Savitribai Phule Pune University, Pune, Maharashtra, India

सावित्रीबाई फुले पुणे विद्यापीठ

Faculty of Science and Technology

National Education Policy (NEP)-2020 Compliant Curriculum

Third Year Engineering (2024 Pattern) in Information Technology

(With effect from Academic Year 2026-27)

Contents

Preface

It is with great pride and a deep sense of responsibility that I present the NEP 2020-compliant detailed 2024 pattern curriculum for the **Third Year Bachelor of Engineering in Information Technology** at **Savitribai Phule Pune University, Pune**.

In alignment with the visionary framework of the **National Education Policy 2020**, this revised curriculum represents a transformative step towards creating a more holistic, multidisciplinary, and flexible engineering education. The focus is not only on deepening the technical knowledge base of our students, but also on fostering innovation, creativity, ethical thinking, and lifelong learning skills essential in the fast-evolving digital era.

The curriculum for the second year has been thoughtfully structured to strike a balance between core computing fundamentals—such as data structures and algorithms, object oriented programming, basics of computer network, computer graphics, database management systems, and software engineering—and emerging domains like digital marketing and social media, e-commerce. In addition, we have integrated value-added courses, skill development modules, interdisciplinary learning components as open electives, and community engagement project to promote practical understanding and industry readiness.

This curriculum is the result of rigorous consultations, deliberations, and collaborative work by esteemed faculty members, industry experts, and academic stakeholders. I extend my heartfelt appreciation to all contributors for their commitment to excellence and innovation in education.

As we implement this forward-looking curriculum, we reaffirm our commitment to nurturing IT professionals who are not only technically sound but also socially responsible and globally competent.

Dr. Sudeep D. Thepade
Chairman, Board of Studies- Information Technology
Savitribai Phule Pune University
Pune

Abbreviations

AEC	Ability Enhancement Course
BSC	Basic Science Course
CCC	Co-Curricular Courses
CCE	Comprehensive Continuous Evaluation
CEP	Common Engineering Project
CO	Course Outcome
ELC	Experiential Learning Courses
ESC	Engineering Science Course
FP	Field Project
IKS	Indian Knowledge System
INT	Internship
MDM	Multidisciplinary Minor
NEP	National Education Policy
OE	Open Elective
OJT	On Job Training
PCC	Program Core Course
PEC	Programme Elective Course
PO	Program Outcomes
PR	Practical
PRJ	Project
PSO	Program Specific Outcome
RM	Research Methodology
TH	Theory
TU	Tutorials
VEC	Value Education Course
VSE	Vocational and Skill Enhancement Course

Knowledge and Attitude Profile (WK)

A Knowledge and Attitude Profile (KAP), often represented as WK (Knowledge and Attitude Profile) in some contexts, is a framework or assessment tool used to evaluate an individual's knowledge and attitudes related to a specific area, topic, or domain.

WK1	A systematic, theory-based understanding of the natural sciences applicable to the discipline and awareness of relevant social sciences.
WK2	Conceptually-based mathematics, numerical analysis, data analysis, statistics and formal aspects of computer and information science to support detailed analysis and modelling applicable to the discipline.
WK3	A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline.
WK4	Engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline; much is at the forefront of the discipline.
WK5	Knowledge, including efficient resource use, environmental impacts, whole-life cost, re-use of resources, net zero carbon, and similar concepts, that supports engineering design and operations in a practice area.
WK6	Knowledge of engineering practice (technology) in the practice areas in the engineering discipline.
WK7	Knowledge of the role of engineering in society and identified issues in engineering practice in the discipline, such as the professional responsibility of an engineer to public safety and sustainable development.
WK8	Engagement with selected knowledge in the current research literature of the discipline, awareness of the power of critical thinking and creative approaches to evaluate emerging issues.
WK9	Ethics, inclusive behavior and conduct. Knowledge of professional ethics, responsibilities, and norms of engineering practice. Awareness of the need for diversity by reason of ethnicity, gender, age, physical ability etc. with mutual understanding and respect, and of inclusive attitudes.

Reference: Self-Assessment Report (SAR) Format Undergraduate Engineering Programs Graduate Attributes and Professional Competencies Version 4.0 (GAPC V4.0) - (August 2024) Page 55.

Savitribai Phule Pune University Third Year Information Technology		
Program Outcomes (POs)		
Learners are expected to know and be able to–		
PO1	Engineering knowledge	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	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/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. (WK1, WK5, and WK7).
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)

A graduate of the Information Technology Program will demonstrate-

PSO1	An ability to apply the theoretical concepts and practical knowledge of Information Technology in analysis, design, development and management of information processing systems and applications in the interdisciplinary domain.
PSO2	An ability to analyze a problem, and identify and define the computing infrastructure and operations requirements appropriate to its solution. IT graduates should be able to work on large-scale computing systems.
PSO3	An understanding of professional, business and business processes, ethical, legal, security and social issues and responsibilities.
PSO4	Practice communication and decision-making skills through the use of appropriate technology and be ready for professional responsibilities.

Savitribai Phule Pune University, Pune

Board of Studies Information Technology

Curriculum Structure of Third Year Engineering (2024 pattern)

SEMESTER V

Course Code	Course Type	Course Name	Teaching Scheme			Examination Scheme						Credits			
			Theory	Tutorial	Practical	CCE	EndSem	TermWork	Practical	Oral	Total	Theory	Tutorial	Practical	Total
PCC-301-IT	PCC 1	Operating System	3	-	-	30	70	-	-	-	100	3	-	-	3
PCC-302- IT	PCC 2	Artificial Intelligence & Machine Learning	3	-	-	30	70	-	-	-	100	3	-	-	3
PCC-303-IT	PCC 3	Theory of Computation	3	-	-	30	70	-	-	-	100	3	-	-	3
PCC-304-IT	PCC Lab 1	Operating System Lab 1	-	-	2	-	-	25	25	-	50	-	-	1	1
PCC-305-IT	PCC Lab 2	Artificial Intelligence & Machine Learning Lab 2	-	-	4	-	-	25	25	-	50	-	-	2	2
PEC-321-IT	PEC 1	Elective – I	3	-	-	30	70	-	-	-	100	3	-	-	3
PEC-322-IT	PEC 1 Lab 3	Elective – 1 Lab 3	-	-	2	-	-	25	25	-	50	-	-	1	1
MDM-331-IT	MDM	IOT based Product Development		2	2	-	-	25	-	25	50	-	2	1	3
OLE-341-IT	OE	Open Elective*	2	-	-	15	35	-	-	-	50	2	-	-	2
ELC-342-IT	ELC	Technical Seminar	-	-	2	-	-	-	-	50	50	-	-	1	1
Total			14	2	12	135	315	100	75	75	700	14	2	6	22

Programme Elective I (PEC)

PEC-321A-IT	Cloud Computing
PEC-321B-IT	Advanced Database Management System
PEC-321C-IT	Web Application Development
PEC-321D-IT	Image Processing Computer Vision

***Open Elective (OLE)**

Students can opt for Open Electives offered by different discipline/faculty like Arts, Science, Commerce, Management, Humanities or Inter-Disciplinary studies.

Example – Open Elective I- IPR and Cyber Laws, Sustainability Development, Digital Personal Data Protection, The Constitution of India.

Savitribai Phule Pune University, Pune

Board of Studies Information Technology

Curriculum Structure of Third Year Engineering (2024 pattern)

SEMESTER VI

Course Code	Course Type	Course Name	Teaching Scheme			Examination Scheme						Credits			
			Theory	Tutorial	Practical	CCE	EndSem	Term Work	Practical	Oral	Total	Theory	Tutorial	Practical	Total
PCC-351-IT	PCC 1	Data Science & Big Data Analytics	3	-	-	30	70	-	-	-	100	3	-	-	3
PCC-352-IT	PCC 2	Software Engineering & Project Management	2	-	-	30	70	-	-	-	100	2	-	-	2
PCC-353-IT	PCC Lab 1	Data Science & Big Data Analytics Lab 1	-	-	4	-	-	25	25	-	50	-	-	2	2
PCC-354-IT	PCC Lab 2	Software Engineering & Project Management Lab 2	-	-	2	-	-	25	25	-	50	-	-	1	1
PEC-361-IT	PEC II	Elective II	3	-	-	30	70	-	-	-	100	3	-	-	3
PEC-362-IT	PEC III	Elective III	3	-	-	30	70	-	-	-	100	3	-	-	3
PEC-363-IT	PEC III Lab 3	Elective III Lab 3	-	-	2	-	-	25	25	-	50	-	-	1	1
MDM-371-IT	MDM	Multimedia & Animation	-	1	2	-	-	25	-	25	50	-	1	1	2
VSE-372-IT	VSE	Full Stack Development with MERN	-	-	2	-	-	50	-	-	50	-	-	1	1
ELC-381-IT	ELC	Internship/OJT	-	-	8	-	-	-	-	50	50	-	-	4	4
Total			11	1	20	120	280	150	75	75	700	11	1	10	22

Programme Elective Course II (PEC)

PEC-361A-IT	DevOps
PEC-361B-IT	Quantum Computing
PEC-361C-IT	Blockchain Technology
PEC-361D-IT	Natural Language Processing

Programme Elective Course III (PEC)

PEC-362A-IT	Power BI & Tableau
PEC-362B-IT	High Performance Computing
PEC-362C-IT	Mobile Application Development
PEC-362D-IT	Cyber Security & Risk Management

General Guidelines

Credit Specifications

i Theory Courses: 13-15 hours of teaching per credit is required in a semester.

ii Laboratory Course: 26-30 hours in laboratory activities per credit is required in a semester.

iii Studio activities: Studio activities involve the engagement of students in creative or artistic activities. Every student is engaged in performing a creative activity to obtain a specific outcome. Studio-based activities involve visual- or aesthetic-focused experiential work. A minimum of 26-30 hours in studio activities per credit in a semester is required.

iv Workshop-based activities: Courses involving workshop-based activities require the engagement of students in hands-on activities related to work/vocation or professional practice. Every student is engaged in performing a skill-based activity. Related to specific learning outcome(s). 26- 30 hours of workshop-based activities per credit in a semester is required.

v Seminar/ Group Discussion: 13-15 hours of participation in seminar/ Group Discussion activity per credit in a semester is required.

vi Internship: Credits for internship shall be one credit per two weeks of internship (or 36-40 hours of engagement), The internship shall be monitored jointly by the faculty and Industry/ Organisation Mentor. Internship of One Semester duration shall be offered either in the VII or VIII semesters. Courses offered during the Internship Semester shall be offered in online mode.

vii Field-based Learning/ Practices: These are the courses requiring students to participate in field-based learning/projects generally under the supervision of faculty. A minimum of 26-30 hours of learning activities per credit in a semester is required.

viii Community Engagement Projects: These are the courses requiring students to participate in field-based learning/projects generally under the supervision of faculty. The curricular component of 'community engagement and service' will involve activities that would expose students to the socio-economic issues in society so that the theoretical learnings can be supplemented by actual life experiences to generate solutions to real-life problems.

26-30 hours of contact time per credit in a semester along with 13-15 hours of activities such as preparation for community engagement and service, preparation of reports, etc., and independent reading and study with 2 credit courses.

Definition of Credit

- 1 hour Lecture (TH) per week per semester = 1 Credit
- 1 hours Tutorial (TU)per week per semester = 1 Credit
- 2 hours Practical (PR)/Laboratory per week = 1 Credit

- 4 Credit theory courses shall be designed for a minimum of 50 hours of the Teaching-Learning process.
- 3 Credit theory courses shall be designed for a minimum of 40 hours of the Teaching-Learning process.
- Two-credit theory courses shall be designed for a minimum of 25 hours of the Teaching-Learning process
- One credit theory course shall be designed for a minimum of 15 hours of the Teaching-Learning process

Guidelines for Examination Scheme

Theory Examination: The theory examination shall be conducted in two different parts Comprehensive Continuous Evaluation (CCE) and End-Semester Examination (ESE).

Comprehensive Continuous Evaluation (CCE) :

- CCE of 30 marks based on all the Units of course syllabus to be scheduled and conducted at institute level.
- Case studies included under each unit are intended to support applied learning and are part of Comprehensive Continuous Evaluation
- These case studies will be assessed through internal assessment components such as presentations, assignments, or group discussions. They shall not be included in the End-Semester Theory Examination.
- To design a Comprehensive Continuous Evaluation scheme for a theory subject of 40 marks with the specified parameters, the allocation of marks and the structure can be detailed as follows:

Sr.	Parameters	Marks	Coverage of Units
1	Unit Test	12 Marks	Units 1 & Unit 2 (6 Marks/Unit)
2	Assignments / Case Study	12 Marks	Units 3 & Unit 4 (6 Marks/Unit)
3	Seminar Presentation / Open Book Test/ Quiz	06 Marks	Unit 5

- CCE of 15 marks based on all the Units of course syllabus to be scheduled and conducted at institute level. To design a Comprehensive Continuous Evaluation (CCE) scheme for a theory subject of 15 marks with the specified parameters, the allocation of marks and the structure can be detailed as follows:

Sr.	Parameters	Marks	Coverage of Units
1	Unit Test	10 Marks	Units 1 & Unit 2 (5 Marks/Unit)
2	Seminar Presentation / Open Book Test/ Assignments/Case Studies	05 Marks	Units 3 & Unit 4

Format and Implementation of Comprehensive Continuous Evaluation (CCE)

Unit Test

- **Format :** Questions designed as per Bloom's Taxonomy guidelines to assess various cognitive levels (Remember, Understand, Apply, Analyze, Evaluate, Create).
- **Implementation**: Schedule the test after completing Units 1 and 2. Ensure the question paper is balanced and covers key concepts and applications.

  - **Sample Question Distribution**
    - Remembering (2 Marks): Define key terms related to [Topic from Units 1 and 2].
    - Understanding (2 Marks): Explain the principle of [Concept] in [Context].
    - Applying (2 Marks): Demonstrate how [Concept] can be used in [Scenario].
    - Analyzing (3 Marks): Compare & contrast [Two related concepts] from Units 1 and 2.
    - Evaluating (3 Marks): Evaluate the effectiveness of [Theory/Model] in [Situation].

- **Assignments / Case Study** : Students should submit one assignment, or one Case Study Report based on Unit 3 and one assignment or one Case Study Report based on Unit 4.
    - **Format:** Problem-solving tasks, theoretical questions, practical exercises, or case studies that require in-depth analysis and application of concepts.
    - **Implementation:** Distribute the assignments or case study after covering Units 3 and 4.
      Provide clear guidelines and a rubric for evaluation.
- **Seminar Presentation:**
- Format: Oral presentation on a topic from Unit 5, followed by a Q&A session.
- Deliverables: Presentation slides, a summary report in 2 to 3 pages, and performance during the presentation.
- Implementation: Schedule the seminar presentations towards the end of the course. Provide students with ample time to prepare and offer guidance on presentation skills.
- Open Book Test:
- Format: Analytical and application-based questions to assess depth of understanding.
- Implementation: Schedule the open book test towards the end of the course, ensuring it covers critical aspects of Unit 5.
- Quiz :
- Format: Quizzes can help your students practice existing knowledge while stimulating interest in learning about new topic in that course. You can set your quizzes to be completed individually or in small groups.
- Implementation: Online tools and software can be used create quiz. Each quiz is made up of a variety of question types including multiple choice, missing words, true or false etc

- **Example Timeline for conducting CCE:**

- Weeks 1-4 : Cover Units 1 and 2
- Week 5 : Conduct Unit Test (12 marks)
- Weeks 6-8 : Cover Units 3 and 4
- Week 9 : Distribute and collect Assignments / Case Study (12 marks)
- Weeks 10-12 : Cover Unit 5
- Week 13 : Conduct Seminar Presentations or Open Book Test or Quiz (6 marks)

- **Evaluation and Feedback:**

- **Unit Test:** Evaluate promptly and provide constructive feedback on strengths and areas for improvement.
- **Assignments / Case Study:** Assess the quality of submissions based on the provided rubric. Offer feedback to help students understand their performance.
- **Seminar Presentation:** Evaluate based on content, delivery, and engagement during the Q&A session. Provide feedback on presentation skills and comprehension of the topic.
- **Open Book Test**: Evaluate based on the depth of analysis and application of concepts. Provide feedback on critical thinking and problem-solving skills.

End-Semester Examination (ESE)

End-Semester Examination (ESE) of 70 marks written theory examination based on all the unit of course syllabus scheduled by university. Question papers will be sent by the University through QPD (Question Paper Delivery). University will schedule and conduct ESE at the end of the semester.

- **Format and Implementation:**

– **Question Paper Design:** Below structure is to be followed to design an End-Semester Examination (ESE) for a theory subject of 60 marks on all 5 units of the syllabus with questions set as per Bloom's Taxonomy guidelines and 14 marks allocated per unit.

– **Balanced Coverage**: Ensure balanced coverage of all units with questions that assess different cognitive levels of Bloom's Taxonomy: Remember, Understand, Apply, Analyse, Evaluate, and Create. The questions should be structured to cover:

* Remembering: Basic recall of facts and concepts.
* Understanding: Explanation of ideas or concepts.
* Applying: Use of information in new situations.
* Analyzing: Drawing connections among ideas.
* Evaluating: Justifying a decision or course of action.
* Creating: Producing new or original work (if applicable).

– **Detailed Scheme:** Unit-Wise Allocation (14 Marks per Unit): Each unit will have a combination of questions designed to assess different cognitive levels. By following this scheme, you can ensure a comprehensive and fair assessment of students' understanding and application of the course material, adhering to Bloom's Taxonomy guidelines for cognitive skills evaluation.

SYLLABUS

SEMESTER - III

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PCC-301-ITT **:** Operating System

Teaching Scheme	**Credits**	**Examination Scheme**
Theory :3 Hours/Week	03	CCE - 30 Marks End Semester - 70 Marks

Prerequisite : Computer Organization and Data Structure

Course Objectives:

- To understand the fundamentals and functions of modern operating systems.
- To learn the concept process, thread management, and CPU scheduling techniques.
- To understand the concept of process synchronization, concurrency control, and deadlock handling.
- To analyze memory management strategies page replacement algorithms.
- To understand I/O management, disk scheduling, and file systems.

Course Outcomes:

- **CO1:** Understanding the role of Modern Operating Systems.
- **CO2:** Apply the concepts of process and thread scheduling.
- **CO3:** Apply the concept of process synchronization, mutual exclusion and the deadlock.
- **CO4:** Understand and apply the concepts of various memory management techniques.
- **CO5:** Make use of concept of I/O management and File system.

Course Contents

Unit : I	Overview Of Operating System	09 Hours

Operating System Objectives and Functions, The Evolution of Operating Systems, Developments Leading to Modern Operating Systems, Virtual Machines. BASH Shell scripting: Basic shell commands, shell as a scripting language.

Case Study : Automating System Administration Using BASH Shell Scripting

Unit : II	Process Description And Control	09 Hours

Process: Concept of a Process, Process States, Process Description, Process Control (Process creation, Waiting for the process/processes, Loading programs into processes and Process Termination), Execution of the Operating System.

Threads: Processes and Threads, Concept of Multithreading, Types of Threads, Thread programming Using Pthreads.

Scheduling: Types of Scheduling, Scheduling Algorithms, and Thread Scheduling.

Case Study: Hospital Management System

Unit : III	Concurrency Control	09 Hours

Process/thread Synchronization and Mutual Exclusion: Principles of Concurrency, Requirements for Mutual Exclusion, Mutual Exclusion: Hardware Support, Operating System Support (Semaphores and Mutex), Programming Language Support (Monitors).

Classical synchronization problems: Readers/Writers Problem, Producer and Consumer problem, Interprocess communication (Pipes, shared memory: system V).

Deadlock: Principles of Deadlock, Deadlock Modelling, Deadlock Prevention, Deadlock Avoidance, Banker's Algorithm Deadlock detection and recovery, An Integrated Deadlock Strategy, Example: Dining Philosophers Problem.

Case Study : E-Commerce Warehouse System – Producer Consumer Problem

Unit : IV	Memory Management	09 Hours

Memory Management: Memory Management Requirements, Memory Partitioning**:** Fixed Partitioning, Dynamic Partitioning, Buddy System, Relocation, Paging, Page table structure, Segmentation

Virtual Memory: Background, Demand Paging, Page Replacement (FIFO, LRU, Optimal), Allocation of frames, Thrashing

Case Study : Google Chrome Browser and Virtual Memory

Unit : V	Input/Output And File Management	09 Hours

I/O Management and Disk Scheduling: I/O Devices, Organization of the I/O Function, I/O Buffering, Disk Scheduling (FIFO, SSTF, SCAN, C-SCAN, LOOK, C-LOOK).

File Management: Overview-Files and File Systems, File structure. File Organization and Access, File Directories, File Sharing, Record Blocking, Secondary Storage Management.

Case Study: Airport Reservation System

Learning Resources

Text Books:

1. William Stallings, Operating System: Internals and Design Principles, Prentice Hall, 8th Edition,2014, ISBN-10: 0133805913 • ISBN-13: 9780133805918

2. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, Operating System Concepts, John Wiley & Sons ,Inc., 9th Edition,2012, ISBN 978-1-118-06333-0.

Reference Books:

1.Andrew S. Tanenbaum & Herbert Bos, Modern Operating System, Pearson, ISBN-13: 9780133592221, 4th Edition.

2. Tom Adelstein and Bill Lubanovic, Linux System Administration, O'Reilly Media, ISBN- 10: 0596009526, ISBN-13: 978-0596009526.

