nptel.ac.in/noc26_cs29/preview
3. Link to topics related to course: https://nptel.ac.in/courses/106105167

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus

4.	Teaching Assessment	25	Entire Semester	Technical Presentations (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	2	2	3	1	1	1	2	2	2	**2.1**
CO-3	3	3	2	3	3	1	1	1	2	2	2	**2.3**
CO-4	3	3	2	3	3	1	1	1	2	2	3	**2.4**
CO-5	3	3	3	3	3	1	1	2	3	2	3	**2.6**
Average	**3**	**2.8**	**2**	**2.4**	**2.8**	**1**	**1**	**1.2**	**2.2**	**1.8**	**2.4**	

Cloud Computing Lab: AWS and AZURE

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: None

Course Objectives:

1. Gain hands-on experience with AWS and Azure cloud services
2. Launch and manage virtual machines and cloud storage
3. Configure cloud networking and IAM
4. Build serverless and containerized applications
5. Use Kubernetes orchestration and monitoring tools
6. Practice migration and disaster recovery workflows

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Deploy and manage virtual machines on AWS and Azure.	Apply
CO2	Implement cloud storage, databases, and networking services	Apply
CO3	Configure IAM, security groups, and access policies.	Apply
CO4	Develop serverless and containerized cloud applications	Apply
CO5	Deploy scalable Kubernetes-based solutions.	Apply
CO6	Use IaC, migration, and disaster recovery tools.	Apply

List of Experiments

S. No	Description	Hours
1	**Experiment 1: AWS & Azure Account Setup** Create AWS Educate / Academy / Skill Builder account Create Azure student account Explore AWS Console and Azure Portal Compare service dashboards	**2**
2	**Experiment 2: Launch Virtual Machines** **AWS** Launch EC2 instance Configure AMI, instance type, storage Create key pair Configure security groups	**2**

	Azure Create Azure VM Configure VNet, subnet, NSG SSH / RDP access	
3	**Experiment 3: Cloud Storage Services** **AWS** Create S3 bucket Upload files Enable static website hosting Configure bucket policies **Azure** Create Blob Storage container Upload files Set access tiers Public blob access	**2**
4	**Experiment 4: Database Deployment** **AWS** Launch RDS instance Connect MySQL/PostgreSQL **Azure** Create Azure SQL Database Configure firewall rules Connect using SSMS / MySQL Workbench	**2**
5	**Experiment 5: Cloud Networking** **AWS** Create VPC Public and private subnets Route tables and internet gateway **Azure** Create VNet Subnets NSG rules Azure Bastion (optional)	**2**
6	**Experiment 6: Identity and Access Management** **AWS** Create IAM users, roles, policies Least privilege access **Azure** Azure Active Directory RBAC role assignments Conditional access basics	**2**
7	**Experiment 7: Serverless Computing** **AWS** Create Lambda function API Gateway integration	**2**

	Trigger via S3 event **Azure** Create Azure Function App HTTP trigger Blob trigger	
8	**Experiment 8: Auto Scaling and Monitoring** **AWS** Auto Scaling Group CloudWatch alarms CPU-based scaling **Azure** VM Scale Sets Azure Monitor Autoscale rules	**2**
9	**Experiment 9: Docker Containerization** Install Docker Create Dockerfile Build image Push image to: AWS ECR Azure Container Registry (ACR)	**2**
10	**Experiment 10: Kubernetes Orchestration** **AWS** Deploy on EKS **Azure** Deploy on AKS **Tasks**: • Pod deployment • Replica sets • Services • Ingress • Scaling pods	
11	**Experiment 11: Infrastructure as Code** **AWS** AWS CloudFormation template **Azure** ARM Template / Bicep **Tasks:** • Deploy VM + Storage + Networking using templates	**2**
12	**Experiment 12: Cloud Migration Tools** **AWS** AWS Migration Hub AWS DMS demo **Azure**	**2**