3. Harvey M. Deitel, Operating Systems, Prentice Hall, ISBN-10: 0131828274, ISBN-13: 978- 0131828278.

4. Thomas W. Doeppner, Operating System in depth: Design and Programming, WILEY, ISBN: 978-0-471-68723-8.

E-Books:

1. https://onlinecourses.swayam2.ac.in/e-learning/preview/ntr26_ed72
2. https://nptel.ac.in/courses/106105214
3. https://www.coursera.org/learn/os-power-user

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PCC-302-ITT : Artificial Intelligence & Machine Learning

Teaching Scheme	Credits	Examination Scheme
Theory :3 Hours/Week	03	CCE - 30 Marks End Semester - 70 Marks

Prerequisite: Programming and Problem Solving, Data Structures, Discrete Mathematics, Probability and Statistics

Course Objectives:

- Understand the fundamental concepts, principles and applications of Artificial Intelligence and Machine Learning.
- Develop problem-solving skills using intelligent agents, search strategies, and knowledge representation techniques.
- Apply data preprocessing methods and supervised machine learning algorithms for predictive modeling and classification tasks.
- Analyze and implement unsupervised and advanced machine learning techniques for pattern discovery, dimensionality reduction and decision-making.
- Evaluate machine learning solutions using appropriate performance metrics and understand ethical, societal and professional aspects of AI-based systems.

Course Outcomes:

- CO1: Analyze AI concepts, intelligent agents, and their real-world applications.
- CO2: Apply search and reasoning methods for AI-based solutions.
- CO3: Build and evaluate machine learning models using supervised learning techniques.
- CO4: Apply unsupervised and advanced machine learning algorithms for pattern discovery and prediction.
- CO5: Analyze machine learning applications, model optimization techniques and ethical implications of AI.

Course Contents

Unit : I	Introduction to Artificial Intelligence and Intelligent Agents	09 Hours

Introduction to AI and AIML, History and Evolution of AI, Types of AI: (ANI, AGI, ASI),AI vs ML vs DL vs Data Science, Intelligent Agents,Agent Environment and PEAS Representation, Types of Agents: (Simple Reflex Agents, Model-Based Agents, Goal-Based Agents, Utility-Based Agents, Learning Agents), Applications of AI in Healthcare, Education, Finance, Agriculture, Industry 4.0 , Challenges and Future of AI.

Case Study : Virtual Assistant (Chatbot), Smart Recommendation System

Unit : II	Problem Solving, Search and Knowledge Representation	09 Hours

Problem Formulation and State Space Representation, Search Strategies, Uninformed Search: (Breadth First Search (BFS), Depth First Search (DFS),Uniform Cost Search), Informed Search: (Heuristic Search,

Greedy Best First Search, A* Search),Local Search and Hill Climbing, Knowledge Representation, Propositional Logic, Predicate Logic, Forward Chaining and Backward Chaining, Introduction to Expert Systems.

Case Study : Route Finding using BFS, DFS and A* Search, Puzzle Solving using Search Algorithms

Unit No: III	Machine Learning Fundamentals and Supervised Learning	09 Hours

Introduction to Machine Learning,ML Workflow and Lifecycle, Data Collection and Dataset Preparation, Data Preprocessing:(Handling Missing Values, Feature Scaling, Encoding Techniques),Training, Validation and Testing, Supervised Learning, Regression:(Linear Regression, Polynomial Regression),Classification:(Logistic Regression, Decision Tree

K-Nearest Neighbors (KNN), Naïve Bayes), Model Evaluation: (Accuracy, Precision,Recall,F1-Score, Confusion Matrix)

Case Study : House Price Prediction using Linear Regression, Spam Email Classification

Unit No: IV	Unsupervised Learning and Advanced Machine Learning Techniques	09 Hours

Introduction to Unsupervised Learning, Clustering Fundamentals, K-Means Clustering, Hierarchical Clustering, DBSCAN, Association Rule Mining, Dimensionality Reduction, Principal Component Analysis (PCA), Feature Selection Techniques, Cluster Evaluation Metrics, Applications of Unsupervised Learning

Case Study : Customer Segmentation using K-Means Clustering, Market Basket Analysis

Unit : V	Advanced Machine Learning, Model Optimization and AI Applications	09 Hours

Support Vector Machines (SVM), Ensemble Learning, Bagging and Boosting, Random Forest, AdaBoost, Introduction to Reinforcement Learning, Hyperparameter Tuning, Cross Validation Techniques, Bias-Variance Tradeoff, Ethical and Responsible AI, AI Applications in Healthcare, Finance, Cybersecurity and Smart Systems.

Case Study : Fraud Detection System, AI-based Healthcare Analytics

Learning Resources:

Text Books:

1.S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Hoboken, NJ: Pearson, 2022.

2.Stephen Marsland, Machine Learning: An Algorithmic Perspective, 2nd Edition, CRC Press.

Reference Books:

1.Elaine Rich, Kevin Knight and Shivashankar B. Nair, Artificial Intelligence, 3rd Edition, McGraw Hill Education.

2.Shalev-Shwartz S, Ben-David S. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press; 2014.

3.C. Müller and S. Guido, *Machine Learning with Python*, 1st ed. Sebastopol, CA: O'Reilly Media, 2016.

4.E. Alpaydin, *Introduction to Machine Learning*, 4th ed. Cambridge, MA: MIT Press, 2020.
E-Books: 1.**https://onlinecourses.nptel.ac.in/e-learning/preview/noc19_cs52** 2.**https://onlinecourses.nptel.ac.in/e-learning/preview/noc22_cs02**

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PCC-303-ITT : Theory of Computation

Teaching Scheme	Credits	Examination Scheme
Theory :3 Hours/Week	03	CCE - 30 Marks End Semester - 70 Marks

Prerequisite : Computer Organization and Data Structure

Course Objectives:

- To introduce mathematical models of computation and formal language theory.
- To study finite automata and regular languages used in language recognition systems.
- To understand context-free grammars and pushdown automata for syntax specification and analysis.
- To study Turing machines as a universal model of computation and computability.
- To analyze computational problems based on decidability and computational complexity.

Course Outcomes:

- **CO1:** Apply concepts of formal languages and finite automata to model computational systems.
- **CO2:** Construct finite automata and regular expressions for language recognition problems.
- **CO3:** Design context-free grammars and pushdown automata for context-free languages.
- **CO4:** Analyze computational models using Turing Machines and computability concepts.
- **CO5:** Classify computational problems based on decidability and computational complexity.

Course Contents

Unit : I	Finite Automata	09 Hours

Basic Concepts: Symbols, Strings, Language, Formal Language.

Finite Automata (FA): Formal definition and notations for FSM, Concept of state transition diagram and transition table for FA, Construction of DFA, NFA, NFA with epsilon moves. Conversion of NFA with epsilon moves to NFA, Conversion of NFA to DFA, and Conversion of NFA with epsilon moves to DFA, Minimization of FA, Equivalence of FAs, and Applications of FA. Finite State Machine with output: Moore and Mealy machines - Definition, Construction, Inter-Conversion.

Case Study : Lexical Analyzer Design in Compiler Construction, Pattern Matching and String Searching Applications.

Unit : II	Regular Expressions and Languages	09 Hours

Definition and Identities of RE, Operators of RE, Equivalence of two regular expressions, Equivalence of regular expressions and regular languages (RL), Conversion of RE to FA using direct method, Conversion of FA to RE using Arden's theorem, Closure properties of RLs, Applications of Regular Expressions.

Case Study : Password and Input Validation Systems, Intrusion Detection and Log Analysis using Pattern Matching

Unit : III	Context-Free Grammars and Language	09 Hours

Grammar: Introduction and Representation, Chomsky Hierarchy, Formal Definition of Regular Grammar (RG), Left Linear Grammar (LLG), Right Linear Grammar (RLG), Conversions: LLG to RLG, RLG to LLG, RG to FA, FA to RG.

Context-Free Grammar (CFG): Definition of CFG, Derivation Tree (Parse Tree), Syntax Trees and Abstract Syntax Trees (AST), Sentential Forms, Leftmost and Rightmost Derivations, Ambiguous and Unambiguous Grammars, Context-Free Languages (CFL), Introduction to Membership Problem and Parsing Concepts.

Case Study : Syntax Analysis in Programming Languages, Expression Parsing and Syntax Tree Construction.

Unit : IV	Pushdown Automata and Turing Machines	09 Hours

Pushdown Automata (PDA) : Introduction to Pushdown Automata, Formal Definition of Pushdown Automata, Instantaneous Description (ID), Transition Diagrams of PDA, Acceptance by Final State, Acceptance by Empty Stack, Equivalence of Acceptance Methods, Design of Pushdown Automata, Equivalence of Context-Free Grammars and Pushdown Automata, Applications of Pushdown Automata.

Turing Machines (TM) : Introduction to Turing Machines, Formal Definition of Turing Machine, Instantaneous Descriptions, Language Acceptance by Turing Machines, Design of Turing Machines, Variants of Turing Machines, Multi-Tape Turing Machines, Universal Turing Machine.

Case Study : Stack-Based Syntax Checking Systems.

Unit : V	Decidability and Computational Complexity	09 Hours

Decidability: Decidable Problems concerning Regular Languages, Decidable Problems concerning Context-Free Languages, Undecidable Problems, Halting Problem.

Pumping Lemma : Regular Languages, Context-Free Languages, Applications of Pumping Lemmas in Language Classification.

Computational Complexity: Measuring Complexity, Complexity Classes P and NP, Examples of Problems in P and NP, Reducibility, Mapping Reducibility, Polynomial Time Reduction, NP-Complete Problems, NP-Hard Problems, Satisfiability (SAT) Problem, Normal Forms for Boolean Expressions, Cook's Theorem, Node Cover Problem, Introduction to Approximation Algorithms and Approximation Ratio.

Case Study : Travelling Salesman Problem and Node Cover Problem.

Learning Resources

Text Books:

1. John C. Martin, Introduction to Language and Theory of Computation, TMH, 3 rd Edition, ISBN: 978-0070660489.

2. Vivek Kulkarni, Theory of Computation, Oxford University Press,ISBN13 : 978-0198084587.

Reference Books:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman, Introduction to Automata Theory Languages and Computation, Addison-Wesley, ISBN 0-201-44124-1.

2. Michael Sipser, Introduction to the Theory of Computation, CENGAGE Learning, 3 rd Edition ISBN-13:978-81-315-2529-6.

3. Daniel Cohen, "Introduction to Computer Theory", Wiley & Sons, ISBN 97881265133454.

E-Books:

1. https://cglab.ca/~michiel/TheoryOfComputation/TheoryOfComputation.pdf
2. https://theory.cs.princeton.edu/complexity/book.pdf

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PCC-304-ITT: Operating System Lab 1

Teaching Scheme	Credits	Examination Scheme
Practical : 2 Hrs /wk	1	TW : 25 Marks PR : 25 Marks

Course Objectives:

- To understand the fundamentals of shell programming and Unix/Linux command-line utilities for file and process management.
- To study process creation, control, and management using system calls.
- To understand thread synchronization mechanisms and apply them to solve classical concurrency problems.
- To study deadlock avoidance and prevention techniques in multi-process systems.
- To implement inter-process communication mechanisms and disk scheduling algorithms for optimizing system performance.

Course Outcomes:

- CO1: Design and implement shell programs to automate file and record management tasks using Unix/Linux system utilities.
- CO2: Implement process creation and control using fork() and wait() system calls.
- CO3: Apply thread synchronization techniques to implement and analyze classical synchronization problems.
- CO4: Analyze deadlock avoidance strategies and implement the algorithm for safe resource allocation.
- CO5: Implement inter-process communication using pipes/FIFOs and disk scheduling to evaluate system and I/O performance.

Guidelines for Instructor's Manual

The faculty member should prepare the laboratory manual for all the experiments and it should be made available to students and laboratory instructor/Assistant.

The instructor's manual should include prologue, university syllabus, conduction & Assessment guidelines, topics under consideration-concept, objectives, outcomes, algorithm written in pseudo language, sample test cases and references. All assignments shall be implemented using C/C++ programming language or Linux bash shell programming as required by the assignment.

Guidelines for Student's Lab Journal

1. The laboratory assignments are to be submitted by students in the form of journals. The Journal consists of prologue, Certificate, table of contents, and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements,Date of Completion, Assessment grade/marks and assessor's sign, Theory Concept, algorithms, printouts of the code written using coding standards, sample test cases etc.)

2. Practical Examination will be based on the term work.

3. Candidate is expected to know the theory involved in the experiment.

4. The practical examination should be conducted if the journal of the candidate is completed in all respects and certified by concerned faculty and head of the department.

5. All the assignment mentioned in the syllabus must be conducted.

Guidelines for Lab /TW Assessment

1. Examiners will assess the term work based on performance of students considering the parameters such as timely conduction of practical assignment, methodology adopted for implementation of practical assignment, timely submission of assignment in the form of handwritten write-up along with results of implemented assignment, attendance etc.

2. Examiners will judge the understanding of the practical performed in the examination by asking some questions related to theory & implementation of experiments he/she has carried out.

3. Appropriate knowledge of usage of software and hardware such as compiler, debugger, coding standards, algorithm to be implemented etc. should be checked by the concerned faculty member(s).

Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The instructor may set multiple sets of assignments and distribute among batches of students. It is appreciated if the assignments are based on real world problems/applications.

All the assignments should be conducted on multicore hardware and 64-bit open-source software.

Guidelines for Practical Examination

Both internal and external examiners should jointly set problem statements for practical examination. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to judge the student‘s understanding of the fundamentals, effective and efficient implementation. The evaluation should be done by both external and internal examiners.

List of Assignments

1. Shell Programming (CO1)

Write a program to implement an address book with options given below:

a) Create address book. b) View address book. c) Insert a record. d) Delete a record e) Modify a record. f) Exit

2. Process Management (CO2)

Process control system calls: The demonstration of FORK and WAIT system calls along with zombie and orphan states. Implement the C program in which main program accepts the integers to be sorted. Main program uses the FORK system call to create a new process called a child process. Parent process sorts the integers using sorting algorithm and waits for child process using WAIT system call to sort the integers using any sorting algorithm. Demonstrate zombie and orphan states.

3: Thread synchronization (CO3)
Thread synchronization using counting semaphores and mutual exclusion using mutex. Application to demonstrate: producer-consumer problem with counting semaphores and mutex.
4. Thread synchronization and Deadlock handling (CO3, CO4)
Deadlock Avoidance Using Semaphores: Implement the deadlock-free solution to Dining Philosophers problem to illustrate the problem of deadlock and/or starvation that can occur when many synchronized threads are competing for limited resources.
5. Deadlock avoidance (CO4)
Implement the C program for Deadlock Avoidance Algorithm: Bankers Algorithm.
6. Interprocess Communication (CO5)
FIFOs: Full duplex communication between two independent processes. First process accepts sentences and writes on one pipe to be read by second process and second process counts number of characters, number of words and number of lines in accepted sentences, writes this output in a text file and writes the contents of the file on second pipe to be read by first process and displays on standard output.
7. Pipes (CO5)
Full duplex communication between parent and child processes. Parent process writes a pathname of a file (the contents of the file are desired) on one pipe to be read by child process and child process writes the contents of the file on second pipe to be read by parent process and displays on standard output.
8. Disk Scheduling (CO5)
Implement the C program for Disk Scheduling Algorithms: SSTF, SCAN, C-Look considering the initial head position moving away from the spindle.
Learning Resources:
Text Books: 1. Das, Sumitabha, UNIX Concepts and Applications, TMH, ISBN-10: 0070635463, ISBN-13: 978-0070635463, 4th Edition. 2. Kay Robbins and Steve Robbins, UNIX Systems Programming, Prentice Hall, ISBN-13: 978-0134424071, ISBN-10: 0134424077, 2nd Edition. 3. Yashwant Kanetkar, UNIX Shell Programming, BPB Publication.
V Lab: 1.Operating Systems Lab (IIIT Hyderabad): https://os-iiith.vlabs.ac.in/ 2.Linux Lab (Bash Shell Programming): https://linuxjourney.com/

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PCC-305-ITT: Artificial Intelligence & Machine Learning Lab 2

Teaching Scheme	Credits	Examination Scheme
Practical : 4 Hrs /wk	2	TW : 25 Marks PR : 25 Marks

Course Objectives:

- Develop practical skills in implementing Artificial Intelligence techniques for problem-solving and decision-making.
- Apply search, reasoning and knowledge representation methods to solve real-world AI problems.
- Perform data preprocessing, visualization and supervised machine learning for predictive analytics.
- Implement unsupervised and advanced machine learning algorithms for pattern discovery and model improvement.
- Design and evaluate AI-based solutions using appropriate tools, performance metrics and optimization techniques.

Course Outcomes:

- CO1: Implement intelligent agents, search algorithms and knowledge representation techniques for AI-based problem solving.
- CO2: Apply data preprocessing, exploratory data analysis and supervised learning algorithms for prediction and classification tasks.
- CO3: Develop and evaluate machine learning models using appropriate performance metrics and validation techniques.
- CO4: Implement clustering, association analysis and dimensionality reduction techniques for pattern discovery in datasets.
- CO5: Build optimized AI/ML solutions using advanced learning algorithms and apply them to real-world applications.

Guidelines for Instructor's Manual

The faculty member should prepare the laboratory manual for all the experiments and it should be made available to students and laboratory instructor/Assistant.

The instructor's manual should include prologue, university syllabus, conduction & Assessment guidelines, topics under consideration-concept, objectives, outcomes, algorithm written in pseudo language, sample test cases and references. All assignments shall be implemented using Python and relevant open-source AI/ML libraries.

From PR2 and PR3, PR7,PR8 : Students shall implement any one search algorithm pair from the following BFS & DFS or Greedy Best First Search and A* Search algorithms. The choice shall be made at the beginning of the term and recorded in the journal.

Guidelines for Student's Lab Journal

1. The laboratory assignments are to be submitted by students in the form of journals. The Journal consists of prologue, Certificate, table of contents, and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements,Date of Completion, Assessment grade/marks and assessor's sign, Theory Concept, algorithms, printouts of the code written using coding standards, sample test cases etc.)
2. Practical Examination will be based on the term work.
3. Candidate is expected to know the theory involved in the experiment.
4. The practical examination should be conducted if the journal of the candidate is completed in all respects and certified by concerned faculty and head of the department.
5. All the assignment mentioned in the syllabus must be conducted.

(From PR2 and PR3, any one search algorithm experiment is to be completed. From PR7 and PR8, any one classification experiment is to be completed).

Guidelines for Lab /TW Assessment

1. Examiners will assess the term work based on performance of students considering the parameters such as timely conduction of practical assignment, methodology adopted for implementation of practical assignment, timely submission of assignment in the form of handwritten write-up along with results of implemented assignment, attendance etc.
2. Examiners will judge the understanding of the practical performed in the examination by asking some questions related to theory & implementation of experiments he/she has carried out.
3. Appropriate knowledge of usage of software and hardware such as compiler, debugger, coding standards, algorithm to be implemented etc. should be checked by the concerned faculty member(s).

Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The instructor may set multiple sets of assignments and distribute among batches of students. It is appreciated if the assignments are based on real world problems/applications.

All the assignments should be conducted on multicore hardware and 64-bit open-source software.

Real-world datasets from public repositories such as Kaggle, UCI Machine Learning Repository, or Open Government Data sources should be used wherever applicable.

Guidelines for Practical Examination

Both internal and external examiners should jointly set problem statements for practical examination. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to judge the student‘s understanding of the fundamentals, effective and efficient implementation. The evaluation should be done by both external and internal examiners.

List of Assignments

PART A

PR1: Intelligent Agents and PEAS Representation (CO1)

Design and implement a simple intelligent agent for a real-world application such as a chatbot, smart home controller or recommendation assistant. Demonstrate the PEAS (Performance Measure, Environment, Actuators, Sensors) representation and agent behavior.

PR2: Uninformed Search Algorithms (CO1, CO2)

Implement Breadth First Search (BFS) and Depth First Search (DFS) algorithms for solving route-finding or maze navigation problems. Compare their performance based on path length, nodes explored and execution time.

PR3: Informed Search Algorithms (CO1, CO2)

Implement Greedy Best First Search and A* Search algorithms for shortest path finding. Analyze optimality, completeness, and efficiency of both approaches.

PR4: Knowledge Representation and Expert System (CO1, CO2)

Develop a rule-based expert system using propositional logic and implement Forward Chaining and Backward Chaining mechanisms for intelligent decision making.

PART B

PR5: Data Preprocessing and Exploratory Data Analysis (CO2)

Download a standard dataset (Heart Disease, Diabetes or Titanic dataset) and perform:

- Data cleaning
- Missing value handling
- Feature scaling
- Encoding techniques
- Statistical analysis
- Data visualization using histograms and boxplots

Split the dataset into training and testing sets.

PR6: Regression Techniques (CO2, CO3)

Implement Linear Regression and Polynomial Regression on a suitable dataset such as California Housing or House Price Prediction dataset.

Evaluate model performance using:

- Mean Squared Error (MSE)
- Mean Absolute Error (MAE)
- R^2 Score

Compare the results obtained by both techniques.

PR7: Classification using Logistic Regression and K-Nearest Neighbors (CO2, CO3)

Perform data preprocessing and implement:

- Logistic Regression
- K-Nearest Neighbors (KNN)

Evaluate the models using:

- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix

Analyze and compare the results.

PR8: Classification using Decision Tree and Naïve Bayes (CO2, CO3)

Implement Decision Tree and Naïve Bayes classifiers on a suitable dataset.

Compare classifier performance using standard evaluation metrics and identify the most suitable model for the selected dataset.

PART C

PR9: Clustering Techniques (CO4)

Implement:

- K-Means Clustering
- Hierarchical Clustering

Use the Elbow Method and Dendrogram for determining the optimal number of clusters. Analyze the generated clusters using a real-world dataset.

PR10: Dimensionality Reduction using Principal Component Analysis (PCA) (CO4)

Apply PCA on a high-dimensional dataset.

Compare:

- Number of features before and after PCA
- Visualization of reduced dimensions
- Impact on classification performance

PR11: Advanced Machine Learning and Model Optimization (CO5)

Implement and compare:

- Support Vector Machine (SVM)
- Random Forest
- AdaBoost

Perform:

- Cross Validation
- Hyperparameter Tuning using Grid Search

Evaluate and compare model performance before and after optimization.

Mini Project

PR12: AI/ML-Based Real-World Application (CO1–CO5)

- Disease Prediction System
- Fraud Detection System
- Customer Segmentation
- Recommendation System

- Student Performance Prediction
- Crop Yield Prediction

The project should include:

- Problem Definition
- Dataset Collection
- Data Preprocessing
- Model Development
- Performance Evaluation
- Model Optimization
- Result Analysis and Documentation

Learning Resources:

Text Books:

1.S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Hoboken, NJ: Pearson, 2022.

2.Stephen Marsland, Machine Learning: An Algorithmic Perspective, 2nd Edition, CRC Press.

Reference Books:

1.A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd ed. Sebastopol, CA: O'Reilly Media, 2023.

2.Shalev-Shwartz S, Ben-David S. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press; 2014.

3.C. Müller and S. Guido, *Machine Learning with Python*, 1st ed. Sebastopol, CA: O'Reilly Media, 2016.

4.S. Raschka and V. Mirjalili, Python Machine Learning, 3rd ed. Birmingham, UK: Packt Publishing, 2019.

V Lab:

1.IIIT Hyderabad, "Artificial Intelligence Virtual Lab," Virtual Labs, IIIT Hyderabad. Available: https://ai1-iiith.vlabs.ac.in

2.IIT Roorkee, "Machine Learning Laboratory," Virtual Labs — A Ministry of Education, Govt. of India Initiative.

3.Available: https://ml-iitr.vlabs.ac.in

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PEC-321A-IT : Cloud Computing

Teaching Scheme	Credits	Examination Scheme
Theory :3 Hours/Week	03	CCE - 30 Marks End Semester - 70 Marks

Prerequisite Courses: Computer Networks

Companion Course, : Programming Elective Course Lab

Course Objectives:

- **To understand** the fundamental concepts, architectures, and service models of cloud computing.
- **To learn** virtualization and containerization technologies used in cloud environments.
- **To develop** the ability to deploy, manage, and utilize cloud platforms and services.
- **To understand** cloud security, privacy, and risk management techniques for protecting cloud resources.
- **To explore** modern cloud technologies, DevOps practices, and emerging trends in cloud computing.

Course Outcomes:

- **To explain** the concepts, characteristics, architectures, and service models of cloud computing.
- **To demonstrate** virtualization and containerization technologies used for cloud infrastructure and application deployment.
- **To deploy and manage** applications, storage, and services using cloud computing platforms.
- **To apply** cloud security mechanisms and risk management practices to protect cloud-based resources and data.
- **To analyze** modern cloud technologies, DevOps practices, and emerging trends in cloud computing.

Course Contents

Unit No: I	Fundamentals of Cloud Computing	09 Hours

Introduction to Cloud Computing, Need and Evolution of Cloud Computing, Characteristics of Cloud Computing, Advantages and Limitations, Cloud Service Models: IaaS, PaaS, SaaS, Cloud Deployment **Models:** Public, Private, Hybrid, Community Cloud, Cloud Architecture, Cloud Storage and Data Management, Overview of Cloud Providers.

Case Study : Migration of a College ERP System to Cloud

Unit No: II	Virtualization and Containerization	09 Hours

Introduction to Virtualization, Benefits and Applications of Virtualization, Types of Virtualization, **Hypervisors:** Type-I and Type-II, Virtual Machines, Introduction to Containers, Docker Fundamentals, Comparison of Virtual Machines and Containers, Basic Container Deployment, Introduction to Kubernetes.

Case Study : VMware and Docker in Enterprise Computing

Unit No: III	Cloud Platforms and Services	09 Hours
Introduction to AWS, Amazon EC2 and S3 Concepts, Introduction to Microsoft Azure, Azure Virtual Machines and Storage, Google Cloud Platform Overview, Open-Source Cloud Platforms, Cloud Databases, Cloud-Based Application Deployment, Cloud Services for AI, IoT and Data Analytics **Case Study :** Cloud Services Used in Google Workspace.		
Unit No: IV	Cloud Security and Risk Management	09 Hours
Introduction to Cloud Security, Security Challenges in Cloud Computing, Data Security and Privacy, Identity and Access Management, Authentication and Authorization, Cloud Threats and Vulnerabilities, Risk Management Process, Business Continuity and Disaster Recovery, Security Best Practices, Compliance and Governance Basics. **Case Study :** Analysis of a Cloud Data Breach Incident		
Unit No: V	Modern Cloud Environment and Emerging Technologies	09 Hours
Edge Computing and Fog Computing, Serverless Computing, DevOps and Cloud, Continuous Integration and Continuous Deployment (CI/CD), Infrastructure as Code, Green Cloud Computing, Multi-Cloud and Hybrid Cloud, AI and Machine Learning in Cloud, Future Trends in Cloud Computing. **Case Study :** Cloud Adoption Journey of Netflix		

Learning Resources

Text Books:

1. A. Srinivasan, J. Suresh, Cloud Computing: A Practical Approach for Learning and Implementation, Pearson Education, ISBN: 978-81-317-7651-3.

2. Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, Mastering Cloud Computing: Foundations and Applications Programming, McGraw Hill Education, ISBN: 978-1-25-902995-0.

Reference Books:

1. Dr. Kris Jamsa, Cloud Computing: SaaS, PaaS, IaaS, Virtualization, Business Models, Mobile, Security and More, Jones & Bartlett Learning/Wiley, ISBN: 978-1-4496-8358-7.

2. Anthony T. Velte, Toby J. Velte, Robert Elsenpeter, Cloud Computing: A Practical Approach, McGraw-Hill Education, ISBN: 978-0-07-162694-1.

3. Thomas Erl, Ricardo Puttini, Zaigham Mahmood, Cloud Computing: Concepts, Technology & Architecture, Pearson Education, ISBN: 978-0-13-338752-0.

4. Tim Mather, Subra Kumaraswamy, Shahed Latif, Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, O'Reilly Media, ISBN: 978-0-596-80276-9.

E-Books:

1.https://dhoto.lecturer.pens.ac.id/lecture_notes/internet_of_things/CLOUD%20COMPUTING%20Principles%20and%20Paradigms.pdf

2.https://www.lpude.in/SLMs/Master%20of%20Computer%20Applications/Sem_2/DECAP470_CLOUD_COMPUTING.pdf

3.https://studytm.wordpress.com/wp-content/uploads/2014/03/hand-book-of-cloud-computing.pdf

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PEC 322A-IT: Cloud Computing Lab

Teaching Scheme	Credits	Examination Scheme
Practical :2 Hours/Week	03	Term Work - 25 Marks Practical : 25 Marks

Prerequisite Courses, if any : Computer Networks

Course Objectives:

1. To understand cloud computing services, platforms, and deployment models through practical implementation.
2. To develop skills in virtualization, containerization, and cloud resource management.
3. To deploy and manage cloud-based applications, storage, and computing resources.
4. To apply cloud security mechanisms and best practices while designing cloud-based solutions.

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

CO1: **To utilize** cloud platforms, cloud services, and storage resources for various applications.

CO2: **To implement** virtualization and containerization technologies for creating and managing cloud environments.

CO3: **To deploy and manage** applications and services in cloud environments using appropriate tools and techniques.

CO4: **To apply** cloud security practices and develop cloud-based solutions for real-world problems..

Guidelines for Instructor's Manual

The instructor‘s manual is to be developed as a reference and hands-on resource. It should in clude prologue (about University/program/ institute/ department/foreword/ preface), curriculum of the course, conduction and Assessment guidelines, topics under consideration, concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

Guidelines for Student's Lab Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Comple tion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor’s sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if appli cable), mathematical model (if applicable), conclusion/analysis). Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution

towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal must be avoided. Use of DVD containing students' programs maintained by Laboratory In-charge is highly encouraged. For reference, one or two journals may be maintained with program prints in the Laboratory.

Guidelines for Lab /TW Assessment

Continuous assessment of laboratory work should be based on the overall performance of a student on laboratory assignments. Each Laboratory assignment assessment will assign grade/marks based on parameters, such as timely completion, performance, innovation, efficient codes, and punctuality.

Guidelines for Practical Examination

Problem statements must be decided jointly by the internal examiner and external examiner. During practical assessment, maximum weightage should be given to satisfactory implementation of the problem statement. Relevant questions may be asked at the time of evaluation to test the student's understanding of the fundamentals, effective and efficient implementation. This will encourage trans parent evaluation and a fair approach, and hence will not create any uncertainty or doubt in the minds of the students. So, adhering to these principles will consummate our team efforts to the promising start of students' academics.

Guidelines for Laboratory Conduction

The instructor is expected to frame laboratory assignments by considering prerequisite knowledge of computer networks, operating systems, and cloud computing fundamentals, along with current industry practices and emerging cloud technologies. Assignments should focus on hands-on implementation of cloud resource provisioning, virtualization, containerization, cloud storage, application deployment, monitoring, and security using platforms such as AWS, Microsoft Azure, or Google Cloud Platform. Different cloud-based case studies and mini-projects may be assigned to different student groups. The laboratory should emphasize practical learning, problem-solving, and real-world cloud application development through continuous assessment based on laboratory performance, documentation, demonstrations, mini-projects, and viva-voce.

List of Assignments Group A

1.Cloud Platform Setup and Exploration - CO1

Create an account on AWS, Microsoft Azure, or Google Cloud Platform. Explore the available cloud services such as compute, storage, networking, and databases. Prepare and document the cloud environment setup..

2. Cloud Storage Management - CO1

Create cloud storage buckets/containers and perform operations such as uploading, downloading, sharing, and managing files using cloud storage services.

3. Virtualization using VirtualBox/VMware - CO2

Install VirtualBox or VMware Workstation and create virtual machines with different operating systems. Configure networking and resource allocation for virtual machines.

4. File Sharing Between Virtual Machines - CO2

Configure network connectivity between virtual machines and implement file transfer using shared folders or network-based file sharing mechanisms.

5. Docker-Based Web Application Deployment - CO2, CO3

Create a Docker image for a simple web application and deploy the application using Docker containers. Verify application accessibility through a web browser.

6. Static Website Hosting on Cloud - CO1, CO3

Host a static website using Amazon S3, Azure Blob Storage, or Google Cloud Storage and configure public access for website hosting.

7. Cloud Monitoring and Logging - CO3, CO4

Configure monitoring and logging services for cloud resources. Generate alerts based on CPU utilization, memory utilization, and storage usage.

8. Mini Project: Cloud-Based Application Deployment - CO1, CO2, CO3, CO4

Design and deploy a cloud-based application using cloud storage, virtual machines or containers, security mechanisms, IAM policies, and monitoring services. Demonstrate the working solution and submit a project report.

Text Books:

1. **A. Srinivasan, J. Suresh**, Cloud Computing: A Practical Approach for Learning and Implementation, Pearson Education, ISBN: 978-81-317-7651-3.

2. **Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi**, Mastering Cloud Computing: Foundations and Applications Programming, McGraw Hill Education, ISBN: 978-1-25-902995-0.

Reference Books:

1. **Anthony T. Velte, Toby J. Velte, Robert Elsenpeter**, Cloud Computing: A Practical Approach, McGraw-Hill Education, ISBN: 978-0-07-162694-1.

2. **Thomas Erl, Ricardo Puttini, Zaigham Mahmood**, Cloud Computing: Concepts, Technology & Architecture, Pearson Education, ISBN: 978-0-13-338752-0.

3. **Tim Mather, Subra Kumaraswamy, Shahed Latif**, Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, O'Reilly Media, ISBN: 978-0-596-80276-9.

4. **Dr. Kris Jamsa**, Cloud Computing: SaaS, PaaS, IaaS, Virtualization, Business Models, Mobile, Security and More, Jones & Bartlett Learning, ISBN: 978-1-4496-8358-7.

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PEC-321B-IT: Web Application Development

Teaching Scheme	Credits	Examination Scheme
Theory :3 Hours/Week	03	CCE - 30 Marks End Semester - 70 Marks

Prerequisite Courses: Programming Fundamentals, Database Management Systems, Computer Network

Companion Course: Advanced Database Management system, Design Thinking

Course Objectives

- To understand the architecture, technologies, and standards involved in modern web application development.
- To design and develop responsive and interactive web interfaces.
- To develop Single Page Applications (SPAs) using modern front-end frameworks and component-based architecture.
- To build full-stack web applications using server-side technologies and database integration.
- To deploy and secure web applications using cloud platforms, version control systems, and DevOps practices.

Course Outcomes:

- **CO1:** Design responsive and accessible web pages using HTML5, CSS3, Bootstrap, and web standards.
- **CO2:** Develop interactive client-side applications using JavaScript, JSON, AJAX, jQuery, and modern scripting techniques.
- **CO3:** Build Single Page Applications (SPAs) using TypeScript and Angular framework.
- **CO4:** Develop RESTful web services and full-stack web applications using Node.js, Express.js, and MongoDB.
- **CO5:** Deploy, manage, and secure web applications using cloud services, version control tools, and DevOps practices.

Course Contents		
Unit No: I	Foundations of Web Technologies and Responsive Design	09 Hours
Introduction to Web Technologies : Internet and Web Architecture, Client-Server Architecture, HTTP and HTTPS Protocols, Web Standards and Accessibility. **HTML5 :**Basic Structure of HTML Document, Text Formatting, Hyperlinks, Lists and Tables, Images and Multimedia, Forms and Validation ,Semantic Elements .**CSS3:** Selectors, Box Model, Positioning, Flexbox, Grid Layout, Animations and Transitions **Bootstrap :** Responsive Design Concepts, Grid System, Components Utilities **Case Study :** Responsive College Website Portal: 1. Design a responsive website for a college using HTML5, CSS3, and Bootstrap. 2. Features: Home, About, Courses, Admissions, Contact.		
Unit No: II	Client-Side Scripting and Interactive Web Applications	09 Hours
JavaScript Fundamentals : Variables and Data Types, Operators and Expressions, Functions, Arrays and Objects, Scope and Closures. **DOM and Event Handling :** DOM Manipulation, Event Handling, Form Validation. **Advanced JavaScript :**ES6 Features ,Arrow Functions, Template Literals ,Callbacks, Promises , Async/Await ,Error Handling. **AJAX and JSON :** JSON Objects, AJAX Concepts ,Fetch API. **jQuery:** Selectors, Events ,Effects , DOM Manipulation, AJAX using jQuery **Case Study :** Weather Information Dashboard, Interactive Online Quiz System		
Unit No: III	Front-End Frameworks and Single Page Applications	09 Hours
Introduction to SPA : Single Page Application Concepts, MVC Architecture, Component-Based Development. **Typescript :** Data Types Classes and Interfaces, Modules. Functions **Angular Framework:** Angular Architecture, Components, Templates, Data Binding, Directives, Pipes, Services and Dependency Injection, Forms and Validation, Routing and Navigation **Front-End Project Development :**Angular CLI,Project Structure, Build and Deployment. **Case Study :** Angular-Based E-Commerce Front-End, Task Management Application		
Unit No: IV	Server-Side Development and Database Integration	09 Hours
Node.js: Introduction, Node Package Manager (NPM),Modules, File System Operations. **Express.js :** Express Framework, Routing, Middleware Request and Response Handling **RESTful Web Services :** REST Architecture, CRUD Operations, API Development **MongoDB :**Introduction to NoSQL, Collections and Documents, CRUD Operations, Querying Data. **Mongoose :**Schema Design ,Data Validation ,Database Connectivity. **Authentication and Authorization :**Session Management,JWT Basics **API Testing :** Postman Tool **Case Study:** Online Book Store API, Student Information Management System.		
Unit No: V	Mobile Web Development, Cloud Deployment and Security	09 Hours
Mobile Web Development :jQuery Mobile, Mobile UI Components, Mobile Navigation, Mobile Forms and Layouts **Progressive Web Applications (PWA) :** Introduction, Service Workers, Web App Manifest**Version Control and Deployment:** Git Fundamentals ,GitHub ,Branching and Merging **Cloud Computing :**Cloud Service Models, AWS Fundamentals ,EC2 and S3 Services ,Application Deployment **Web Application Security :**Authentication and Authorization ,OWASP Top 10 Overview, Secure Coding		

Practices **Introduction to DevOps :**CI/CD Concepts, Docker Fundamentals

Case Study : Progressive Web Application (PWA) for Event Management, Cloud Deployment of Full-Stack Application

Learning Resources

Text Books:

1.Terry Felke-Morris, *Web Development and Design Foundations with HTML5*, Pearson Education.

2.Amos Q. Haviv, *Full Stack Development with MongoDB, Express, Angular and Node.js (MEAN Stack)*, Packt Publishing.

REFERENCE BOOKS:

1. Jon Duckett, *HTML and CSS: Design and Build Websites*, Wiley.

2. Jon Duckett, *JavaScript and JQuery: Interactive Front-End Web Development*, Wiley.

3. Ethan Brown, *Web Development with Node and Express*, O'Reilly Media.

4. Shyam Seshadri and Brad Green, *Angular: Up and Running*, O'Reilly Media.

5. Kristina Chodorow, *MongoDB: The Definitive Guide*, O'Reilly Media.

6. Kogent Learning Solutions Inc., *HTML5 Black Book*, Dreamtech Press.

E-Books:

1. MDN Web Docs – Learn Web Development :https://developer.mozilla.org/en-US/docs/Learn

2. Eloquent JavaScript (3rd Edition): https://eloquentjavascript.net/

3. Web Technologies (SWAYAM): Web Technologies (SWAYAM)

4. Web-Designing and Multimedia Technology (SWAYAM) : Web-Designing and Multimedia Technology (SWAYAM)

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PEC-322B-IT: Web Application Development Lab

Teaching Scheme	Credits	Examination Scheme
Practical :2 Hours/Week	03	Term Work - 25 Marks Practical : 25 Marks

Prerequisite Courses : Programming and Problem-Solving concepts

Course Objectives:

- To understand the architecture, technologies, and standards involved in modern web application development.
- To design and develop responsive and interactive web interfaces using HTML5, CSS3, Bootstrap, and JavaScript.
- To develop Single Page Applications (SPAs) using modern front-end frameworks and component-based architecture.
- To build full-stack web applications using server-side technologies, RESTful APIs, and database integration.
- To deploy and secure web applications using cloud platforms, version control systems, and DevOps practices.

Course Outcomes:

- **CO1:** Design responsive and accessible web pages using HTML5, CSS3, Bootstrap, and web standards.
- **CO2:** Develop interactive client-side applications using JavaScript, JSON, AJAX, jQuery, and modern scripting techniques.
- **CO3:** Build Single Page Applications (SPAs) using Typescript and Angular framework.
- **CO4:** Develop RESTful web services and full-stack web applications using Node.js, Express.js, and MongoDB.
- **CO5:** Deploy, manage, and secure web applications using cloud services, version control tools, and DevOps practices.

Guidelines for Instructor's Manual

Lab Assignments: Following is a list of suggested laboratory assignments for reference. Laboratory Instructors may design a suitable set of assignments for their respective courses at their level. Beyond curriculum assignments, the mini-project is also included as a part of laboratory work. The Inclusion of few optional assignments that are intricate and/or beyond the scope of curriculum will surely be the value addition for the students and it will satisfy the intellectuals within the group of the learners and will add to the perspective of the learners. For each laboratory assignment, it is essential for students to draw/write/generate flowchart, algorithm, test cases, mathematical model, Test data set and comparative/complexity analysis (as applicable).

Guidelines for Student's Lab Journal

Program codes with sample output of all performed assignments are to be submitted as softcopy. Use of DVD or similar media containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journals may be avoided. Submission of journal/ term work in the form of softcopy is desirable and appreciated.

Guidelines for Lab /TW Assessment

Term work is continuous assessment that evaluates a student's progress throughout the semester. Term work assessment criteria specify the standards that must be met and the evidence that will be gathered to demonstrate the achievement of course outcomes. Categorical assessment criteria for the term work should establish unambiguous standards of achievement for each course outcome. They should describe what the learner is expected to perform in the laboratories or on the fields to show that the course outcomes have been achieved. It is recommended to conduct an internal monthly practical examination as part of continuous assessment.

Guidelines for Practical Examination

Internal and external examiners should jointly prepare practical assignments based on the syllabus. During the assessment, the examiner should focus on:

a. The student's ability to explain the implemented code and underlying logic.

b. Understanding of core concepts

c. Answering supplementary questions to assess advanced learning.

Guidelines for Laboratory Conduction

Following is a list of suggested laboratory assignments for reference. Laboratory Instructors may design a suitable set of assignments for respective courses at their level. Beyond curriculum assignments and mini-project may be included as a part of laboratory work. The instructor may set multiple sets of assignments and distribute among batches of students. It is appreciated if the assignments are based on real world problems/applications. For each laboratory assignment, it is essential for students to draw/write/generate flowchart, algorithm, test cases, mathematical model, Test data set and comparative/complexity analysis (as applicable). Batch size for practical and tutorials may be as per guidelines of authority.

List of Assignments

1 Modern Responsive Web Design –CO1

Create a responsive university/e-commerce dashboard using HTML5, CSS3, Bootstrap 5, and CSS Grid/Flexbox. Perform accessibility testing and responsive validation using browser developer tools.

2: JavaScript, JSON and REST APIs – CO2

Develop a student registration portal using JavaScript ES6+, JSON, Local Storage, and Form Validation. Consume a public REST API using Fetch API/AJAX and display data dynamically.