	Azure Migrate demo VM assessment	
13	**Experiment 13: Backup and Disaster Recovery** **AWS** EBS snapshot S3 versioning Multi-AZ RDS **Azure** Azure Backup Azure Site Recovery	**2**
14	**Experiment 14: Mini Project / Capstone** Students must implement **one cloud-native project** using AWS or Azure. Suggested Projects • 3-tier web app deployment • Serverless attendance system • Secure file-sharing portal • Kubernetes microservice app • Cost-optimized backup solution • AI API deployment on cloud • Etc…	2
Total Lab hours		**28**

Suggested/Resources:

1. AWS Academy Learner Lab
2. AWS Skill Builder
3. Azure for Students
4. Microsoft Learn Sandbox
5. Docker Desktop
6. Minikube / AKS / EKS

Industry Certification:

AWS

- AWS Cloud Practitioner
- AWS Solutions Architect Associate

Azure

- AZ-900 Azure Fundamentals
- AZ-104 Azure Administrator

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, AWS Lab, Azure Lab, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	–	3	–	–	–	1	1	–	**2**
CO-2	3	2	3	1	3	–	1	–	–	1	–	**2**
CO-3	3	3	3	1	3	1	–	2	–	1	–	**2.1**
CO-4	3	2	3	2	3	–	–	–	2	1	1	**2.1**
CO-5	3	2	3	2	3	–	–	–	2	1	1	**2.1**
CO-6	2	3	3	3	3	1	1	1	–	1	2	**2**
Average	**2.8**	**2.3**	**2.8**	**1.8**	**3**	**1**	**1**	**1.5**	**1.6**	**1**	**1.3**	

Machine Learning and Natural Language Processing

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Probability and Statistics, Linear Algebra, Python Programming, Data Structures and Algorithms.

Course Objectives:

1. To introduce the principles, algorithms, and mathematical foundations of supervised and unsupervised machine learning.
2. To develop understanding of ensemble methods, model selection, regularisation, and performance evaluation techniques.
3. To familiarise students with the linguistic foundations of NLP and classical text processing pipelines including language models, POS tagging, and parsing.
4. To build understanding of deep learning architectures — RNNs, LSTMs, and Transformers — and their application to NLP tasks.
5. To explore state-of-the-art pre-trained language models (BERT, GPT, T5) and analyse their application to downstream NLP tasks.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the foundational concepts of machine learning including supervised, unsupervised, and reinforcement learning paradigms, along with bias-variance trade-off and model evaluation techniques.	Understand
CO-2	Explain core NLP concepts including text preprocessing, language models, POS tagging, parsing, and sentiment analysis using both statistical and neural approaches.	Understand
CO-3	Apply classical ML algorithms — linear/logistic regression, SVMs, decision trees, and ensemble methods — to solve real-world structured and text-based problems.	Apply
CO-4	Apply deep learning architectures — RNNs, LSTMs, and Transformers — and pre-trained language models (BERT, GPT, T5) to NLP tasks such as classification, NER, and machine translation.	Apply
CO-5	Analyse challenges in ML and NLP pipelines including statistical heterogeneity, model selection, hyperparameter tuning, and performance interpretation on benchmark datasets.	Analyze

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Machine Learning: Types of ML — supervised, unsupervised, semi-supervised, reinforcement learning; the ML pipeline — data collection, preprocessing, feature engineering, model training, evaluation, deployment; bias-variance trade-off; overfitting and underfitting; cross-validation; performance metrics — accuracy, precision, recall, F1, ROC-AUC.	**4**