3: Angular Fundamentals –CO3

Create an Angular application demonstrating: Components, Data Binding, Directives, Pipes Implement Angular Routing for navigation between pages.
4: Angular Forms and Services – CO3
Develop User Registration and Login using Reactive Forms Create Angular Services and consume REST APIs using HttpClient.
5 : Node.js and Express.js –CO4
Develop a Node.js application serving static and dynamic content.Create RESTful APIs using Express.js for CRUD operations.
6. MongoDB Integration –CO4
Design MongoDB schemas using Mongoose. Integrate MongoDB with Express.js and perform CRUD operations.
7. Docker and GitHub– CO5
Containerize the full-stack application using Docker. Manage source code using Git and GitHub.
8: Mobile Website using jQuery Mobile –CO5
Create a simple mobile-friendly website using jQuery Mobile. Include: 1. Multiple mobile pages 2. Navigation menu 3.Forms and buttons 4.Responsive mobile layout
Text Books: 1. Terry Felke-Morris, *Web Development and Design Foundations with HTML5*, Pearson Education. 2. Amos Q. Haviv, *Full Stack Development with MongoDB, Express, Angular and Node.js (MEAN Stack)*, Packt Publishing.
Reference Books: 1. Jon Duckett, *HTML and CSS: Design and Build Websites*, Wiley. 2. Jon Duckett, *JavaScript and JQuery: Interactive Front-End Web Development*, Wiley. 3. Ethan Brown, *Web Development with Node and Express*, O'Reilly Media. 4. Shyam Seshadri and Brad Green, *Angular: Up and Running*, O'Reilly Media. 5. Kristina Chodorow, *MongoDB: The Definitive Guide*, O'Reilly Media. 6.Kogent Learning Solutions Inc., *HTML5 Black Book*, Dreamtech Press.

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PEC-321C-IT: Image Processing Computer Vision

Teaching Scheme	Credits	Examination Scheme
Theory :3 Hours/Week	03	CCE - 30 Marks End Semester - 70 Marks

Prerequisite Courses, Computer Graphics

Course Objectives:

- To understand the fundamentals of digital image formation, sensing mechanisms, sampling, quantization, and the core concepts of computer vision.
- To understand and implement various image enhancement techniques in both spatial and frequency domains, including gray-level transformations, histogram manipulation, and filtering operations.
- To understand the principles of image segmentation techniques and analyze various image and video compression standards.
- To study state-of-the-art object recognition frameworks and learn modern motion analysis and tracking techniques for video data.
- To explore advanced computer vision applications and understand how hybrid intelligence systems like fuzzy-neural algorithms handle uncertainty in real-world visual data.

Course Outcomes:

- **CO1:** Explain the fundamental concepts of digital image processing, sampling, and quantization, and identify relevant applications of computer vision.
- **CO-2:** Apply spatial and frequency domain filtering techniques to enhance image quality, remove noise, and sharpen structural details.
- **CO-3:** Apply segmentation algorithms to extract object boundaries and utilize lossy/lossless compression techniques to reduce image and video data storage.
- **CO-4:** Implement deep learning object detection models like YOLO and apply motion tracking algorithms to extract temporal information from video sequences.
- **CO-5:** Analyze and design intelligent vision systems for real-world applications including biometric recognition, OCR, and automated surveillance networks.

Course Contents

Unit No: I	Introduction to Image Processing and Computer Vision	09 Hours

Digital Image fundamentals, Image Sensing and acquisition, Sampling and Quantization, Image formation models, Overview of Computer Vision, Applications of Image processing and Computer Vision. Fundamental concepts of digital image processing, sampling, and quantization, and identify relevant applications of computer vision.

Case Study : Smart Toll Plaza Automated Number Plate Recognition (ANPR) System

Unit No: II	Image Enhancement	09 Hours

Image enhancement in spatial domain, Basic grey level Transformations, Histogram Processing Techniques, Spatial Filtering, Image smoothing and Image Sharpening, Image enhancement process in frequency domain, Low pass filtering, High pass filtering.

Case Study : Medical X-Ray and MRI Contrast Optimization

Unit No: III	Image Segmentation and Compression	09 Hours

Point, line and edge detection, Thresholding, Regions Based segmentation, Edge linking and boundary detection. Fundamental of redundancies, Basic Compression Methods: Huffman Coding, Concept of Discrete cosine transform, JPEG,MPEG Compression Standard

Case Study : Traffic Monitoring and Automated Enforcement System (Segmentation Domain)

Unit No: IV	Object Recognition and Motion Estimation	09 Hours

Object Recognition techniques: Viola-Jones, Yolo, Deep learning algorithms for Object Recognition. Optical Flow, Gaussian Mixture Model (GMM), Structure of Motion, Motion Estimation.

Case Study : Smart Traffic Intersection Monitoring and Speed Enforcement

Unit No: V	Applications of Image Processing and Computer vision	09 Hours

Review of Computer Vision applications; Fuzzy-Neural algorithms for computer vision applications

Face Recognition, Facial Expression Recognition, Optical Character Recognition, Automated Video Surveillance

Case Study : Smart Airport Contactless Immigration Clearing Gate

Learning Resources

Text Books:

1. Gonzalez & Woods, “Digital Image Processing”, Pearson Education, 3 rd Edition, 2008
2. S Sridhar, “Digital Image Processing”, Oxford University Press, 2nd Edition.

Reference Books:

1. Jain Anil K., “Fundamentals Digital Image Processing”, Prentice Hall India, 4 th Edition.
2. Milan Sonka, Vaclav Hlavav, Roger Boyle, “Image Processing, Analysis and Machine Vision”, Thomson Learning, 2 nd Edition., 2001.
3. Pratt W.K, “Digital Image Processing”, John Wiley & Sons, 3 rd Edition, 2007.
4. Jayaraman. S, Veerakumar. T, “Digital Image Processing”, McGraw Hill Education, 2nd Edition.

Links to online SWAYAM/NPTEL Courses:

1. NPTEL Course on Digital Image Processing: https://onlinecourses.nptel.ac.in/e-learning/preview/noc22_ee116
2. NPTEL course on Computer Vision : https://nptel.ac.in/courses/106106090

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PEC-322B-IT : Image Processing and Computer Vision Lab

Teaching Scheme	Credits	Examination Scheme
Practical :2 Hours/Week	02	Term Work - 25 Marks Practical : 25 Marks

Prerequisite Courses : Programming and Problem-Solving concepts, Python Programming

Course Objectives:

- Develop proficiency in digital image representation, analysis, and enhancement techniques for improving image quality.
- Apply feature extraction and segmentation methods to derive meaningful information from images.
- Implement feature matching, motion analysis, and object tracking techniques for visual scene understanding.
- Design practical computer vision applications involving OCR, recognition, and authentication systems.
- Build intelligent vision solutions using deep learning and design thinking principles to address real-world challenges.

Course Outcomes:

- **CO1: Analyze** digital images using image representation and enhancement techniques.
- **CO2: Apply** image feature extraction and segmentation techniques to identify meaningful image regions and objects.
- **CO3: Implement** feature matching and motion analysis techniques for visual scene understanding.
- **CO4: Develop** computer vision applications for document analysis and biometric authentication.
- **CO5: Design** AI-enabled computer vision solutions for real-world applications.

Guidelines for Instructor's Manual

Lab Assignments: Following is a list of suggested laboratory assignments for reference. Laboratory Instructors may design a suitable set of assignments for their respective courses at their level. Beyond curriculum assignments, the mini-project is also included as a part of laboratory work. The Inclusion of few optional assignments that are intricate and/or beyond the scope of curriculum will surely be the value addition for the students and it will satisfy the intellectuals within the group of the learners and will add to the perspective of the learners. For each laboratory assignment, it is essential for students to draw/write/generate flowchart, algorithm, test cases, mathematical model, Test data set and comparative/complexity analysis (as applicable).

Guidelines for Student's Lab Journal

Program codes with sample output of all performed assignments are to be submitted as softcopy. Use of DVD or similar media containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journals may be avoided. Submission of journal/ term work in the form of softcopy is desirable and appreciated.

Guidelines for Lab /TW Assessment

Term work is continuous assessment that evaluates a student's progress throughout the semester. Term work assessment criteria specify the standards that must be met and the evidence that will be gathered to demonstrate the achievement of course outcomes. Categorical assessment criteria for the term work should establish unambiguous standards of achievement for each course outcome. They should describe what the learner is expected to perform in the laboratories or on the fields to show that the course outcomes have been achieved. It is recommended to conduct an internal monthly practical examination as part of continuous assessment.

Guidelines for Practical Examination

1.Internal and external examiners should jointly prepare assignments based on the syllabus.

2.During the practical assessment, the examiner should focus on:

a. The student's ability to explain the implemented code and underlying logic.

b. Understanding of core concepts

c. Answering supplementary questions to assess advanced learning and application of concepts.

3.The student should demonstrate clear communication and effective problem-solving skills.

Guidelines for Laboratory Conduction

Following is a list of suggested laboratory assignments for reference. Laboratory Instructors may design a suitable set of assignments for respective courses at their level. Beyond curriculum assignments and mini-project may be included as a part of laboratory work. The instructor may set multiple sets of assignments and distribute among batches of students. It is appreciated if the assignments are based on real world problems/applications. For each laboratory assignment, it is essential for students to draw/write/generate flowchart, algorithm, test cases, mathematical model, Test data set and comparative/complexity analysis (as applicable).

List of Assignments

1: Digital Image Representation and Analysis–CO1

- Read grayscale and color images.
- Extract image dimensions, channels, and bit depth.
- Generate image histograms and visualize pixel intensity distributions.
- Compare grayscale and RGB representations.

2: Enhance image quality using point and neighborhood operations– CO1

- Perform brightness and contrast adjustment.
- Apply histogram equalization.
- Implement smoothing filters.
- Implement sharpening filters.
- Compare enhancement results quantitatively.

3: Investigate how image processing prepares images for vision tasks – sCO2

- Apply Sobel, Prewitt, and Canny edge detectors.
- Compare extracted edges.
- Identify significant visual features.
- Evaluate robustness under noisy conditions.

4: Transform raw images into meaningful visual regions – CO2

- Apply threshold-based segmentation.
- Perform region-based segmentation.
- Detect and isolate objects.
- Compare segmentation outputs under varying thresholds.

5: Detect and match visual features across multiple images – CO3

- Implement Harris Corner Detection.
- Detect ORB features.
- Match corresponding features between image pairs.
- Visualize matching results.

6.Analyze dynamic visual scenes – CO3

- Detect moving objects in video sequences.
- Implement optical flow techniques.
- Track motion trajectories.
- Visualize object movement.

7. Develop a vision-enabled document analysis application – CO4

- Acquire document images.
- Perform preprocessing.
- Extract text using OCR.
- Store extracted information in structured format.
- Evaluate extraction accuracy.

8: Implement a biometric authentication application - CO4

- Detect faces in images/videos.
- Extract facial features.
- Train a recognition model.
- Authenticate users.
- Analyze system performance.

9: Evaluate deep learning-based vision solutions – CO5

- Use a pre-trained CNN model.
- Fine-tune for a custom dataset.
- Train and validate the model.
- Compute accuracy, precision, recall, and F1-score.
- Interpret results.

10: Apply design thinking to solve a real-world problem using image processing and computer vision – CO5

Choose any one domain from healthcare, agriculture, transportation, education, environment monitoring :

- Identify a problem.
- Design the vision pipeline.
- Develop a prototype.
- Test system performance.
- Present findings and improvements.

Text Books:

1. Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, 4th Edition, Pearson Education.
2. Joseph Howse, Joe Minichino, *Learning OpenCV 4 Computer Vision with Python,* Third Edition, Packt

Reference Books:

1. Richard Szeliski, *Computer Vision: Algorithms and Applications,* Second Edition, Springer

2. Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning,* MIT Press, 2016

3. Ahmed Fawzy Gad, *Practical Computer Vision Applications Using Deep Learning with CNNs,* Apress, 2018.

Learning Resources:

1. https://www.kaggle.com/learn/computer-vision
2. https://docs.ultralytics.com/quickstart
3. https://www.tensorflow.org/tutorials

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PEC-321D-ITT : Advanced Database Management System

Teaching Scheme	Credits	Examination Scheme
Theory :3 Hours/Week	03	CCE - 30 Marks End Semester - 70 Marks

Prerequisite Courses: Basic of Database Management Systems (DBMS)

Course Objectives:

- Understand distributed and parallel database architectures and their applications.
- Apply concepts of NoSQL, graph databases, and MongoDB for modern data management.
- Understand vector databases and handling of high-dimensional unstructured data.
- Analyse cloud data warehousing and lakehouse architectures for big data analytics.
- Explore streaming, time-series, and next-generation database systems.

Course Outcomes:

- Analyze and compare parallel and distributed database architectures, fragmentation strategies, query processing, and concurrency control mechanisms.
- Apply NoSQL database concepts and implement MongoDB-based solutions using CRUD operations and indexing techniques.
- Model and manage complex and unstructured data using column-family, graph, and vector database technologies.
- Design analytical data solutions using dimensional modeling, OLAP architectures, ETL processes, and cloud data warehousing platforms.
- Evaluate streaming, time-series, and emerging database technologies for real-time analytics, IoT applications, and modern data-driven systems.

Course Contents

Unit No: I	Parallel & Distributed Databases	09 Hours

Relational Data Model and Integrity Constraints , Need for Parallel and Distributed Databases, Introduction to Parallel Databases , Parallel Database Architectures (Shared Memory Architecture, Shared Disk Architecture, Shared Nothing Architecture),Introduction to Distributed Databases, Distributed Data Storage , Data Fragmentation Techniques (Horizontal Fragmentation ,Vertical Fragmentation),Catalog Management, Distributed Query Processing , Distributed Concurrency Control , Comparison of Parallel and Distributed Database Architectures.

Case Study: Globally Distributed Database Architecture

Unit No: II	Introduction to NoSQL Database Systems and MongoDB	09 Hours

Limitations of Relational Databases, Introduction to NoSQL Databases, Characteristics of NoSQL Databases, Four Types of NoSQL Databases (Key-Value Stores , Document Stores , Column-Family Stores , Graph Databases), Document-Oriented Databases, Introduction to MongoDB, MongoDB Architecture and Collections , CRUD Operations(insertOne(), insertMany() , find(), FindOne(),

updateOne(), updateMany(), deleteOne(), deleteMany()),Indexing in MongoDB

Case Study : MongoDB-Based E-Commerce Product Catalog Management System

Unit No: III	Advanced NoSQL Databases and Emerging Data Models	09 Hours

Advanced NoSQL Databases: Column-Oriented Databases, Apache Cassandra Architecture, Introduction to Graph Databases, Connected Data and Graph Processing, Introduction to Neo4j, Nodes, Relationships and Properties.

Evolution from NoSQL to Vector Databases, Introduction to Vector Databases, Need for Vector Databases, Characteristics of Unstructured Data (Text , Images, Audio, Video), Vector Embeddings, Transforming Unstructured Data into Numerical Vectors, Challenges of Storing and Retrieving High-Dimensional Data**.**

Case Study : AI-Powered E-Commerce Product Search using Vector Embeddings

Unit No: IV	Cloud Data Warehousing	09 Hours

Data Warehousing: System architecture models, core processing components, Dimensional Modeling: Star Schema, Snowflake Schema, and Fact Constellation Schema. Online Analytical Processing (OLAP): Architectural tiers (ROLAP, MOLAP, HOLAP), Extraction-Transformation-Loading (ETL) engineering Principles. **Cloud Data Warehousing:** Limitations of traditional OLAP (ROLAP/MOLAP). Introduction to Cloud Data Warehouses (Snowflake architecture, Google BigQuery).

Case Study: Cloud Data Warehouse Implementation for Retail Sales Analytics Using Snowflake

Unit No: V	Stream Processing, Time-Series, and Next-Gen Models	09 Hours

Streaming Data: Introduction to real-time data ingestion and event streaming (Apache Kafka ecosystem basics integrated with databases). **Time-Series & IoT:** Introduction to Time-Series Databases (**InfluxDB / TimescaleDB**) for IoT and metrics tracking. **Specialized Models:** Spatial and Geographical Information Systems (GIS) data handling , Active databases, and triggers. **Modern Analytics:** Quick overview of the data mining KDD process using automated cloud ML tools.

Case Study : Apache Kafka real-time streaming system

Learning Resources :

Text Books:

1. Abraham Silberschatz, Database System Concepts, McGraw Hill.

2. S.K. Singh, Database Systems: Concepts, Design and Application, Pearson.

Reference Books:

1. MongoDB: The Definitive Guide – O Reilly

2. Principles of Distributed Database Systems – M. Tamer Ozsu

3. Data Mining: Concepts and Techniques – Jiawei Han

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PEC-322D-IT : Advanced Database Management System Lab 1

Teaching Scheme	Credits	Examination Scheme
Practical :2 Hours/Week	02	Term Work - 25 Marks Practical : 25 Marks

Prerequisite Courses :

- Database Management Systems (DBMS)
- Basic SQL Programming
- Fundamentals of Data Structures

Course Objectives:

- To understand implementation of advanced database concepts.
- To develop practical skills in SQL and NoSQL databases.
- To apply distributed and cloud database concepts in real systems.
- To implement data modeling and query processing techniques.
- To gain hands-on experience with modern database tools.

Course Outcomes:

- To design and implement database schemas using SQL.
- To apply CRUD operations in relational and NoSQL databases.
- To implement indexing and query optimization techniques.
- To analyze distributed and cloud-based database systems.
- To develop practical understanding of modern database architectures.

Guidelines for Instructor's Manual

The faculty member should prepare the laboratory manual for all the experiments and it should be made available to students and laboratory instructor/Assistant.

Guidelines for Student's Lab Journal

1.Student should submit term work in the form of a handwritten journal based on a specified list of assignments.

2. Practical Examination will be based on all the assignments in the lab manual

3. Candidate is expected to know the theory involved in the experiment.

4. The practical examination should be conducted if and only if the journal of the candidate is complete in all respects.

Guidelines for Lab /TW Assessment

Examiners will assess the student based on performance of students considering the parameters such as timely conduction of practical assignment, methodology adopted for implementation of practical assignment, timely submission of assignment in the form of handwritten write-up along with results. Appropriate knowledge of usage of software and hardware related to respective laboratory should be checked by the concerned faculty member. There must be hand-written write-ups for every assignment in the journal.

Guidelines for Laboratory Conduction

The instructor is expected to design the laboratory assignments by considering the course prerequisites, recent developments in database technologies, industrial relevance, and practical applications of Advanced Database Management Systems.

It is recommended that the assignments be based on real-world applications

All practical assignments should be conducted using 64-bit open-source database software and tools such as MongoDB Community Edition, MongoDB Compass, Neo4j Desktop, and suitable SQL database systems (MySQL/PostgreSQL).

Group A assignments are compulsory and should be performed by individual student. Group B case study may be performed in a group of 3/4. Mini project of Group C can be implemented using any suitable front-end. But the back-end must be MongoDB.

List of Assignments

Group A :

1. Create a database with a suitable example using MongoDB & perform creation and document insertion operations.
2. Implementation of document deletion and update operations in MongoDB.
3. Execute and analyze MongoDB queries using Find, Query Criteria, Type-Specific Queries, $where clauses, and Cursor Operations.
4. Comparative study of relational databases and graph databases for market trend analysis.

Group B :

1. Design and analysis of Star Schema for an Online Shopping System.
2. Design and analysis of Snowflake Schema for an Online Shopping System.
3. Study and implementation of the Property Graph Model using Neo4j Graph Database.

Group C :

Mini Project based on Advanced Database Management System Concepts (Choose any one topic):

1. Real Time Food Delivery Tracker
2. Retail Sales Dashboard
3. Hospital Database Management using MONGODB.
4. AI Product Search
5. Movie Recommendation System

Text Books:
1. MongoDB: The Definitive Guide – Shannon Bradshaw, Kristina Chodorow, Michael Dirolf. 2. Learning Neo4j – Rik Van Bruggen.
Reference Books:
1. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence – Pramod J. Sadalage and Martin Fowler. 2. Graph Databases – Ian Robinson, Jim Webber, Emil Eifrem.

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

MDM-331-ITT: IoT-based Product Development

Teaching Scheme	Credits	Examination Scheme
Tutorial :02 Hours/Week	2	Term Work : 25 Marks Oral: 25 Marks

Prerequisite Courses: Essential baseline knowledge includes Computer Networks

Course Objectives:

- Understand the internal architecture of microcontrollers and microprocessors and evaluate data acquisition interfaces of IoT cloud platforms.
- Learn the interfacing and timing principles for connecting analog and digital sensors and industrial actuators to embedded systems.
- Understand client-server architecture, local web hosting, and API communication in resource-constrained embedded devices.
- Develop knowledge of TCP/UDP socket communication and publish-subscribe messaging protocols for IoT applications.
- Explore end-to-end product design, simulation platforms, and time-series data analytics for intelligent systems.

Course Outcomes:

- Analyze and select appropriate development boards and cloud services to meet the requirements of IoT systems.
- Develop interfacing circuits for sensors and actuators in embedded systems.
- Develop embedded web servers and implement HTTP-based communication between edge devices and applications.
- Implement network communication using socket programming and MQTT protocols for IoT applications.
- Build and evaluate end-to-end IoT solutions using simulation platforms and cloud analytics.

Course Contents

Unit No: I	Hardware Architecture and IoT Platforms	03 Hours

Comparative analysis of microcontroller and microprocessor architectures in IoT development boards. Understanding GPIO layouts, processing power boundaries, and physical memory configurations. **System-on-Chip (SoC) evaluation for wireless architectures:** Internal architectural review of the ESP8266 and ESP32 modules. Technical review of prominent commercial and open-source IoT Platforms. Evaluating device management registry setups, data ingestion interfaces, and dashboard visualization widgets.

Case Study : Industrial Logistics Fleet Management Tracking System

Unit No: II	Transduction, Actuation, and Peripheral Interfacing	03 Hours

Principles of sensor data acquisition: Interfacing analog sensors and digital sensor ICs to standard microcontrollers. Calculating analog-to-digital conversion (ADC) resolutions and step sizes. **Electronic interfacing and driver mechanics for actuators:** Utilizing transistor switches and Optocouplers to drive mechanical relays. Understanding pulse-width modulation (PWM) signals to precisely manipulate servo motor positions

High-power load management: Analyzing H-Bridge circuit configurations and timing sequences required to perform bidirectional control of DC Motors and step-angle sequences for Stepper Motors using Raspberry Pi GPIO pins

Case Study : Automated Precision Environmental Climate Vault

Unit No: III	Web Servers and Application-Layer Protocols	03 Hours

Architecture of local web environments: Designing lightweight standalone HTTP socket servers directly on embedded hardware (ESP32/ESP8266). **Client-Server transactional frameworks:** Deconstructing HTTP request methodologies. Parsing JSON/XML payloads on resource-constrained microcontrollers. **Application integration:** Interfacing local hardware networks with public cloud API gateways. Constructing authenticated URL query requests to regularly update fields on the Things peak platform and configuring mobile app Webhooks via Blynk.

Case Study : Off-Grid Solar Micro-Grid Remote Dashboard

Unit No: IV	IoT Network Layer, Socket Programming, Messaging Protocols	03 Hours

Network Socket Architecture: Writing network scripts to configure an Arduino or Raspberry Pi as a dedicated TCP/UDP server. Handling incoming concurrent client connections, managing data buffer boundaries, and delivering structured sensor arrays over active network sockets.

Constrained wireless infrastructure topologies: IEEE 802.15.4 standard framework for low-rate wireless personal area networks (LR-WPANs). Analyzing IPv6 header compression, routing structures, and packet delivery using 6LoWPAN protocol implementations on edge gateways. **Pub-Sub Messaging Frameworks:** Detailed study of the MQTT protocol architecture Contrast MQTT with RESTful API web-services for regular ambient temperature streaming over constrained industrial networks.

Case Study : Hazardous Warehouse Gas Leakage Safety Mesh Network

Unit No: V	Edge Analytics, Predictive Modeling, and Product Design	03 Hours

Processing telemetry data in cloud environments: Introduction to edge and cloud data analytics. Formatting time-series sensor databases and implementing simple linear regression or moving-average models for temperature trend forecasting.

IoT Simulation and prototyping environments: Leveraging virtual platforms to rapidly evaluate protocol configurations, verify code behavior, and map virtual node topologies before physical component layout.

Case Study : Municipal Smart Water Distribution and Distribution Analytics Grid

Learning Resources

Text Books:

1. "Internet of Things: Architecture and Design Principles" by Dr. Raj Kamal (McGraw Hill Education, 2nd Edition)
2. "Internet of Things: A Hands-On Approach" by Arshdeep Bahga and Vijay Madisetti (Universities Press)

Reference Books:

1. "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things" by David Hanes, Gonzalo Salgueiro, and Patrick Grossetete (Cisco Press / Pearson Education)
2. "Designing the Internet of Things" by Adrian McEwen and Hakim Cassimally (Wiley India)
3. "Exploring Raspberry Pi: Interfacing to the Real World with Embedded Linux" by Derek Molloy (Wiley)
4. "Analytics for the Internet of Things (IoT)" by Andrew Minteer

E-Books:

1. NPTEL Course on “Introduction to Internet of Things”

Link of the Course: https://onlinecourses.nptel.ac.in/e-learning/preview/noc22_cs53

2. NPTEL Course on “Design for Internet of Things”

Link of the Course: https://onlinecourses.nptel.ac.in/e-learning/preview/noc21_ee85

Savitribai Phule Pune University

Third Year of Information Technology Engineering (2024 Course)

MDM-331-ITT: IoT-based Product Development Lab

Teaching Scheme	Credits	Examination Scheme
Practical :2 Hours/Week	01	Term Work - 25 Marks Practical : 25 Marks

List of Experiments :

Group A: All compulsory
1. Study and survey of Various Development Boards for IoT.
2. Study and survey of various IoT Platforms
3. LED blinking with Arduino Uno.
4. IoT sensors interface with Arduino (Temperature/Light sensors)
5. Integration of Actuators with nodeMCU (Servo motor/Relay).
6. Implementation of DC Motor and Stepper Motor Control with Raspberry Pi
Group B: Any Three to be Performed
7. LED Control Using , ESP8266 and Blynk app
8. Cloud Ready Temperature Sensor (DHT11) Using Raspberry-pi and Thingspeak
9. Web-based home automation using Arduino Uno and Bluetooth
10. Introduction to HTTP. Hosting a basic server from the ESP32 to control various digital based actuators (led, buzzer, relay) from a simple web page.
11. Displaying humidity and temperature data on a web-based application
Group C: Course Project (Any 1 Group of 3 Students)
12. Write a program to create a TCP/UDP server on Arduino/Raspberry Pi and respond with sensor data to a TCP client when requested.
13. Demonstration of LoWPAN protocol for IoT gateway functions to send IPv6 packets over IEEE 802.15.4 based networks.
14. Measurement of ambient temperature from sensors and publishing using MQTT/RESTful APIs.
15. IoT based small project implementation on the topics based on small problem statements of the fields like chat bot, smart home (Home Automation), social issues and environmental issues etc. This project can be built on any IoT simulation platform like Tinkercad, Cooja etc.

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

OLE-341-IT : Open Elective-I

Teaching Scheme	Credits	Examination Scheme
Theory :2 Hrs/wk	02	CCE : 15 Marks Endsemester :35 Marks

Open Electives (OE) are multidisciplinary courses allowing students to study subjects outside their core discipline to foster holistic development and skill enhancement. Students pick subjects outside their core specialization from the following list to broaden their knowledge base.

Sr. No.	Name of the Open Elective	Offering Discipline(Board/Faculty)
1	Business Law	Commerce & Management
2	IPR and Cyber Laws	Law / Faculty of Humanities
3	Business Marketing	Commerce & Management
4	Agri Business Management: Banking Operation and finance	Commerce & Management
5	Banking ,Finance & Insurance	Commerce & Management
6	Statistics and Computer Applications	Commerce & Management
7	Business Administration	Commerce & Management
8	Cost & Works Accounting	Commerce & Management
9	Sustainability Development	
10	The Constitution of India	
11	Digital Personal Data Protection	

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

ELC-342-IT **:** Technical Seminar

Teaching Scheme	Credits	Examination Scheme
Practical : 2 Hrs/Week	1	OR – 50 Marks

Course Objectives: The objective of this course is

- To develop research orientation and technical communication skills in emerging Computer Engineering and Artificial Intelligence domains.
- To enable students to critically review, analyze, and synthesize contemporary research papers, white papers, patents, and technical standards.
- To promote interdisciplinary thinking aligned with NEP-2020 multidisciplinary philosophy.
- To inculcate ethical awareness, sustainability perspective, and societal impact analysis of AI systems.
- To prepare students for industry, higher education, entrepreneurship, and innovation ecosystems.

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

- Identify and select emerging trends in IT industry via research problems through a systematic literature survey.
- Critically analyze research papers using technical, methodological, and ethical perspectives.
- Prepare structured structures technical reports and presentations.
- Deliver effective oral presentations demonstrating clarity, confidence, and domain understanding.
- Demonstrate professional ethics and academic integrity

Guidelines for Conduct of Technical Seminar

The Technical Seminar shall be research-oriented and domain-specific, focusing strictly on recent development in IT industry.

Topic Selection Guidelines

1. Topic must be from emerging trends in the information technology industry (last 3–5 years).
2. Must involve a minimum of 5 recent research papers (IEEE, ACM, Elsevier, Springer, etc.). They should summarize the papers by reading abstracts and identifying ideas, conclusions, advantages of their approaches, and drawbacks. Generalize results from a research paper to related research problems. Compare approaches and identify weak-nesses and strengths in recent research articles on the subject. Practical sessions on how to read, analyze, and summarize research papers should be included.
3. Should not be a basic textbook topic.
4. Include: Problem statement, State-of-the-art analysis, Comparative study, Ethical & societal impact, Future research scope, Interdisciplinary themes aligned with NEP encouraged.
5. Topic approval by a faculty panel

Seminar Process

Stage 1: Orientation & Topic Finalization (Week 1–2)

1] Conduct an orientation session explaining:

- Objectives of the technical seminar
- Evaluation criteria and expected outcomes

2] Each student must submit:

- Title of the seminar
- Problem statement
- Relevance to current technology trends
- Approval by guide is mandatory before proceeding.
- Literature Survey & Problem Understanding (Week 2–4)

3] Students must:

- Refer minimum 5–8 recent research papers (IEEE, Springer, Elsevier, ACM, etc.)
- Use scholarly databases like IEEE Xplore, Google Scholar, ScienceDirect

4] Prepare a literature survey matrix, including:

- Author/year
- Methodology used
- Key findings
- Limitations

5] Identify:

- Research gaps and challenges in existing approaches.
- Synopsis Preparation & Presentation (Week 4–5)
- Submit a 2–3 page synopsis including:
- Introduction
- Literature insights
- Objectives
- Proposed seminar scope

6] Conduct a Synopsis Presentation (5–7 minutes):

- Evaluate clarity of understanding
- Receive feedback for improvement

7] Approval required before proceeding to full report

- In-depth Study & Content Development (Week 5–8)

8] Students should:

- Deeply analyze concepts, models, architectures, or case studies
- Include diagrams, flowcharts, and comparative tables
- Weekly review meetings with guide:
- Track progress
- Ensure conceptual clarity

9] Emphasis on:

- Critical analysis (not just description)
- Real-world applications
- Draft Report Submission & Review (Week 8–10)

10] Submit first draft of the report

11] Guide provides feedback on:

- Technical content quality
- Structure and coherence
- Referencing and plagiarism

12]Students must revise based on suggestions by the guide.

- Pre-Seminar Presentation (Mock Evaluation) (Week 10–11)
- Conduct a mock presentation simulating final evaluation

13] Focus on:

- Presentation skills
- Time management
- Handling questions
- Peer and faculty feedback should be incorporated
- Final Report Submission (Week 11–12)

14] Ensure:

- Proper formatting
- Plagiarism compliance (<20%)
- Correct referencing using reference managers like Zotero and Mendeley Desktop.
- Final Seminar Presentation & Viva Voce (Week 12–13)

15] Presentation duration: 10–15 minutes, followed by Q&A session (5–10 minutes)

16] Evaluation based on:

- Depth of understanding
- Analytical ability
- Communication skills

Method of Evaluation

1. During the final oral examination, each student is expected to prepare and present a topic on engineering/technology, for a duration of about 12to 15 minutes.

2. Each student is expected to present atleast twice during the semester and the student is evaluated based on that.

3. At the end of the semester, he/she can submit a report on his/her topic of seminar and marks are given based on the report and final presentation.

4. A Faculty guide is to be allotted and he/she will guide and monitor the progress of the student and maintain attendance.

SYLLABUS

SEMESTER - IV

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PCC-351-IT: Data Science & Big Data Analytics

Teaching Scheme	Credits	Examination Scheme
Theory :3 Hours/Week	3	CCE - 30 Marks End Sem - 70 Marks

Prerequisite Courses: Probability and Statistics, Database Management Systems, Programming Fundamentals

Companion Course: Machine Learning

Course Objectives:

- To introduce the concepts, characteristics, architectures, storage technologies, and applications of Big Data.
- To develop the ability to perform data cleaning, transformation, integration, and analytics using Python and Pandas.
- To impart knowledge of Hadoop-based distributed storage and processing using HDFS, MapReduce, YARN, and the Hadoop ecosystem.
- To familiarize students with Apache Spark architecture, Spark SQL, DataFrames, and RDD-based distributed data processing.
- To develop the ability to analyze and communicate insights through data visualization techniques and visualization tools

Course Outcomes:

- **CO1:** Explain the fundamental concepts, characteristics, architectures, storage technologies, and applications of Big Data.
- **CO2:** Apply Python and Pandas techniques for data cleaning, transformation, aggregation, integration, and analytical processing of datasets.
- **CO3:** Analyze and illustrate distributed storage and processing mechanisms using HDFS, MapReduce, YARN, and components of the Hadoop ecosystem.
- **CO4:** Apply Spark-based processing techniques using Spark SQL, DataFrames, and RDD operations for Big Data analytics.
- **CO5:** Analyze data and communicate insights using appropriate visualization techniques, EDA methods, and visualization tools.

Course Contents

Unit : I	Big Data Overview	09 Hours

Introduction: Need of Big Data, Classification of Data – Structured, Semi-structured and Unstructured, Big Data Definitions, Big Data Characteristics - 3Vs, Big Data Types, Big Data Classification, Big Data Handling Techniques. **Big Data Scalability techniques:** Brief introduction to **–** Massively Parallel Processing Platforms, Cloud Computing, Grid and Cluster Computing, Big Data Architecture – layers. **Data sources, quality, preprocessing and storage:** Various kinds of Data sources, Data quality

aspects, Data preprocessing steps, Exporting data store to cloud. **Big Data Applications and Case Studies –** Big data in marketing and sales, Big data in healthcare, Big data in medicine, Big data in Advertising.

Unit : II	Data Analytics Using Python	09 Hours

Introduction to Data Analytics: Data analytics Definition , Types of data analytics **–** Descriptive, Diagnostic, Predictive and Prescriptive Analytics, Data Analytics life Cycle.

Using Python for Data Analytics: Data Cleaning and Preparation – Handling missing data, filtering out missing data, filling in missing data, Data transformation – removing duplicates, transforming data – using a function, using a mapping, replacing values, renaming indexes, Discretization and Binning, Detecting and Filtering Outliers, Handling Categorical data – Encoding types, **Data Wrangling with Python** : Concept of Hierarchical Indexing – reordering and sorting levels, summary statistics by level, Indexing with columns, Combining and Merging Datasets using – merge , join and concat, Reshaping and Pivoting using – pivot and melt, Data Aggregation and Group Operations – concept of group operation, grouping with functions, grouping by index levels

Case Studies: Student Performance Analytics (Resource - UCI Student Performance Dataset)

Unit : III	Hadoop Storage And Processing	09 Hours

Introduction: Programming models – Centralized computing, Distributed computing, Introduction to Hadoop Core Components – Hadoop Distributed File System (HDFS), Map reduce v1, YARN, Map reduce v2. Features of Hadoop. Hadoop Distributed File System **–** Hadoop Data storage, Physical organization of file system. **MapReduce Framework and Programming Model:** Concept of - aggregation function, reducer , mapper, features of map reduce framework. Hadoop mapreduce programming model – Concept of key value pair, Grouping by keys, Map reduce Processing steps – Shuffle, Sort , Partitioning and combiner. Example to illustrate usage of HDFS and the map and reduce functions. Hadoop YARN – Hadoop 2 execution model. Hadoop 2 need. Overview of Hadoop ecosystem – Zookeeper, Oozie, Sqoop, Flume, Hive, Pig and Hbase

Case Study: Web Log Analysis - Analyze millions of website visits to identify popular pages (Resource - NASA Web Server Log Dataset)

Unit : IV	Spark Processing	09 Hours

Introduction to Spark: Need for Spark, What is Spark, Features of Spark, Applications of Spark, Hadoop vs. Spark, Concept of In-Memory Distributed Computing. **Spark Cluster Architectural Components:** Spark Driver, Spark Context, Cluster Manager, Worker Nodes, Executors, Spark Program Execution Flow. **Introduction to Spark SQL:** Features of Spark SQL, DataFrames, Schema and Structured Data, DataFrame Operations – Selection, Filtering, Aggregation, Grouping. Introduction to .

Case Study: Movie Recommendation Analytics (Resource - MovieLens Dataset)

Unit : V	Big Data Visualization	09 Hours

Introduction to Data Visualization: Need for Data Visualization, Benefits of Visualization, Role of Visualization in Data Analytics – Simplifying complex data, Identifying trends and patterns, Detecting outliers and anomalies, Supporting decision making, Communicating analytical results, Business intelligence and Decision Support.

Common Visualization Techniques: Bar Chart, Pie Chart, Line Chart, Histogram, Scatter Plot, Heat Map. Exploratory Data Analysis (EDA) – Descriptive Statistics, Trend Analysis, Pattern Discovery, Outlier Identification. Visualization Tools Overview – Matplotlib, Seaborn, and Tableau

Case Study: Smart City Traffic Analytics (Resource **-** Open Government Data Platform India)

Learning Resources

Text Books:

1.Raj Kamal, Preeti Saxena, *Big Data Analytics*, McGraw Hill Education (India) Pvt. Ltd., 2019. ISBN: 978-93-5316-496-6

2. Wes McKinney, "Python for Data Analysis - Data Wrangling with pandas,NumPy, and Jupyter", Third edition, Oreilly Publication, 2022

Reference Books:

1. Tom White, “Hadoop: The Definitive Guide” , Fourth Edition, Oreilly

2. Bill Chambers and Matei Zaharia, “*Spark: The Definitive Guide*, O'Reilly Media, Inc.”, 2018. ISBN: 978-1491912218.

E-Books:

1. NPTEL,https://onlinecourses.nptel.ac.in/e-learning/preview/noc20_cs92

2. https://www.my-mooc.com/en/categorie/big-data

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PCC352-IT : Software Engineering & Project Management

Teaching Scheme	Credits	Examination Scheme
Theory :2 Hours/Week	2	CCE - 30 Marks End Sem - 70 Marks

Prerequisite Courses: Fundamentals of Programming and Problem Solving

Course Objectives:

- To understand software engineering principles and software process models for software development.
- To analyze and model software requirements using UML and software design practices.
- To introduce Agile, DevOps, and collaborative software development methodologies.
- To understand software testing, quality assurance, and maintenance strategies.
- To develop project management skills using estimation, scheduling, and modern software tools.

Course Outcomes:

- **CO1:** Select suitable software development models and engineering practices for software projects.
- **CO2:** Analyze software requirements and prepare UML-based models and SRS documentation.
- **CO3:** Apply Agile methodologies, DevOps principles, and collaborative software tools in software development.
- **CO4:** Perform software testing, quality assurance, and maintenance activities.
- **CO5:** Plan and manage software projects using estimation, scheduling, and tracking tools.

Course Contents

Unit : I	Foundations Of Software Engineering	09 Hours

Introduction to Software: Software myths and engineering approach, Engineering, Nature and Characteristics of Software, Software Crisis and Challenges**,** Software Process

Software Process Framework: Waterfall, Incremental, RAD, Spiral, Component-Based, **Introduction Cleanroom Engineering**

Software Quality Attributes: SQA Plans, Frameworks, ISO 9000 Models, CMM Models.

Case Study : Selection of Suitable Software Process Model for Online Food Delivery System.

Unit : II	Requirements Engineering And UML-Based Design	09 Hours

Requirements Capturing Process: requirements engineering (elicitation, specification, validation, negotiation, prioritizing requirements (Kano diagram) - real life application case study.

Requirement Elicitation Techniques Functional and Non-functional Requirements, Software Requirement Specification (SRS)

UML Modeling: Use Case Diagram, Class Diagram, Sequence Diagram, Activity Diagram Data modeling, data and control flow model, behavioral modeling using state diagrams.

Case Study : **SRS preparation for** Smart Hospital Management System

Unit : III	Agile Software Development And Devops	09 Hours

Agile Development: Agile manifesto, agility and cost of change, agility principles, myth of planned development, toolset for the agile process.

Extreme Programming: XP values, process, industrial XP, SCRUM - process flow, scrum roles, scrum cycle description, product backlog, sprint planning meeting, sprint backlog, sprint execution, daily scrum meeting, maintaining sprint backlog and burn-down chart, sprint review and retrospective.

DevOps Fundamentals: CI/CD Concepts, Version Control using Git and GitHub.

Case Study: Agile Scrum-based Development of College Management System

To understand Agile principles, Scrum workflow, sprint planning, backlog management, and Agile project execution.

Unit : IV	Software Testing, Quality Assurance And Maintenance	09 Hours

Software Testing Fundamentals: Verification and Validation (V&V), Difference between Verification and Validation, Importance of software testing in software engineering,

Levels of Testing: Unit Testing, Integration Testing, System Testing, Acceptance Testing Test Case Design Techniques: Black Box Testing Techniques, White Box Testing Techniques, Boundary Value Analysis, Equivalence Partitioning, Basic Defect Tracking Concepts

Software Quality Assurance (SQA): SQA Framework, Quality Standards and Models ISO 9000 Models, Capability Maturity Model (CMM/CMMI)

Software Metrics: Introduction to Software Metrics, Product Metrics, Process Metrics Software Maintenance and Reengineering

Case Study: Testing and Quality Assurance of Online Banking System

Unit No: V	Software Project Management And Emerging Trends	09 Hours

Project Initiation and Scope Management: Project Initiation Process**,** Project Planning and Scope Management**,** Project Deliverables and Constraints**,** Work Breakdown Structure (WBS)

Project Effort Estimation and Scheduling: Importance of Project Scheduling, Estimating Activity Resources, Estimating Activity Duration, Estimation Techniques: LOC (Lines of Code), Function Point Analysis (Overview), Basic COCOMO Model.

Project Schedule Development Techniques: Introduction to Gantt Charts, Planning Cost Management & Quality Management. Introduction to project Risk Management.

Emerging Trends in Software Engineering: Generative AI in Software Engineering, AI-assisted Coding Tools (GitHub Copilot, ChatGPT), Cloud-based Project Management Tools

Case Study: Project Planning and Scheduling of Smart Healthcare System using Gantt Chart & PERT/CPM

Learning Resources

Text Books:

1. Roger S. Pressman and Bruce R. Maxim, Software Engineering: A Practitioner's Approach, 9th Edition, McGraw-Hill-Education.