2	Supervised Learning Algorithms: Linear regression — OLS, gradient descent, regularisation (Ridge, Lasso); logistic regression — binary and multiclass; Support Vector Machines — hard and soft margin, kernel trick; k-Nearest Neighbours; Naive Bayes classifier; decision trees — ID3, C4.5, CART; ensemble methods — bagging, random forests, boosting (AdaBoost, Gradient Boosting, XGBoost).	**9**
3	Unsupervised Learning and Dimensionality Reduction: K-Means clustering — algorithm, elbow method; hierarchical clustering — agglomerative, dendrogram; DBSCAN; Gaussian Mixture Models and EM algorithm; dimensionality reduction — PCA, t-SNE; anomaly detection.	**5**
4	Introduction to Natural Language Processing: Text as data — challenges and pipeline; text preprocessing — tokenisation, stop-word removal, stemming, lemmatisation; text representation — Bag of Words, TF-IDF, n-grams; word embeddings — Word2Vec (CBOW, Skip-gram), GloVe, FastText; introduction to NLTK and spaCy.	**5**
5	Core NLP Tasks — Statistical and Classical Approaches: Language models — N-gram models, smoothing techniques; Part-of-Speech (POS) tagging — HMM-based tagging, Viterbi algorithm; Named Entity Recognition (NER) — rule-based and CRF-based; syntactic parsing — constituency and dependency parsing; sentiment analysis — lexicon-based and ML-based approaches; topic modelling — LDA.	**7**
6	Deep Learning for NLP: Recurrent Neural Networks (RNNs) — vanishing gradient problem; Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU); sequence-to-sequence models with attention mechanism; Convolutional Neural Networks for text classification; Transformer architecture — self-attention, multi-head attention, positional encoding; encoder-decoder structure.	**6**
7	Pre-trained Language Models and Transfer Learning: BERT — pre-training objectives (MLM, NSP), fine-tuning for downstream tasks; GPT — autoregressive language modelling; RoBERTa, DistilBERT; T5 and BART for seq2seq tasks; prompt engineering and in-context learning; introduction to Large Language Models (LLMs); Hugging Face Transformers library — pipelines and tokenisers.	**6**
Total lectures		**42**

Suggested Text Book(s):

1. Bishop, C. M., Pattern Recognition and Machine Learning, Springer, 2006.
2. Jurafsky, D. and Martin, J. H., Speech and Language Processing, 3rd Edition (Draft), Pearson, 2023. [Available free online]
3. Géron, A., Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition, O'Reilly Media, 2022.
4. Goldberg, Y., Neural Network Methods for Natural Language Processing, Morgan & Claypool Publishers, 2017.

Suggested Reference Book(s):

1. Mitchell, T. M., Machine Learning, McGraw-Hill Education, 1997.
2. Manning, C. D. and Schütze, H., Foundations of Statistical Natural Language Processing, MIT Press, 1999.
3. Goodfellow, I., Bengio, Y., and Courville, A., Deep Learning, MIT Press, 2016.
4. Tunstall, L., von Werra, L., and Wolf, T., Natural Language Processing with Transformers, O'Reilly Media, 2022.

Other useful resource(s):

1. Hugging Face Transformers Documentation: https://huggingface.co/docs/transformers
2. spaCy Documentation and Tutorials: https://spacy.io
3. NLTK Book (Free): https://www.nltk.org/book
4. fast.ai Practical Deep Learning for Coders: https://course.fast.ai
5. NPTEL – Introduction to Machine Learning (IIT Madras): https://nptel.ac.in/courses/106106139
6. NPTEL – Deep Learning for NLP: https://nptel.ac.in/courses/106106198
7. Papers With Code – NLP Benchmarks: https://paperswithcode.com/area/natural-language-processing
8. Stanford CS224N – NLP with Deep Learning (Lecture Notes & Videos): https://web.stanford.edu/class/cs224n

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	1.6
CO-2	3	2	2	2	2	1	1	1	2	1	2	1.7
CO-3	3	3	3	2	3	1	1	1	2	2	2	2.1
CO-4	3	3	3	2	3	1	1	1	2	2	2	2.1
CO-5	2	3	2	3	2	1	1	1	2	2	3	2.0
Average	**2.8**	**2.6**	**2.4**	**2.0**	**2.4**	**1.0**	**1.0**	**1.0**	**2.0**	**1.6**	**2.2**	

Ethical Hacking & Prevention

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Basic knowledge of Computer Networks and Operating Systems

Course Objectives:

1. To understand the fundamentals of ethical hacking, cybersecurity, and threat landscape.
2. To learn various attack techniques used by hackers and their prevention mechanisms.
3. To analyze vulnerabilities in systems, networks, and web applications.
4. To apply penetration testing tools and methodologies in real-world scenarios.
5. To develop secure systems by implementing best security practices and counter measures.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain fundamentals of ethical hacking, cybersecurity concepts, and types of threats.	Understand
CO-2	Apply various scanning, enumeration, and vulnerability assessment techniques.	Apply
CO-3	Analyze different attack vectors including network, system, and web-based attacks.	Analyze
CO-4	Evaluate security tools and perform penetration testing.	Evaluate
CO-5	Design secure systems and implement preventive measures against cyber threats.	Create