2. Ian Sommerville, Software Engineering, 10th Edition, Pearson Education.

Reference Books:

1. Pankaj Jalote, Software Engineering: A Precise Approach, Wiley India.

2. Bob Hughes, Mike Cotterell and Rajib Mall, Software Project Management, 7th Edition, McGraw-Hill Education

3. Roger S. Pressman, Software Engineering, McGraw Hill (for SDLC, Testing, SQA, Project Management concepts).

4. Richard Fairley, Software Engineering Concepts, McGraw-Hill

E-Books:

1. NPTEL- Software Engineering, Project Management, Agile Development Courses https://nptel.ac.in/

2. SWAYAM – Engineering and Software courses https://swayam.gov.in/

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PCC-353-IT : Data Science & Big Data Analytics Lab 1

Teaching Scheme	Credits	Examination Scheme
Practical : 4 Hrs /week	2	TW: 25 Marks PR : 25 Marks

Course Objectives:

- To provide practical exposure to data cleaning, preprocessing, transformation, and analytical operations using Python and Pandas.
- To develop skills in organizing, integrating, reshaping, and analyzing datasets using advanced data manipulation techniques.
- To enable students to configure Hadoop environments and develop MapReduce applications for distributed storage and processing of large-scale data.
- To familiarize students with PySpark and Spark SQL for scalable data processing and analytics.
- To develop the ability to visualize, interpret, and communicate insights from data using Python visualization libraries and Tableau.

Course Outcomes:

- CO1: Apply data cleaning, preprocessing , transformation, and encoding techniques using Python and Pandas for data analytics.
- CO2: Analyze and manipulate datasets using hierarchical indexing, grouping, integration, reshaping, and pivoting operations in Pandas.
- CO3: Develop Hadoop-based applications using HDFS and MapReduce for distributed storage and processing of large datasets.
- CO4: Implement scalable data processing and analytical applications using PySpark DataFrames and Spark SQL.
- CO5: Perform exploratory data analysis and create effective visualizations and dashboards using Python visualization libraries and Tableau.

Guidelines for Instructor's Manual

The faculty member should prepare the laboratory manual for all the experiments and it should be made available to students and laboratory instructor/Assistant.

The instructor's manual should include prologue, university syllabus, conduction & Assessment guidelines, topics under consideration-concept, objectives, Laboratory conduction guidelines and assessment scheme, Aim, Problem Statement , Dataset Description , Algorithm/Procedure , Program Code , Outcomes, Conclusion, Sample test cases and references.

Guidelines for Student's Lab Journal

• Students shall maintain a Laboratory Journal as a part of the Term Work (TW). The journal shall include a prologue, certificate, table of contents, and a record of each assignment containing the title, objectives, problem statement, course outcomes, software and hardware requirements, theory

concepts, algorithm in pseudocode, program code, sample test cases and outputs, date of completion, assessment grade/marks, and assessor's signature.

- Term Work (TW) assessment shall be based on the completion and quality of laboratory assignments, maintenance of the laboratory journal, and performance during practical sessions.
- Students are expected to understand the theoretical concepts and implementation details related to the experiments performed.
- A student shall be eligible for the Practical/Oral Examination only if the Laboratory Journal is completed in all respects and duly certified by the concerned faculty member and the Head of the Department.
- All assignments including **PART A and PART B** prescribed in the syllabus must be performed and documented in the Laboratory Journal.

Guidelines for Lab /TW Assessment

1. Examiners will assess the term work based on performance of students considering the parameters such as timely conduction of practical assignment, methodology adopted for implementation of practical assignment, timely submission of assignment in the form of handwritten write-up along with results of implemented assignment, attendance etc.
2. Examiners will judge the understanding of the practical performed in the examination by asking some questions related to theory and implementation of experiments he/she has carried out.
3. Appropriate knowledge and usage of software and hardware such as IDE, Dataset, tools , coding standards , algorithm to be implemented etc. should be checked by the course instructor.

Guidelines for Laboratory Conduction

The instructor is expected to frame the problem statements of assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic.The instructor may set multiple sets of assignments and distribute among batches of students. It is appreciated if the assignments are based on real world problems/applications.
All the assignments should be conducted on multi-core hardware and 64-bit open-source software.

Guidelines for Practical Examination

Both internal and external examiners should jointly set problem statements for practical examination. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to judge the student‘s understanding of the fundamentals, effective and efficient implementation.The evaluation should be done by both external and internal examiners.

List of Assignments

Part A: Data Analytics using Python and Pandas

1: Data Cleaning and Data Preprocessing using Pandas

Apply data cleaning techniques by creating a sample CSV file containing the fields CustID, Name, Age, Income, City, and Country, and perform the following operations:

i. Create the dataset with missing values. Load the dataset and display the count of missing values.
ii. Explore the dataset using info(), head(10), describe(), and isnull().sum().

iii. Fill missing values in Age using the median, Income using the mean, and Country using the mode.

iv. Remove duplicate records based on Name, City, and Income.

v. Detect and filter out outliers in Income using the Interquartile Range (IQR) method.

vi. Transform the Name column to title case using a suitable function.

vii. Download the Customer Purchase Behaviour dataset (or any equivalent customer-related dataset) from Kaggle (or an equivalent data repository) and perform all the above operations on it.

2: Binning and Categorical Data Encoding

Apply binning and categorical data encoding techniques by creating a sample CSV file containing the fields Student_ID, Score, and Grade_Letter. Keep the Grade_Letter column initially empty. Load the dataset and perform the following operations:

i. Remove duplicate records (if any).

ii. Explore the dataset using appropriate Pandas functions.

iii. Bin the Score attribute into the categories F, D, C, B, and A using the intervals [0,40), [40,55), [55,70), [70,85), and [85,100].

iv. Create and fill the Grade_Letter column based on the above score ranges.

v. Perform Label Encoding on the Grade_Letter attribute (e.g., F=0, D=1, C=2, B=3, A=4).

vi. Perform One-Hot Encoding on the Grade_Letter attribute.

vii. Download the Students Performance in Exams dataset (or any equivalent student performance dataset) from Kaggle (or an equivalent data repository) and perform all the above operations on it.

3: Hierarchical Indexing and Group Operations

Demonstrate the use of hierarchical indexing and group operations by creating a sample CSV file containing the fields City, Product, Year, Quarter, and Sales. Load and explore the dataset and perform the following operations:

i. Set City and Product as a hierarchical index.

ii. Sort the hierarchical index with City in descending order and Product in ascending order.

iii. Compute the mean, sum, and maximum of Sales grouped by the City level only.

iv. Group the data by City and Year to determine the total sales for each city in each year.

v. Group the data using a custom function that maps Q1 and Q2 to "H1" and Q3 and Q4 to "H2", and compute the total sales for each group.

vi. Group the data by the Product index level (level=1) and calculate the total sales for each product.

vii. Download the Sales Forecasting dataset (or any equivalent sales-related dataset) from Kaggle (or an equivalent data repository) and perform all the above operations on it. Use Region and Category (or equivalent categorical attributes) as hierarchical indices and Sales (or an equivalent numerical attribute) as the value column.

4: Data integration – Combining DataFrames using merge() , join() and concat()

Illustrate the use of merge(), join(), and concat() by creating the following sample CSV files:

a. orders.csv with fields: OrderID, CustomerID, OrderDate

b. customers.csv with fields: CustomerID, Name, City

c. order_details.csv with fields: OrderID, Product, Amount

Perform the following operations on the above sample datasets:

i. Merge orders and customers on CustomerID using an inner join and store the result as order_cust.

ii. Merge order_cust with order_details on OrderID using a left join and store the result as full_data.

iii. Create the same combined dataset using join() instead of merge() by setting appropriate indexes.

iv. Create a new DataFrame containing two additional customer records and concatenate it with the customers DataFrame.

v. Filter out rows having missing values in the Name column after merging.

vi. Save the final merged dataset as complete_orders.csv.

vii. Download any suitable sales, retail, or customer-order dataset from Kaggle (or an equivalent data repository) and perform the above data integration operations using appropriate common attributes for merging and joining the datasets.

5: Reshaping and Pivoting using pivot() , pivot_table() , and melt()

Analyze reshaping operations by creating a sample CSV file containing the fields: Store, Product, Month, and Sales. Load the dataset and perform the following operations:

i. Load the CSV file and convert the data into wide format using pivot() with Product as rows, Month as columns, and Sales as values.

ii. Add a duplicate row (e.g., A, Laptop, Jan, 5100) and use pivot_table() with aggfunc='mean'.

iii. Melt the pivoted table back to the original long format without considering the duplicate row.

iv. Create a hierarchical index using Store and Product, and use stack() and unstack() to move Month from columns to rows and back.

v. Before pivoting, detect and remove outliers in Sales (e.g., values greater than 2 standard deviations from the mean).

vi. Download any suitable sales or transaction dataset from Kaggle (or an equivalent data repository) and perform the above reshaping and pivoting operations using appropriate categorical and numerical attributes.

Part B: Big Data Analytics using Hadoop and Spark

6. Installation and Configuration of Single Node Hadoop System.

Install and configure a single-node Hadoop cluster and perform basic HDFS operations using HDFS commands

7. Develop a MapReduce program

Develop a MapReduce program to calculate the frequency of each word in a given text file stored in HDFS.

8. Web Log Analysis

Develop a MapReduce program to analyze a web server log file and calculate the total number of visits from each IP address. Download web server log dataset (or similar) from Kaggle or an equivalent data repository and store it in HDFS.

9. PySpark Installation and Environment Setup

Install and configure PySpark on a desktop computer. Verify the installation by creating a SparkSession and a sample DataFrame.

10. Spark DataFrame and Spark SQL Operations

Implement a PySpark application to load any suitable CSV dataset into a DataFrame and perform the following operations:

i. Select required columns.

ii. Filter records based on specified conditions.

iii. Perform aggregation and grouping operations.

iv. Sort records.

v. Execute equivalent SQL queries using Spark SQL for the above operations.

11. Data Visualization and Exploratory Data Analysis (EDA)

Perform Exploratory Data Analysis (EDA) using Python libraries such as Matplotlib and Seaborn by downloading the Air Quality dataset (or any suitable dataset) from Kaggle (or an equivalent data repository). Perform the following operations:

i. Load, explore, and preprocess the dataset using appropriate Pandas functions.

ii. Visualize the distribution of numerical attributes using histograms.

iii. Plot boxplots to identify outliers in numerical attributes.

iv. Create scatter plots to analyze relationships between selected attributes.

v. Generate bar charts for the analysis of categorical attributes.

vi. Plot a correlation heatmap to observe relationships among numerical attributes.

vii. Summarize the key observations obtained from the visualizations.

12. Data visualization using Tableau

Perform data visualization by installing Tableau Desktop (or using Tableau Public). Download the Air Quality dataset (or any suitable dataset) from Kaggle (or an equivalent data repository) and perform the following data visualization operations:

i. Import the dataset into Tableau.

ii. Create worksheets using appropriate visualizations such as bar charts, line charts, pie charts, histograms, and scatter plots.

iii. Apply filters and sorting to analyze the data from different perspectives by changing dimensions, filters, or levels of aggregation.

iv. Create calculated fields, if required, for meaningful analysis.

v. Design an interactive dashboard by combining multiple visualizations.

vi. Add appropriate titles, labels, legends, and formatting to improve readability.

vii. Summarize the key insights obtained from the dashboard.

Virtual Laboratory

https://www.my-mooc.com/en/categorie/big-data

Additional Reference

https://nptel.ac.in/courses/106104189

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PCC354-IT : Software Engineering & Project Management Lab 2

Teaching Scheme	Teaching Scheme	Teaching Scheme
Practical : 2 Hrs./week.	1	TW: 25 Marks PR :25 Marks

Prerequisite Courses, if any:

- Good understanding of Object Oriented Programming Concepts.
- Basic knowledge of software development processes and problem-solving

Course Objectives:

- To understand the fundamentals of Software Engineering and the Software Development Life Cycle (SDLC).
- To understand project planning, effort estimation, and scheduling techniques.
- To apply software quality assurance, testing strategies, and defect management.
- To understand and apply risk management and software configuration management.
- To practice documentation and use of industry-standard project management tools.

Course Outcomes:

- CO1: Analyze software engineering fundamentals and apply appropriate SDLC process models for given requirements.
- CO2: Prepare Software Requirement Specification (SRS), project plans, and effort/cost estimates using standard techniques.
- CO3: Apply project scheduling, risk management, and quality assurance strategies for a software project.
- CO4: Perform software testing using Black Box and White Box strategies; prepare test plans and defect reports.
- CO5: Apply Software Configuration Management (SCM) and version control using industry tools.

Guidelines for Instructor's Manual

The instructor's manual is to be developed as a hands-on resource and reference. The instructor's manual should include a prologue (about University/program/institute/department/foreword/preface), University syllabus, conduction and assessment guidelines, topics under consideration with concept, objectives, outcomes, a set of typical applications/assignments/guidelines, and references

Guidelines for Student's Lab Journal

1.The laboratory assignments are to be submitted by student in the form of journal. 2. Journal consists of prologue, Certificate, table of contents, and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory- SEPM feature/Concept in brief, algorithm, flowchart, test cases, conclusion/analysis.

3. Program codes with sample output of all performed assignments are to be submitted as hardcopy.

4. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal may be avoided.

5. Use of DVD containing students programs maintained by lab In-charge is highly encouraged.

6. For reference one or two journals may be maintained with program prints at Laboratory.

***Note*: Students** shall form a group of **4 to 5 candidates** and select **one software project title/problem statement** from real-world application domains. The same selected project shall be continued across **all seven practical assignments** to ensure continuity in requirement analysis, UML modeling, Agile planning, testing, scheduling, and project management. Each group shall perform all SEPM practical assignments using the same project title.

Guidelines for Lab /TW Assessment

1. Continuous assessment of laboratory work is done based on overall performance and individual assignment performance.
2. Each lab assignment assessment will assign grade/marks based on parameters with appropriate weightage.
3. Suggested parameters for overall assessment as well as each lab assignment assessment include: timely completion, performance, innovation, use of tools, quality of documentation, punctuality, and neatness.

Guidelines for Practical Examination

Both internal and external examiners should jointly set problem statements for practical examination. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to judge the student 's understanding of the fundamentals, effective and efficient implementation. The evaluation should be done by both external and internal examiners.

Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent industry trends related to the topic. The assignment framing policy should address average students and include elements to attract and promote intelligent students. The instructor may set multiple sets of assignments with similar complexity levels and distribute among batches. Encourage students to use industry-standard documentation formats, proper UML notations, and open-source project management tools. A single running case study (e.g., Online Healthcare System, E-Learning Platform, Smart City Dashboard) used across all 7 assignments and the mini project is strongly recommended to provide continuity and depth.

Operating System Recommended: 64-bit Open-source Linux or its derivative / Windows 10+

Tools Recommended: ProjectLibre / MS Project, Git & GitHub/GitLab, Draw.io / StarUML / Lucidchart, any UML modeling tool, JIRA / Trello (open-source alternatives: Taiga, OpenProject).

List of Assignments
SDLC - CO1, CO2
1. Comparative study of SDLC models for real-world software applications
SRS – CO2, CO3
2. Requirement gathering and preparation of SRS document MS word using one Problem statement.
UML Diagrams - CO1, CO3
3. Requirement modeling using UML tools (StarUML / Draw.io / Lucidchart / Visual Paradigm)
Agile - CO1, CO4
4. Agile sprint planning mini case study
Testing - CO3, CO4
5.Test case generation and defect prediction Testing Tools Excel sheets
Project Scheduling – CO5
6.Gantt Chart preparation and project scheduling Trello / MS Project
Mini Project – CO5
7. AI-assisted Project Management Analysis using GitHub/ Copilot/ ChatGPT
Text Books:
1. Software Engineering: A Practitioner's Approach by Roger S. Pressman and Bruce R. Maxim, McGraw-Hill (8th Edition). 2. Software Engineering by Ian Sommerville, Pearson (10th Edition).
Reference Books:
1. Software Project Management by Bob Hughes and Mike Cotterell, McGraw-Hill. 2. Agile Estimating and Planning by Mike Cohn, Prentice Hall.

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PEC-361A-IT: DevOps

Teaching Scheme	Credits	Examination Scheme
Theory :3 Hrs/Week	3	CCE - 30 Marks End Sem - 70 Marks

Prerequisite Courses : Software Engineering

Course Objectives:

- To understand the need of DevOps as a software engineering practice.
- To understand the background of DevOps Evolution.
- To know and understand the concept of Continuous Integration Continuous Delivery (CICD).
- To learn the concept of continuous deployment and test strategies.
- To learn the monitoring system and reliability engineering.

Course Outcomes:

- **CO1:** Understand the fundamental concepts, culture, principles, and practices of DevOps.
- **CO2:** Explain microservices architecture, cloud-native development, containerization, and orchestration technologies used in DevOps environments.
- **CO3:** Comprehend and implement Continuous Integration (CI) and Continuous Delivery (CD) concepts, strategies, and pipelines.
- **CO4:** Analyze and compare various stages of Continuous Deployment, testing strategies, and deployment pipeline practices.
- **CO5:** Evaluate monitoring systems, Site Reliability Engineering (SRE) practices, and apply modern DevOps tools such as Git, Docker, Kubernetes, Jenkins, Ansible, Selenium, Prometheus, Jira, and ELK Stack in software development and operations.

Course Contents

Unit : I	Introduction To Devops And The Culture.	09 Hours

What is DevOps, DevOps culture, collaboration, automation, and Agile connection, High-level DevOps lifecycle and pipeline phases, Roles of developer, operations, and DevOps engineer.

Case Study : Improving Collaboration in an E-Commerce Startup

Unit : II	Microservices Architecture And Cloud Native Development.	09 Hours

Monolithic vs. Microservices architecture, Benefits and challenges of microservices, Containerization principles, Introduction to Docker: image, container, Dockerfile, Orchestration vs. automation.

Case Study: Migrating from Monolithic to Microservices

Unit : III	Continuous Integration And Test-Driven Development.	09 Hours

Why CI/CD is used: faster feedback and quality, Code repository and CI server concepts, CI vs. CD, Basic CI/CD pipeline stages, Simple pipeline examples and reports. Case Study: Automating Software Builds

Unit : IV	Continuous Deployment And Orchestration.	09 Hours

Continuous Deployment: idea and benefits/risks, Deployment pipeline overview, Basic testing stages: unit, integration, smoke/acceptance, Release readiness and rollback awareness.

Case Study: Reducing Production Bugs through Automated Testing

Unit : V	Continuous Monitoring And Site Reliability.	09Hours

Why monitoring matters, Logs, metrics, dashboards, White-box vs. Black-box monitoring, What is SRE? How SRE relates to DevOps, Tool awareness: Git, Docker, Jenkins, Ansible, Selenium, Prometheus, Jira, ELK Stack.

Case Study: Ensuring High Availability of a Streaming Platform

Learning Resources

Text Books:

1. PierluigiRiti, “Pro DevOps with Google Cloud Platform”, Apress, ISBN: 978-1-4842-3896-7.

2. Katrina Clokie, “A Practical Guide to Testing in DevOps”, Lean Publishing published on 2017-08-01

3. Jez Humble and David Farley, “Continuous Delivery”, Pearson Education, Inc, ISBN: 978–0–321–60191–9

Reference Books:

1. Viktor Farcic, “The DevOps 2.0 Toolkit: Automating the Continuous Deployment Pipeline with Containerized Microservices”

2. Jennifer Davis and Katherine Daniels, “Effective DevOps: Building a Culture of Collaboration, Anity, and Tooling at Scale”, O’Reilly Media, Inc., ISBN: 978-1-491-92630-7

3. Sanjeev Sharma and Bernie Coyne, “DevOps for Dummies”, John Wiley & Sons, Inc., 2nd IBM Limited Edition, ISBN: 978-1-119-04705-6

E-Books:

1.https://www.redhat.com/en/resources/cloud-native-container-design-whitepaper

2.https://www.redhat.com/en/topics/automation/what-is-orchestration

3. https://www.atlassian.com/continuous-delivery/continuous-integration

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PEC-361B-IT : Quantum Computing

Teaching Scheme	Credits	Examination Scheme
Theory :3 Hrs/Week	3	CCE - 30 Marks End Sem - 70 Marks

Prerequisite Courses: Data Structure, Data Science

Course Objectives:

- To understand basics of quantum computing.
- To describe qubit representation, quantum states, and measurement models and discuss mathematics required for quantum computing.
- To apply quantum gates and circuit rules to construct simple quantum circuits.
- To analyze multi-qubit systems and basic quantum algorithms and to understand building blocks of quantum computing and design algorithms.
- To develop and test small quantum programs using quantum simulation tools.

Course Outcomes:

- **CO1:** To define key terms such as qubit, superposition, entanglement, and quantum gate.
- **CO2:** To explain quantum state representation using Dirac notation and Bloch sphere.
- **CO3:** To construct and simulate basic quantum circuits using standard gate sets.
- **CO4:** To analyze the working of fundamental quantum algorithms and entangled systems and identify the various quantum algorithms
- **CO5:**To implement simple quantum programs using Qiskit/Cirq and interpret results and to describe usage of tools for quantum computing.

Course Contents

Unit : I	Foundations Of Quantum Computing.	09 Hours

Motivation for studying Quantum Computing, Origin of Quantum Computing, Quantum Computer vs. Classical Computer, Need for quantum computing, Classical bits vs qubits, Basic quantum mechanics concepts for computing as Superposition and Entanglement, Complex numbers and vectors, Dirac notation, Bloch sphere representation.

Case Study: Quantum vs Classical Information — Modeling a Qubit System Using Bloch Sphere

Unit : II	Mathematical Foundations For Quantum Computing, Qubits And Quantum Gates.	09 Hours

Qubit states and normalization, Superposition principle, Measurement and probability, Single qubit gates: X, Y, Z, H, Phase, Rotation gates, Matrix representation of gates, Gate operations on qubits, Matrix Algebra: basis vectors and orthogonality, inner product and Hilbert spaces, matrices and tensors, unitary operators and projectors, Dirac notation, Eigen values and Eigen vectors.

Case Study: Design and Analysis of a Single-Qubit Quantum Random Number Generator (QRNG)

Unit : III	Multi-Qubit Systems, Quantum Circuits And Building	09 Hours

	Blocks For Quantum Program.	
Architecture of a Quantum Computing platform, Details of q-bit system of information representation, Multi-qubits States, Quantum superposition of qubits (valid and invalid superposition, Multi-qubit representation, Tensor product concept, Controlled gates (CNOT, CZ), Entanglement and Bell states, Quantum circuit model, No-cloning theorem, Quantum teleportation. Case Study: Design and Analysis of a Two-Qubit Entangled System for Quantum Teleportation		
Unit : IV	Unit Iv Basic Quantum Algorithms.	09 Hours
Quantum parallelism, Deutsch algorithm, Deutsch–Jozsa algorithm, Grover search algorithm, Shor algorithm, Quantum cryptography basics, Quantum Machine Learning and Quantum AI, Quantum Neural Networks, Quantum Natural Language Understanding, Quantum Cryptography. Case Study: Speedup Through Quantum Algorithms -From Deutsch to Grover		
Unit : V	Quantum Programming & Applications.	09 Hours
Introduction to Qiskit / Cirq, writing basic quantum programs, Circuit simulation, Result visualization, Application Domains for quantum image processing overview, Current challenges and future scope, OSS Toolkits for implementing Quantum program, IBM quantum experience, Microsoft Q Rigetti PyQuil (QPU/QVM). Case Study: Building and Deploying a Quantum Program Using Open-Source Quantum Toolkits		

Learning Resources

Text Books:

1. Michael A. Nielsen, Quantum Computation and Quantum Information||, Cambridge University Press.

2. David McMahon, Quantum Computing Explained, Wiley, 2008.

3. Qiskit textbook https://qiskit.org/textbook-beta/

4. Vladimir Silva, Practical Quantum Computing for Developers, 2018 .

5. Wittek, "Quantum Machine Learning (What Quantum Computing Means to Data Mining)", Peter University of Boras, Sweden - Elsevier Publications

Reference Books:

1. Bernard Zygelman, A First Introduction to Quantum Computing and Information,2018

2. Supriyo Bandopadhyay and Marc Cahy, Introduction to Spintronics||, CRC Press, 2008

3. The Second Quantum Revolution: From Entanglement to Quantum Computing and Other SuperTechnologies, Lars Jaeger

4. La Guardia, Giuliano Gladioli Quantum Error correction codes Springer,2021

5. Press Stephen Kan, "MetricsandModelsinSoftwareQualityEngineering||,Pearson,ISBN-10:0133988082; ISBN-13:978-0133988086 2.

E-Books:

1. https://quantum.cloud.ibm.com/learning/en

2. https://quantum.cloud.ibm.com/docs/en/guides/quick-start

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PEC-361C-ITT : Blockchain Technology

Teaching Scheme	Credits	Examination Scheme
Theory :3 Hrs/Week	3	CCE - 30 Marks EndSem - 70 Marks

Prerequisite Courses: Distributed System, Data Structure

Course Objectives:

- To introduce the concepts of Blockchain and Distributed Ledger Technologies.
- To understand the cryptographic principles used in blockchain systems.
- To study consensus mechanisms and cryptocurrency ecosystems.
- To analyze blockchain platforms and smart contract concepts.
- To examine blockchain applications, challenges and emerging trends.

Course Outcomes:

- **CO1:** Understand the fundamentals of blockchain and distributed ledger technologies.
- **CO2:** Describe cryptographic techniques and transaction verification mechanisms used in blockchain systems.
- **CO3:** Analyze consensus protocols and cryptocurrency operations.
- **CO4:** Analyze blockchain platforms and smart contract architectures.
- **CO5:** Evaluate blockchain applications, challenges, and emerging trends.

Course Contents

Unit : I	Foundations Of Blockchain And Distributed Ledger Technology	09 Hours

Evolution of digital transactions and record keeping systems, centralized, decentralized and distributed systems, peer-to-peer networks, distributed ledger technology, blockchain fundamentals, characteristics of blockchain, structure of a block, blockchain architecture, Merkle trees, types of blockchain (public, private, consortium and hybrid blockchain), advantages and limitations of blockchain.

Case Studies : Blockchain-based Digital Governance.

Unit : II	Cryptographic Foundations Of Blockchain	09 Hours

Fundamentals of cryptography, cryptographic hash functions and their properties, SHA-256, public key cryptography, digital signatures, wallets and blockchain addresses, transaction verification process, integrity, authenticity and non-repudiation in blockchain systems, role of cryptography in blockchain security.

Case Studies : Secure Transaction Processing in Blockchain Networks.

Unit No: III	Consensus Mechanisms And Cryptocurrency Systems	09 Hours

Introduction to cryptocurrencies, Bitcoin architecture, Bitcoin transactions, mining process, double spending problem, Byzantine Generals Problem, Proof of Work (PoW), Proof of Stake (PoS), Ethereum consensus evolution (PoW to PoS),Delegated Proof of Stake (DPoS), Practical Byzantine Fault Tolerance (PBFT), comparison of consensus mechanisms, benefits and limitations of consensus protocols.
Case Studies : Bitcoin Network and Cryptocurrency Ecosystem.

Unit : IV	Blockchain Platforms And Smart Contracts	09 Hours

Overview of Ethereum architecture, Ethereum accounts, Ether and Gas concepts, Ethereum Virtual Machine (EVM), introduction to smart contracts, characteristics and applications of smart contracts, overview of Hyperledger Fabric, consortium blockchain architecture, comparative study of blockchain platforms and enterprise blockchain solutions.
Case Studies : Enterprise Blockchain Applications.

Unit No: V	Blockchain Applications, Challenges And Emerging Trends	09 Hours

Applications of blockchain in banking and finance, supply chain management, healthcare, digital identity management, e-governance, education and certification systems. Challenges in blockchain adoption: scalability, interoperability, privacy, security and regulatory concerns. Introduction to Decentralized Finance (DeFi), Non-Fungible Tokens (NFTs), Central Bank Digital Currency (CBDC), Blockchain and Artificial Intelligence.
Case Studies : Supply Chain Traceability, Academic Certificate Verification

Learning Resources

Text Books:

1. Blockchain Technology by T Chandramouli Subramanian, Asha A. George, Abhilash K. A., Meena Karthikeyan Publisher: Universities Press
2. Blockchain Basics: A Non-Technical Introduction in 25 Steps by Daniel Drescher Publisher: Apress

Reference Books:

1. Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction by Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder
Publisher: Princeton University Press
2. The Science of the Blockchain by Roger Wattenhofer

e-Books:

1. NPTEL, "Blockchain and its Applications" By Prof. Sandip Chakraborty, Prof. Shamik Sural

Link: https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs34

2. NPTEL, "Blockchain Architecture Design an" By Prof. Sandip Chakraborty & Dr. Praveen Jayachandran

Link: https://onlinecourses.nptel.ac.in/e-learning/preview/noc19_cs63

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PEC-361D-ITT **:** Natural Language Processing

Teaching Scheme	**Credits**	**Examination Scheme**
Theory :3 H/W	3	CCE - 30 Marks EndSem - 70 Marks

Prerequisite Courses: Operating System, Data Science

Course Objectives:

- To introduce the fundamental concepts and challenges of Natural Language Processing.
- To study lexical, syntactic and semantic analysis techniques used in NLP.
- To understand statistical approaches and language models for text processing.
- To analyze information extraction and text analytics techniques.
- To interpret modern NLP applications and emerging trends.

Course Outcomes:

- **CO1:** Understand the concepts, challenges and applications of Natural Language Processing.
- **CO2:** Apply text preprocessing and lexical analysis techniques to textual data.
- **CO3:** Analyze syntactic and semantic structures in natural language.
- **CO4:** Analyze statistical approaches and language models used in NLP systems.
- **CO5:** Evaluate NLP applications and recent developments in language technologies.

Course Contents

Unit : I	Introduction To Natural Language Processing	09 Hours

Introduction to Natural Language Processing, applications of NLP, characteristics of natural language, ambiguity in natural language, levels of language processing (morphological, lexical, syntactic, semantic, discourse and pragmatic processing), challenges in NLP, corpus and corpus analysis Case Studies: Intelligent Virtual Assistants and Conversational Systems.

Unit : II	Lexical Analysis And Text Processing	09 Hours

Text preprocessing, tokenization, normalization, stop-word removal, stemming, lemmatization, regular expressions, lexical analysis, lexical resources, dictionaries and thesauri, WordNet, part-of-speech tagging and applications.

Case Studies: Document Classification and Text Processing Systems

Unit : III	Syntactic And Semantic Processing	09 Hours

Context-Free Grammar (CFG), parsing techniques, parse trees, syntactic ambiguity, dependency grammar, semantic representation, word sense disambiguation, semantic similarity, named entity recognition, information extraction techniques.

Case Studies: Information Extraction from Unstructured Text

Unit : IV	Statistical Natural Language Processing	09 Hours

Probabilistic approaches to NLP, N-gram language models, language modeling, Hidden Markov Models, Bayesian approaches, vector space models, Bag-of-Words representation, TF-IDF weighting,

word embeddings, introduction to neural language models.

Case Studies: Sentiment Analysis and Opinion Mining.

Unit : V	NLP Applications And Emerging Trends	07 Hours

Machine translation, information retrieval, question answering systems, text summarization, chatbots and conversational AI, multilingual NLP, transformer architecture (overview), Large Language Models and Retrieval-Augmented Generation (overview), ethical issues in NLP, bias and fairness, future trends and research challenges.

Case Studies: Machine Translation Systems, Conversational AI Platforms

Learning Resources

Text Books:

1. Natural Language Processing and Information Retrieval by Tanveer Siddiqui, U. S. Tiwary Publisher: Oxford University Press
2. Speech and Language Processing by Daniel Jurafsky, James H. Martin

Reference Books:

1. Natural Language Processing with Python by Steven Bird, Ewan Klein, Edward Loper Publisher: O'Reilly Media
2. Foundations of Statistical Natural Language Processing by Christopher Manning, Hinrich Schütze
3. Natural Language Understanding by James Allen

E-Books:

1.https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs180

2. https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs181

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PEC-362-IT **:** Power BI & Tableau

Teaching Scheme	Credits	Examination Scheme
Theory :3 Hrs/Week	3	CCE - 30 Marks End Sem - 70 Marks

Prerequisite Courses:

1.Basic knowledge of Computer Fundamentals

2.Understanding of Database Management Systems (DBMS) concepts

3. Basic knowledge of MS Excel and Spreadsheet operations

Course Objectives:

- To introduce students to Business Intelligence and Data Visualization concepts.
- To develop skills in data preparation, transformation, and modeling using Power BI and Tableau.
- To enable students to create interactive dashboards and reports for decision-making.
- To apply analytical techniques for extracting insights from structured datasets.
- To familiarize students with real-time data visualization applications in industry.

Course Outcomes:

- **CO1:** To Understand the fundamentals of Business Intelligence and Data Analytics tools.
- **CO2:** To Perform data cleaning, transformation, and visualization using Power BI.
- **CO3:** To Design interactive dashboards and analytical reports using Tableau.
- **CO4:** To Apply data analysis techniques for solving business and engineering problems.
- **CO5:** To Develop complete visualization solutions for real-world datasets.

Course Contents

Unit : I	Introduction To Business Intelligence And Data Visualization	09 Hours

Introduction to Business Intelligence (BI): Definition of Business Intelligence Evolution of BI Systems Importance of BI in modern organizations Data-driven decision-making Components of BI Architecture

Concepts of Data Analytics and Visualization: Types of Analytics, Descriptive Analytics, Diagnostic Analytics, Prescriptive Analytics. Introduction to Power BI and Tableau, Types of Charts and Visualizations, Data Sources and Data Import Techniques, Applications of BI in Industry Case Studies: Hospital Management BI Dashboard

Unit : II	Power BI Fundamentals	09 Hours

Power BI Interface and Components: Overview of Microsoft Power BI Business Intelligence using Power BI Applications of Power BI in organizations Power BI ecosystem **Components of Power BI**: Power BI Desktop, Power BI Service (Cloud Platform), Power BI Mobile Applications, Power Query Editor, Data Modeling and Relationships, Creating Charts, Graphs, and

Tables,Filters,Slicers,andDrill-downFeatures. Case Studies: E-Commerce Customer Analytics Dashboard

Unit : III	Advanced Power BI and Dashboard Design	09 Hours

DAX Functions and Calculated Measures: Introduction to DAX (Data Analysis Expressions)

Common DAX Functions: Aggregation (SUM, AVERAGE, COUNTROWS)Logical (IF, SWITCH), Time Intelligence (DATEADD, SAMEPERIODLASTYEAR, TOTALYTD), Filter Functions (CALCULATE, FILTER, ALL) Interactive Dashboard Development and Business Reports, Publishing and Sharing Reports,Real-Time-Data-Monitoring.

Case Studies: Sustainable Waste Management Monitoring (aligned with your research interest)

Unit : IV	Tableau Fundamentals And Visualization Techniques	09 Hours

Introduction to Tableau Environment: Introduction to Business Intelligence using Tableau Features and capabilities of Tableau Tableau Architecture, Tableau Interface Components,

Connecting Data Sources in Tableau: File-Based Data Connections, Database Connections Data Blending and Data Preparation, Worksheets, Dashboards, and Stories, Visualization Techniques in Tableau, Geographical and Statistical Visualizations

Case Studies: Smart City Traffic Analytics Dashboard

Unit : V	Advanced Tableau And Industry Applications	09 Hours

Advanced Calculations and Parameters: Overview of Advanced Calculations in Tableau, Level of Detail (LOD) Expressions, Conditional logic (IF, CASE statements), Interactive Dashboards and Filters, Forecasting and Trend Analysis, Publishing Tableau Reports.

Case Studies: Airline Passenger Trend Forecasting

Learning Resources

Text Books:

1. Learn Power BI: A beginner's guide to developing interactive business intelligence solutions using Microsoft Power BI by Greg Deckler.
2. Tableau Desktop Specialist Guide by Tyler Monroe.

Reference Books:

1. Storytelling with Data: A Data Visualization Guide for Business Professionals Cole Nussbaumer Knaflic ISBN-97811190022532.
2. Practical Tableau: 100 Tips, Tutorials, and Strategies from a Tableau Zen Master, Ryan Sleeper ISBN- 9781491977316.
3. Microsoft Power BI Cookbook: Creating Business Intelligence Solutions of Analytical Data Models, Reports, and Dashboards, Brett Powell ISBN- 9781788290142
4. Data Visualization: Principles and Practice (Second Edition), Alexandru C. Telea ISBN-9781466585263

E-Books:

1.https://onlinecourses.nptel.ac.in/noc26_cs64/preview?utm_source=chatgpt.com

2. https://onlinecourses- archive.nptel.ac.in/noc18_mg34

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PEC-363-IT : Power BI and Tableau Lab

Teaching Scheme	Credits	Examination Scheme
Practical : 2 Hrs/week	01	TW: 25 Marks PR :25 Marks

Prerequisite Courses:

1. Basic knowledge of Computer Fundamentals
2. Understanding of Database Management Systems (DBMS) concepts
3. Basic knowledge of MS Excel and Spreadsheet operations

Course Objectives:

- To provide students with foundational knowledge of Business Intelligence and Data Visualization concepts.
- To build competency in data preparation, transformation, and modeling using tools such as Power BI and Tableau.
- To enable students to design and develop interactive dashboards and reports that support effective decision-making.

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

- CO1: To provide students with a clear understanding of the fundamentals of Business Intelligence and Data Analytics tools.
- CO2: To enable students to perform data cleaning, transformation, and visualization using Power BI effectively.
- CO3: To equip students with the ability to design interactive dashboards and analytical reports using Tableau.
- CO4: To develop skills in applying data analysis techniques to address business and engineering problems.

Guidelines for Instructor's Manual

The instructor ‘s manual is to be developed as a hands-on resource and reference. The instructor's manual need to include prologue (about university/program/ institute/ department/foreword/ preface etc.), University syllabus, conduction & Assessment guidelines.

Guidelines for Student's Lab Journal

1. The laboratory assignments are to be submitted by student in the form of journal.
2. Journal consists of prologue, Certificate, table of contents, and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory- Power BI and Tableau feature/Concept in brief, conclusion/analysis.
3. Program codes with sample output of all performed assignments are to be submitted as hardcopy.

4. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal may be avoided.
5. Use of DVD containing students' programs maintained by lab In-charge is highly encouraged.
6. For reference one or two journals may be maintained with program prints at Laboratory.

Guidelines for Lab /TW Assessment

1. Examiners will assess the student based on performance of students considering the parameters such as timely completion of practical assignment, methodology adopted for implementation of practical assignment, timely submission of assignment in the form of handwritten write-up along with results of implemented assignment, attendance etc.
2. Examiners will judge the understanding of the practical performed in the examination by asking some questions related to theory and implementation of experiments he/she has carried out.
3. Appropriate knowledge of usage of software and hardware related to respective laboratory should be checked by the concerned faculty member.

Guidelines for Practical Examination

1.Examiners will assess the student based on performance of students considering the parameters such as timely completion of practical assignment, methodology adopted for implementation of practical assignment, timely submission of assignment in the form of handwritten write-up along with results of implemented assignment, attendance etc.
2. Examiners will judge the understanding of the practical performed in the examination by asking some questions related to theory and implementation of experiments he/she has carried out.
3. Appropriate knowledge of usage of software and hardware related to respective laboratory should be checked by the concerned faculty member**.**

Guidelines for Laboratory Conduction

The instructor is expected to frame the laboratory assignments by understanding the prerequisites, technological advancements, industrial applications, utility, and recent trends in the field of Business Intelligence (BI), Data Analytics, and Data Visualization. The assignment design should cater to average learners while incorporating advanced challenges to motivate and engage academically strong students.

The instructor may create multiple sets of assignments with equivalent complexity levels and distribute them among different student batches to encourage originality and reduce duplication of work.

Students should be encouraged to follow industry best practices for dashboard development, including appropriate naming conventions, documentation of calculated fields, data modeling standards, visualization design principles, and proper commenting of formulas and business logic wherever applicable.

The use of open-source datasets, publicly available data repositories, and free educational software versions is highly encouraged.

List of Assignments

1. Installation Of Power BI/Tableau(CO1, CO2)

Install and configure Power BI/Tableau environment. Introduction to Business Intelligence, data visualization, and software environment setup.

2. Data Import and Exploration in Power BI- CO1, CO3

Import a dataset into Power BI and explore data fields, types, and initial visualization preview. Refer this following data set link https://www.kaggle.com/datasets/apoorvaappz/global-super-store-dataset?utm_source=chatgpt.com

3.Data Cleaning and Transformation Using Power Query Editor- CO1, CO3

Data cleaning and transformation using Power Query Editor, Clean raw retail data by handling missing values, duplicates, incorrect formats, and transform it for analysis. Refer this following data set link https://www.kaggle.com/datasets/yasserh/titanic-dataset?utm_source

4. Data Modeling and DAX Measures in Power BI- CO1, CO3

Create relationships between tables and design basic DAX measures like Total Sales, Profit, and Quantity in Power BI. Refer this following data set link https://www.kaggle.com/datasets/ukveteran/adventure-works?utm_source

5. Interactive Reports Using Charts, Filters, and Drill-Down in Power BI(CO1, CO3**)**

Creating charts, tables, slicers, filters, and drill-down reports in Power BI. Refer this following data set link https://www.kaggle.com/datasets/vivek468/superstore-dataset-final?utm_source

6. Interactive Sales Dashboard Design and Publishing Using Power BI Service - CO4

Design an interactive dashboard for sales performance and publish it using Power BI Service. Refer this following data set link https://www.kaggle.com/datasets/carrie1/ecommerce-data?utm_source

7. Tableau Data Connection and Worksheet Basics- CO4

Connect Tableau to a dataset and create basic worksheets with dimensions and measures. Refer this following data set link https://www.kaggle.com/datasets/worldbank/world-development-indicators?utm_source

8. Multi-Sheet Dashboard Design in Tableau – CO4

Build interactive dashboards using filters, parameters, and multiple visual sheets. Refer this following data set link https://www.kaggle.com/datasets/airbnb/seattle?utm_source

9. Forecasting, Trend Analysis, and Business Case Visualization in Tableau – CO4

Forecasting, trend analysis, and business case-based visualization in Tableau. Refer this following data set link https://www.kaggle.com/datasets/vivek468/superstore-dataset-final?utm_source

10.Mini-Project in Tableau /Power BI – CO4

Mini project: end-to-end dashboard/report development using Power BI or Tableau on a real dataset. Data set link https://www.kaggle.com/datasets/shivamb/netflix-shows?utm_source

Note: Group of 4-5 students should select a real-world dataset and develop a complete visualization solution using Power BI or Tableau. The project should include data cleaning, transformation, visualization, dashboard creation, short presentation and students should submit the document related to it as part of journal of insights.
Text Books:
1. Learn Power BI: A beginner's guide to developing interactive business intelligence solutions using Microsoft Power BI by Greg Deckler. 2.Tableau Desktop Specialist Guide by Tyler Monroe.
Reference Books:
1. Storytelling with Data by Cole Nussbaumer Knaflic. 2.Practical Tableau by Ryan Sleeper. 3.Microsoft Power BI Cookbook by Brett Powell. 4.Data Visualization: Principles and Practice by Alexandru C. Telea.