Course Contents:
Course Contents:

Unit	Contents	Lectures required
1	Introduction to Ethical Hacking: Basics of Cybersecurity, Types of Hackers, Cyber Laws, Footprinting	**4**
2	Footprinting& Reconnaissance: WHOIS, DNS, Social Engineering	**5**
3	Scanning & Enumeration: Port Scanning, Vulnerability Scanning, SNMP, LDAP	**6**
4	System Hacking: Password Cracking, Malware, Privilege Escalation	**6**
5	Network Attacks & Prevention: DoS/DDoS, MITM, Firewalls, IDS/IPS	**7**
6	Web Application Security: SQL Injection, XSS, CSRF	**7**

7	Wireless & Mobile Security: Wi-Fi Attacks, Mobile Security	**4**
8	Penetration Testing & Countermeasures: Tools and Phases	**3**
Total lectures		**42**

Suggested Text Book(s):

1. Kevin Mitnick, The Art of Invisibility, Little, Brown and Company, 2017
2. Michael T. Simpson, Ethical Hacking and Network Defense, Cengage Learning, 3rd Edition
3. Rafay Baloch, Ethical Hacking and Penetration Testing Guide, CRC Press, 2014
4. William Stallings, Network Security Essentials, Pearson, 6th Edition

Suggested Reference Book(s):

1. Jon Erickson, Hacking: The Art of Exploitation, No Starch Press
2. Georgia Weidman, Penetration Testing: A Hands-On Introduction to Hacking, No Starch Press

Other useful resource(s):

1. OWASP Foundation: https://owasp.org
2. Kali Linux Documentation: https://www.kali.org

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO 1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO1 0	PO1 1	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.2**
Average	**2.8**	**2.4**	**2.4**	**2.0**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.2**	

Introduction to Large Scale Database Systems

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Database Management Systems, Data Structures, Operating Systems

Course Objectives:

1. To understand the design challenges of large-scale and distributed database systems.
2. To study modern data storage architectures used in cloud and big data platforms.
3. To analyze scalability, consistency, availability, and fault tolerance in distributed databases.
4. To explore NoSQL databases and large-scale data processing frameworks.
5. To design scalable data management solutions for real-world applications.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain architecture and principles of large-scale database systems.	Understand
CO-2	Apply distributed database concepts for scalable data storage and processing.	Apply
CO-3	Analyze consistency models, replication strategies, and fault tolerance mechanisms.	Analyze
CO-4	Evaluate NoSQL and big data technologies for different application scenarios.	Evaluate
CO-5	Design scalable database solutions for large-scale real-world systems.	Create

Course Contents:

Unit	Contents	Lectures Required
1	**Introduction to Large-Scale Data Systems:** Evolution from traditional DBMS to distributed databases, data-intensive applications, scalability challenges, and centralized vs distributed databases, overview of cloud data platforms.	5
2	**Distributed Database Architecture:** Data partitioning (sharding), replication strategies, distributed query processing, distributed transactions, two-phase commit protocol, distributed indexing.	7
3	**Consistency and CAP Theorem:** Consistency models, CAP theorem, ACID vs BASE properties, eventual consistency, quorum-based systems, concurrency control in distributed environments.	6
4	**NoSQL Database Systems:** Key-value stores, document databases, column-family databases, graph databases; systems such as MongoDB, Cassandra, HBase; schema design for scalability.	7

5	**Large-Scale Data Processing Frameworks:** MapReduce paradigm, Hadoop ecosystem, Spark architecture, distributed storage using HDFS, batch vs stream processing concepts.	6
6	**Fault Tolerance and Reliability:** Data replication, failure detection, recovery mechanisms, consensus algorithms (Paxos/Raft overview), load balancing, distributed monitoring.	6
7	**Cloud Databases and Modern Data Platforms:** Database-as-a-Service (DBaaS), serverless databases, data lakes and data warehouses, real-time analytics systems, security and privacy in large-scale databases.	5
Total lectures		**42**

Suggested Text Book(s)

1. Martin Kleppmann, Designing Data-Intensive Applications, O'Reilly Media.
2. Hector Garcia-Molina, Jeffrey Ullman & Jennifer Widom, Database Systems: The Complete Book, Pearson.
3. Raghu Ramakrishnan & Johannes Gehrke, Database Management Systems, McGraw-Hill.Tom White, Hadoop: The Definitive Guide, O'Reilly.

Suggested Reference Book(s)

1. Ozsu&Valduriez, Principles of Distributed Database Systems, Springer.
2. George Coulouris et al., Distributed Systems: Concepts and Design, Pearson.
3. Bill Chambers &MateiZaharia, Spark: The Definitive Guide, O'Reilly.
4. Kleppmann Lecture Notes on Distributed Systems (University of Cambridge).

Other useful resource(s):

1. MIT OpenCourseWare – Distributed Computer Systems Engineering (6.824): https://ocw.mit.edu/courses/6-824-distributed-computer-systems-engineering-spring-2006/
2. NPTEL – Big Data Computing (IIT Patna): https://onlinecourses.nptel.ac.in/noc24_cs130/
3. NPTEL – Distributed Systems: https://onlinecourses-archive.nptel.ac.in/noc17_cs42/
4. Apache Hadoop Official Documentation: https://hadoop.apache.org/docs/
5. MongoDB University Courses: https://university.mongodb.com

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2) -10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	2	1	1	2	**1.8**
CO-2	3	3	3	2	3	1	1	2	2	2	2	**2.2**
CO-3	3	3	2	3	3	1	1	2	2	2	2	**2.2**
CO-4	2	3	3	3	3	1	1	2	2	2	3	**2.3**
CO-5	3	2	3	2	3	1	1	3	3	2	2	**2.3**
Average	**2.8**	**2.6**	**2.6**	**2.2**	**2.8**	**1.0**	**1.0**	**2.2**	**2.0**	**1.8**	**2.2**	

Introduction to DevOps

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: N/A

Course Objectives:

1. To understand DevOps principles, culture, and evolution in software development.
2. To apply DevOps practices like CI/CD, TDD, and automation.
3. To analyze cloud-native architectures and microservices.
4. To evaluate collaboration, workflows, and DevOps tools.
5. To measure DevOps performance using metrics and continuous improvement strategies.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain DevOps concepts, principles, and cultural transformation.	Understand
CO-2	Apply DevOps practices like CI/CD, TDD, and automation tools.	Apply
CO-3	Analyze cloud-native architectures and microservices-based systems.	Analyze
CO-4	Evaluate collaboration, workflows, and DevOps lifecycle processes.	Analyze
CO-5	Measure DevOps performance using metrics and continuous improvement strategies.	Evaluate

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to DevOps:** Evolution of software development (Waterfall vs Agile), innovation and business disruption, role of technology, DevOps definition, misconceptions, DevOps culture, characteristics, history and evolution of DevOps movement	**8**
2	**Thinking DevOps:** Social coding, version control and shared repositories, pair programming, working in small batches, MVP concept, feedback loops, TDD, BDD, cloud-native architecture, microservices, resilience engineering	**10**
3	**Working with DevOps:** Software development as craft vs factory, silos and bottlenecks, collaboration, Infrastructure as Code (IaC), ephemeral infrastructure, containers and Docker basics, immutable infrastructure, CI/CD pipelines, automation	**10**

4	**Organizing for DevOps:** Organizational impact, DevOps team structures, cross-functional teams, shared responsibility, misconceptions, consequences in workflows, building shared mindset	**7**
5	**Measuring DevOps:** DevOps metrics (deployment frequency, lead time), social metrics, continuous improvement, actionable metrics, blameless culture, measuring culture, reward systems	**7**
Total lectures		**42**

Suggested Text Book(s):

1. The Phoenix Project – Gene Kim, Kevin Behr, George Spafford
2. The DevOps Handbook – Gene Kim, Jez Humble, Patrick Debois
3. Accelerate – Nicole Forsgren, Jez Humble, Gene Kim

Suggested Reference Book(s):

1. Continuous Delivery – Jez Humble and David Farley
2. Site Reliability Engineering – Betsy Beyer et al.

Other useful resource(s):

1. NPTEL DevOps Courses:https://elearn.nptel.ac.in/shop/completed-courses/short-term-programs-completed/cicd-devops-automation-and-devsecops-automation/
2. IBM DevOps Learning Resources:https://developer.ibm.com/devpractices/devops/tutorials/
3. AWS DevOps Tutorials: https://market.tutorialspoint.com/course/aws-devops-engineer-certification-2022/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	2	1	1	1	2	1	2	1.8
CO-2	3	3	3	2	3	1	1	1	2	2	2	2.1
CO-3	3	2	3	3	3	1	1	1	2	2	2	2.2
CO-4	2	3	2	3	2	1	1	1	2	2	2	2.0
CO-5	3	2	3	2	3	1	1	2	3	2	2	2.2
Average	**2.8**	**2.4**	**2.6**	**2.4**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.0**	

Industrial Automation and IoT

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Basic Electronics, Microprocessors and Microcontrollers, Computer Networks, Programming in C/Python

Course Objectives:

1. To understand the architecture, components, and hierarchy of modern industrial automation systems including sensors, actuators, PLCs, and DCS.
2. To study SCADA systems, HMI design, and industrial communication protocols used in process control and manufacturing environments.
3. To explore the architecture, hardware platforms, and communication protocols of the Internet of Things and Industrial IoT.
4. To analyse edge computing, digital twin concepts, and IIoT data pipelines for industrial monitoring and predictive maintenance.
5. To examine cybersecurity threats, safety standards, and regulatory frameworks governing industrial automation and IIoT deployments.

Course Outcomes:

S.No.	Course Outcomes	Level of Attainment
CO-1	Explain the fundamental concepts of industrial automation, control systems, sensors, actuators, and programmable logic controllers (PLCs) used in modern manufacturing environments.	Understand
CO-2	Explain the architecture, communication protocols, and standards of the Industrial Internet of Things (IIoT) including MQTT, OPC-UA, Modbus, and PROFINET.	Understand
CO-3	Apply SCADA systems, HMI design principles, and industrial networking concepts to model real-world automated process control scenarios.	Apply
CO-4	Apply IoT sensing, edge computing, and data acquisition techniques to design IIoT data pipelines for industrial monitoring and predictive maintenance.	Apply
CO-5	Analyse industrial automation systems and IIoT deployments with respect to reliability, cybersecurity threats, real-time constraints, and safety standards.	Analyze

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Industrial Automation: Evolution of automation — manual, mechanised, automated, and smart manufacturing; automation hierarchy — field level, control level, supervisory level, enterprise level; types of automation — fixed, programmable, flexible; key components — sensors, actuators, controllers, HMI; introduction to Industry 4.0 and cyber-physical systems (CPS); overview of smart factories.	**5**

2	Sensors, Actuators and Signal Conditioning: Classification of sensors — temperature (RTD, thermocouple), pressure, flow, proximity, vision; signal types — analog vs. digital, 4–20 mA current loops; signal conditioning — amplification, filtering, ADC/DAC; actuators — electric motors (DC, stepper, servo), pneumatic and hydraulic actuators, solenoid valves; transducer characteristics — accuracy, resolution, linearity, repeatability.	**6**
3	Programmable Logic Controllers (PLCs) and Control Systems: PLC architecture — CPU, memory, I/O modules, power supply; PLC programming languages (IEC 61131-3) — Ladder Diagram, Function Block Diagram, Structured Text, Sequential Function Chart; closed-loop control — PID controllers, tuning methods; motion control basics; safety PLCs and functional safety standards (IEC 61508, IEC 62061); introduction to Distributed Control Systems (DCS).	**8**
4	SCADA, HMI and Industrial Networking: SCADA system architecture — field devices, RTUs/PLCs, communication network, SCADA server, historian; HMI design principles — alarm management, data visualisation, operator interface; industrial communication protocols — Modbus RTU/TCP, PROFIBUS, PROFINET, EtherNet/IP, OPC and OPC-UA; fieldbus vs. industrial Ethernet; network topologies in automation; introduction to time-sensitive networking (TSN).	**7**
5	Internet of Things (IoT) — Architecture and Protocols: IoT reference architecture — perception, network, and application layers; IoT hardware — microcontrollers (Arduino, ESP32), single-board computers (Raspberry Pi); wireless communication technologies — Wi-Fi, Bluetooth, Zigbee, Z-Wave, LoRa, NB-IoT, 5G; IoT application protocols — MQTT, CoAP, AMQP, HTTP/REST; IoT platforms — AWS IoT, Azure IoT Hub, Google Cloud IoT, Things Board.	**6**
6	Industrial IoT (IIoT) and Edge Computing: IIoT vs. consumer IoT; IIoT reference architecture (RAMI 4.0, IIRA); edge computing in industrial settings — edge gateways, fog nodes; real-time data acquisition and preprocessing; time-series data handling; digital twins — concept, architecture, and industrial applications; OPC-UA over MQTT (Sparkplug B); IIoT use cases — predictive maintenance, condition monitoring, energy management, asset tracking.	**7**
7	Cybersecurity, Safety and Standards in Industrial Automation and IIoT: Cybersecurity threats in OT/ICS environments — ransomware, man-in-the-middle, denial of service; ICS security frameworks — IEC 62443, NIST SP 800-82; network segmentation — DMZ, Purdue model, zero-trust architecture; secure remote access; safety instrumented systems (SIS) and safety integrity levels (SIL); regulatory and interoperability standards — ISO 9001, ISO 50001, IEEE 2413; emerging trends — AI in automation, autonomous robots, collaborative robots (cobots), 5G in IIoT.	**3**
Total lectures		**42**

Suggested Text Book(s):

1. Groover, M. P., Automation, Production Systems, and Computer-Integrated Manufacturing, 4th Edition, Pearson, 2015.
2. Mackay, S., Wright, E., Reynders, D., and Park, J., Practical Industrial Data Networks: Design, Installation and Troubleshooting, Newnes, 2004.
3. Bahga, A. and Madisetti, V., Internet of Things: A Hands-On Approach, VPT, 2014.
4. Bolton, W., Programmable Logic Controllers, 6th Edition, Newnes, 2015.

Suggested Reference Book(s):

1. Mahalik, N. P., Fieldbus Technology: Industrial Network Standards for Real-Time Distributed Control, Springer, 2003.
2. Buyya, R. and Dastjerdi, A. V., Internet of Things: Principles and Paradigms, Morgan Kaufmann, 2016.
3. Zurawski, R. (Ed.), Industrial Communication Technology Handbook, 2nd Edition, CRC Press, 2014.
4. Benaben, F. and Stuber, W. (Eds.), Interoperability in Digital Industry, Springer, 2021.

Other useful resource(s):

1. NPTEL – Industrial Automation (IIT Kharagpur): https://nptel.ac.in/courses/10710506.
2. NPTEL – Internet of Things (IIT Bombay): https://nptel.ac.in/courses/108101093
3. Siemens Automation Learning Portal: https://support.industry.siemens.com
4. Eclipse IoT (Open-source IoT Frameworks): https://iot.eclipse.org
5. ThingsBoard IoT Platform (Open-source): https://thingsboard.io
6. IEC 61131-3 PLC Programming Reference: https://plcopen.org
7. Papers With Code – IoT and Edge Computing: https://paperswithcode.com

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) -10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	2	1	2	1	2	1.7
CO-2	3	2	2	2	3	1	1	1	2	1	2	1.8
CO-3	3	3	3	2	3	1	1	1	2	2	2	2.1
CO-4	3	3	2	2	3	1	1	1	2	2	2	2.0
CO-5	2	3	2	3	2	2	2	2	2	2	2	2.2
Average	**2.8**	**2.6**	**2.2**	**2.0**	**2.6**	**1.2**	**1.4**	**1.2**	**2.0**	**1.6**	**2.0**	