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PEC-362B-IT : Elective III High Performance Computing

Teaching Scheme	Credits	Examination Scheme
Theory :3 Hrs/Week	3	CCE - 30 Marks End Sem - 70 Marks

Prerequisite Courses: Computer Architecture, Operating Systems, Programming in C/C++

Course Objectives:

- To understand the fundamental concepts of parallel computing, architecture, and high-performance system design.
- To study parallel programming models including OpenMP, MPI, CUDA, and OpenCL for shared and distributed memory systems.
- To analyze performance metrics, parallel algorithms, and optimization techniques for HPC workloads.
- To explore emerging HPC trends including GPU computing, cloud HPC, quantum computing, and AI/ML accelerators.

Course Outcomes:

- **CO1:** Explain parallel computing architectures (SIMD, MIMD, shared/distributed memory) and evaluate their performance characteristics.
- **CO2:** Develop parallel programs using OpenMP (shared memory) and MPI (distributed memory) paradigms.
- **CO3:** Implement GPU-accelerated solutions using CUDA/OpenCL for compute-intensive applications.
- **CO4:** Analyze, measure, and optimize the performance of parallel programs using profiling and benchmarking tools.

Course Contents

Unit : I	Introduction To Parallel Computing And Hpc Architectures	09 Hours

Overview of High Performance Computing: motivation, history, and applications (scientific simulations, Big Data, AI/ML)

Flynn's Taxonomy: SISD, SIMD, MISD, MIMD; Shared Memory vs. Distributed Memory systems

Parallel Computer Architecture: multicore processors, cache coherence (MESI protocol), NUMA architecture

Interconnection Networks: bus, crossbar, mesh, hypercube, fat-tree topologies; bandwidth and latency

Performance Metrics: Speedup (Amdahl's Law, Gustafson's Law), Efficiency, Scalability, Flops, TOP500 benchmarks

Case Studies: LINPACK benchmark and Exascale computing – Frontier Supercomputer .

Unit : II	Shared Memory Programming with OpenMP	09 Hours

Introduction to Thread-based Parallelism: processes vs. threads, POSIX threads (pthreads) overview

OpenMP Programming Model: fork-join model, compiler directives (#pragma omp), execution model

Work Sharing Constructs: parallel for, sections, single; schedule clauses (static, dynamic, guided)

Synchronization: critical, atomic, barrier, ordered; data environment: shared, private, firstprivate, reduction, Nested Parallelism, Tasks (OpenMP 3.0+), SIMD directives, and Thread-Affinity

Case Studies: OpenMP in HPC weather forecasting (WRF model).

Unit : III	Distributed Memory Programming with MPI	09 Hours

Distributed Memory Model: message passing concept, MPI overview, program structure and compilation (mpicc/mpirun)

Point-to-Point Communication: MPI_Send, MPI_Recv, blocking vs. non-blocking (MPI_Isend, MPI_Irecv), MPI_Sendrecv

Collective Communication: MPI_Bcast, MPI_Scatter, MPI_Gather, MPI_Allreduce, MPI_Barrier, MPI_Reduce ,MPI Communicators, Groups, Virtual Topologies (Cartesian grids),Derived-Datatypes

Case Studies: MPI-based parallel genome assembly (BLAST onHPC clusters).

Unit No: IV	GPU Computing with CUDA and OpenCL	09 Hours

GPU Architecture: NVIDIA GPU architecture (SM, CUDA cores, warp, warp scheduler), memory hierarchy (global, shared, registers, constant, texture)

CUDA Programming Model: host-device model, kernels, threads/blocks/grids, thread indexing, CUDA execution model

CUDA Memory Management: cudaMalloc, cudaMemcpy, unified memory, memory coalescing, shared memory optimization CUDA Optimization: occupancy, warp divergence, memory bank conflicts, streams and asynchronous execution, CUDA events for profiling OpenCL Framework: platform model, execution model, memory model

Case Studies: Deep Learning training acceleration using CUDA and cuDNN (ResNet on CIFAR-10).

Unit No: V	High Performance Computing Applications	09 Hours

Scope of Parallel Computing, **Parallel Search Algorithms**: Depth First Search(DFS), Breadth First Search(BFS), Parallel Sorting: Bubble and Merge, **Distributed Computing**: Document classification, **Frameworks** – Kuberbets, GPU Applications, Parallel Computing forAI/ML.Case Studies: Disaster detection and management/ Smart Mobility/Urban planning

Learning Resources

Text Books:

1. Peter Pacheco, Matthew Malensek, "An Introduction to Parallel Programming", 2nd Edition, Morgan Kaufmann (Elsevier), 2021. ISBN: 978-0128046050

2. Kirk, D.B. and Hwu, W.W., "Programming Massively Parallel Processors: A Hands-on Approach",

4th Edition, Morgan Kaufmann, 2022. ISBN: 978-0323912310

Reference Books:

1. Ananth Grama, Vipin Kumar, Anshul Gupta, George Karypis, "Introduction to Parallel Computing", 2nd Edition, Pearson Education, 2003. ISBN: 978-0201648652

2. Michael J. Quinn, "Parallel Programming in C with MPI and OpenMP", McGraw-Hill Education, 2004. ISBN: 978-0072822564

3. Thomas Rauber, Gudula Runger, "Parallel Programming for Multicore and Cluster Systems", 3rd Edition, Springer, 2023. ISBN: 978-3662653005

4. Rob Farber, "CUDA Application Design and Development", Morgan Kaufmann, 2011. ISBN: 978-0123884268

E-Books:

1.https://nptel.ac.in/courses/106/105/106105191

2.https://docs.nvidia.com/cuda/cuda-c-programming-guide/

3. https://www.openmp.org/specifications/

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PEC-363-IT: Elective III Lab 3(HPC)

Teaching Scheme	Credits	Examination Scheme
Practical : 2 Hrs/week	1	TW: 25 Marks PR :25 Marks

Prerequisite Courses: Computer Architecture, Operating Systems, Programming in C/C++

Course Objectives:

- To provide hands-on experience with parallel programming using OpenMP and MPI on multi-core systems and clusters.
- To enable students to write, compile, run, and profile CUDA programs on GPU hardware.
- To apply parallel algorithms for real-world problem solving and measure performance improvements.
- To develop skills in benchmarking, profiling, and optimization of HPC applications.

Course Outcomes:

- CO1: Set up and run parallel programs on multi-core and distributed computing environments.
- CO2: Implement parallel algorithms using OpenMP, MPI, and CUDA, and measure their performance.
- CO3: Debug, profile, and optimize parallel programs using tools such as gprof, Valgrind, nvprof/Nsight.
- CO4: Design and present a mini-project involving HPC concepts and parallel programming techniques.

Guidelines for Instructor's Manual

The instructor's manual must include university syllabus, conduction and assessment guidelines, concept and objectives for each experiment, algorithm/pseudocode, sample test cases, expected output, and references. Labs should be conducted on Linux (64-bit) systems with multi-core processors. CUDA experiments require an NVIDIA GPU with CUDA Toolkit installed.

Guidelines for Student's Lab Journal

1. The laboratory assignments are to be submitted by student in the form of journal.
2. Journal consists of prologue, Certificate, table of contents, and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory- HPC feature/Concept in brief, algorithm, flowchart, test cases, conclusion/analysis.
3. Program codes with sample output of all performed assignments are to be submitted as hardcopy.
4. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal may be avoided.

5. Use of DVD containing students programs maintained by lab In-charge is highly encouraged.
6. For reference one or two journals may be maintained with program prints at Laboratory.

Guidelines for Lab /TW Assessment

1. Continuous assessment of laboratory work is done based on overall performance and lab assignments performance of student.
2. Each lab assignment assessment will assign grade/marks based on parameters with appropriate weightage.
3. Suggested parameters for overall assessment as well as each lab assignment assessment include- timely completion, performance, innovation, efficient codes, punctuality .

Guidelines for PR Examination

1.Both internal and external examiners should jointly set problem statements for practical examination.
2.During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement.
3. The supplementary and relevant questions may be asked at the time of evaluation to judge the student 's understanding of the fundamentals, effective and efficient implementation. The evaluation should be done by both external and internal examiners.

Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy need to address the average students and inclusive of an element to attract and promote the intelligent students.

The instructor may set multiple sets of assignments without changing its complexity level and distribute among batches of students. Encourage students for the use of industry coding standards such as appropriate use of Hungarian notation, Indentation and comments. Use of open-source software is encouraged. Set of suggested assignment list is provided, instructors may take different case studies with similar complexity level. Operating System recommended: -

64-bit Ubuntu Linux (20.04/22.04 LTS). Tools: GCC with OpenMP (-fopenmp), MPICH/OpenMPI, CUDA Toolkit (11.x/12.x), NVIDIA Nsight, Intel VTune Profiler (optional). Virtual Lab: https://hpc.vlabs.ac.in/ . Use of open-source tools is encouraged.

List of Assignments

1. OpenMP – Parallel Loop and Work Sharing – CO1, CO2

Implement the following using OpenMP on a multi-core system: a) Parallel computation of sum of an array of N integers using parallel for with reduction. b) Matrix addition and matrix multiplication of 1000x1000 matrices. c) Compare execution time (omp_get_wtime) for 1, 2, 4, and 8 threads. d) Calculate Speedup and Efficiency for each configuration and plot the graph.

2. OpenMP – Synchronization and Critical Sections – CO1, CO2
Implement a multi-threaded bank account simulation using OpenMP: a) Simulate 1000 concurrent deposit and withdrawal operations using critical sections. b) Implement the same using atomic directives and compare performance. c) Demonstrate race condition without synchronization and its fix with proper OpenMP constructs. d) Measure overhead introduced by synchronization.
3. MPI – Point-to-Point Communication – CO1, CO2
Implement the following using MPI: a) Master-Slave communication: Master broadcasts an array and each slave computes sum of its chunk. b) Ring communication: each process sends a token to the next (ring topology) and measure latency. c) Parallel computation of Pi using numerical integration (MPI_Send/MPI_Recv). d) Compare blocking vs non-blocking communication performance.
4. MPI – Collective Communication and Parallel Sorting – CO1, CO2
Implement the following using MPI collective operations: a) Parallel matrix multiplication with data distribution using MPI_Scatter and MPI_Gather. b) MPI_Allreduce to compute global minimum, maximum, and average of a distributed array. c) Implement parallel Odd-Even Transposition Sort using MPI. d) Measure and report speedup with 2, 4, and 8 processes
5. CUDA – Vector Operations and Kernel Programming – CO2, CO3
Implement the following using MPI collective operations: a) Parallel matrix multiplication with data distribution using MPI_Scatter and MPI_Gather. b) MPI_Allreduce to compute global minimum, maximum, and average of a distributed array. c) Implement parallel Odd-Even Transposition Sort using MPI. d) Measure and report speedup with 2, 4, and 8 processes
6. CUDA – Memory Optimization and Profiling – CO3, CO4
Optimize and profile CUDA programs: a) Implement 2D convolution (image blurring) using global memory and optimize with shared memory + halo cells. b) Use NVIDIA Nsight Compute (or nvprof) to profile a kernel: analyze memory bandwidth, occupancy, warp efficiency. c) Demonstrate memory coalescing: compare coalesced vs. non-coalesced memory access patterns. d) Report achieved bandwidth vs. peak theoretical bandwidth of the GPU.
7. Performance Benchmarking and Profiling – CO4
Benchmark and profile parallel programs: a) Use HPL (High-Performance Linpack) benchmark on a multi-core machine and report GFlops. b) Profile a serial vs. OpenMP program using gprof and Valgrind; identify hotspots. c) Implement a parallel N-body gravitational simulation (simplified) in OpenMP and MPI variants; compare performance. d) Generate and interpret performance plots (speedup, efficiency) for different problem sizes (weak and strong scaling).
8. Hybrid MPI+OpenMP and Cloud HPC – CO1, CO2, CO4
Explore hybrid and cloud-based HPC: a) Implement a hybrid MPI+OpenMP parallel program for matrix-vector multiplication; test on a 2-node, 4-thread-per-node setup. b) Compare performance of pure MPI, pure OpenMP, and hybrid MPI+OpenMP for the same problem. c) Use Google Colab or AWS Free Tier to run a CUDA program on a cloud GPU; document setup steps, execution, and cost comparison. d) Explore

NVIDIA RAPIDS cuDF library for GPU-accelerated data processing (demonstrate on a CSV dataset).

Learning Resources:

Text Books:

1. Peter Pacheco, Matthew Malensek, "An Introduction to Parallel Programming", 2nd Edition, Morgan Kaufmann, 2021.

2. Kirk, D.B. and Hwu, W.W., "Programming Massively Parallel Processors: A Hands-on Approach", 4th Edition, Morgan Kaufmann, 2022.

Reference Books:

1. Michael J. Quinn, "Parallel Programming in C with MPI and OpenMP", McGraw-Hill Education, 2004.

2. Thomas Rauber, Gudula Runger, "Parallel Programming for Multicore and Cluster Systems", 3rd Ed., Springer, 2023.

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PEC-362C-IT: Elective III - Mobile Application Development

Teaching Scheme	Credits	Examination Scheme
Theory :3 Hours/Week	03	CCE -30 Marks ESE - 70 Marks

Prerequisite Courses: Object Oriented Programming,Database Management Systems

Course Objectives:

1. To understand mobile computing concepts, architectures, and mobile application development ecosystems.
2. To study principles of mobile user interface and user experience design.
3. To understand mobile application components and application development frameworks.
4. To learn techniques for data management, web service integration, and cloud connectivity in mobile applications.
5. To study advanced mobile application features, security mechanisms, testing, and deployment practices.

Course Outcomes:

- **CO1:** Explain mobile computing concepts, mobile platforms, architectures, and application lifecycle models.
- **CO2:** Apply principles of user interface and user experience design for mobile applications.
- **CO3:** Explain and analyze mobile application components, navigation mechanisms, and event-driven programming models.
- **CO4:** Explain data management techniques, web service integration, and cloud-based mobile application architectures.
- **CO5:** Apply security, testing, optimization, and deployment concepts in mobile application development.

Course Contents

Unit No: I	Introduction to Mobile Computing and Android Platform	09 Hours

Introduction to Mobile Computing, Evolution of Mobile Devices, Characteristics of Mobile Computing, Mobile Computing Architecture, Mobile Operating Systems Overview, Android and Cross-Platform Ecosystems, Mobile Application Architecture, Flutter/Kotlin Development Environment, Project Structure, Build Systems, Application Lifecycle, Introduction to Dart/Kotlin Programming.

Case Studies: Android vs iOS Architecture Comparison

Unit No: II	User Interface Design and User Experience	09 Hours

Widgets and GUI Components, Layout Management, Themes, Fonts and Colors, Material Design Principles, Event Handling Concepts, Menus and Dialogs, Navigation Components, Accessibility Guidelines, Responsive Design, Cross-Platform UI Design Principles, User Experience Considerations.

Case Studies: Accessibility Features in Healthcare Applications.

Unit No: III	Mobile Application Components and Application Logic	09 Hours

Activities and Screens, Fragments and Widgets, Intents and Navigation, Application State Management, Services, Background Tasks, Broadcast Receivers, Notifications, Gesture Handling Concepts, Animation Concepts, Application Workflow Design, Event-Driven Programming-Models.

Case Studies: Food Delivery Application Workflow

Unit No: IV	Data Management and Service Integration	09 Hours

Shared Preferences, Local Storage, SQLite Database, Data Persistence Libraries, JSON Processing, RESTful Web Services, HTTP Communication, API Integration, Authentication Services, Cloud Databases, Cloud Storage, Data Synchronization Techniques, Mobile Backend Services.

Case Studies: Expense Tracking Application, Smart Campus Mobile Portal

Unit No: V	Advanced Mobile Application Development	09 Hours

Location-Based Services, OpenStreetMap and Map Integration, Sensors and Device Features, Camera Integration, Push Notification Services, Mobile Security Concepts, Secure Coding Practices, Mobile Application Testing, Unit Testing, UI Testing, Performance Optimization, Application Packaging, Deployment Concepts, Mobile Application Publishing Process.

Case Studies: Emergency Response and Navigation Applications.

Learning Resources :

Text Books:

1. Bill Phillips and Chris Stewart, Android Programming: The Big Nerd Ranch Guide
2. Dawn Griffiths and David Griffiths, Head First Android Development

Reference Books:

1. Reto Meier, Professional Android
2. Ian F. Darwin, Android Cookbook
3. Antonio Leiva, Kotlin for Android Developers
4. Barry Burd, Android Application Development All-in-One For Dummies
5. Muhammad Usama Mirza, Mastering Flutter: Build Enterprise-Grade, Modular, and Scalable Cross-Platform Applications

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PEC-362C-IT : Elective III Lab 3 - Mobile Application Development Lab

Teaching Scheme	Credits	Examination Scheme
Practical : 2 Hrs/week	1	PR : 25 Marks TW : 25 Marks

Prerequisite Courses : Object Oriented Programming,Database Management Systems

Course Objectives:

1. To familiarize students with Flutter/Kotlin multi-platform application development environment.
2. To design and develop mobile applications using modern UI components and event-driven programming.
3. To implement mobile applications with networking and web service integration.
4. To develop location-aware and notification-enabled mobile applications.
5. To build complete multi-platform mobile applications using open-source technologies.

Course Outcomes:

- **CO1:** Configure and use Flutter/Kotlin multi-platform development tools for mobile application development.
- **CO2:** Design responsive mobile user interfaces using widgets, layouts, fonts, colors, and event-driven programming concepts.
- **CO3:** Develop mobile applications incorporating animations, gesture handling, and user interaction mechanisms.
- **CO4:** Integrate web services, APIs, and push notification mechanisms using HTTP communication and related technologies.
- **CO5:** Design and develop complete multi-platform mobile applications integrating location-based services and other advanced mobile application features using open-source frameworks.

Guidelines for Instructor's Manual

The faculty member should prepare the laboratory manual for all the experiments and it should be made available to students and laboratory instructor/Assistant.

Guidelines for Student's Lab Journal

1. Student should submit term work in the form of handwritten journal based on specified list of assignments.
2. Practical Examination will be based on the term work.
3. Candidate is expected to know the theory involved in the experiment.
4. The practical examination should be conducted if and only if the journal of the candidate is complete in all aspects.

Guidelines for Lab /TW Assessment

1. Examiners will assess the term work based on performance of students considering the parameters such as timely conduction of practical assignment, methodology adopted for implementation of practical assignment, timely submission of assignment in the form of handwritten write-up along with results of implemented assignment, attendance etc.
2. Examiners will judge the understanding of the practical performed in the examination by asking some questions related to the theory & implementation of the experiments he/she has carried out.
3. Appropriate knowledge of usage of software and hardware related to respective laboratory should be checked by the concerned faculty member.

Guidelines for Laboratory Conduction

As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers of the program in journal may be avoided. There must be hand-written write-ups for every assignment in the journal. The DVD/CD containing student's programs should be attached to the journal by every student and same to be maintained by department/lab In-charge is highly encouraged. For reference one or two journals may be maintained with program prints at Laboratory

List of Assignments

1. Flutter/Kotlin Development Environment Setup – CO1

Study and installation of Flutter/Kotlin multi-platform development environment. Configure emulator/device and execute a sample mobile application.

2. Mobile UI Design using Widgets – CO2

Develop an application that uses widgets, GUI components, fonts, colors, layouts, and themes to create a responsive user interface.

3. Native Calculator Application – CO2, CO3

Develop a calculator application supporting basic arithmetic operations using event-driven programming.

4. Mobile Gaming Application – CO3

Develop a gaming application that uses 2-D animations and gesture handling techniques.

5. Movie Rating Application – CO2, CO4

Develop a movie rating application (similar to IMDB) displaying movie details, ratings, and reviews using appropriate UI components.

6. Web Service Integration – CO4

Develop an application to connect to a web service and retrieve data using HTTP communication and JSON processing.

7. Shopping Application – CO2, CO4

Develop a simple shopping application with product listing, navigation, and data management features.

8. Push Notification Service – CO4, CO5
Design and implement a web server supporting push notifications for mobile applications.
9. Location-Based Mobile Application – CO5
Develop an application by integrating location and map services using open-source mapping technologies.
10. Mini Project – CO1, CO2, CO3, CO4, CO5
Develop a complete multi-platform mobile application integrating multiple concepts such as UI design, networking, notifications, location services, and data management.
Text Books:
1. Bill Phillips and Chris Stewart, Android Programming: The Big Nerd Ranch Guide 2. Dawn Griffiths and David Griffiths, Head First Android Development 3. Alberto Miola, Flutter Complete Reference
Reference Books:
1. Reto Meier, Professional Android 2. Ian F. Darwin, Android Cookbook 3. Antonio Leiva, Kotlin for Android Developers 4. Barry Burd, Android Application Development All-in-One For Dummies 5. Muhammad Usama Mirza, Mastering Flutter: Build Enterprise-Grade, Modular, and Scalable Cross-Platform Applications

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PEC-362D-IT : Cyber Security & Risk Management

Teaching Scheme	Credits	Examination Scheme
Theory :3 Hrs/Week	3	CCE - 30 Marks End Sem - 70 Marks

Prerequisite Courses: Computer Networks, Operating System

Course Objectives:

- To introduce the fundamental concepts, principles, and challenges of cyber security and information security.
- To develop an understanding of cyber threats, attacks, vulnerabilities, and security controls used to protect digital assets.
- To impart knowledge of network security, application security, cryptographic techniques, and authentication mechanisms.
- To enable students to identify, assess, and respond to security incidents using appropriate security practices.
- To provide knowledge of cyber risk management, governance, compliance, business continuity, and disaster recovery planning.

Course Outcomes:

- **CO1:** To explain the fundamental concepts of cybersecurity, information security principles, threats, vulnerabilities, and security controls.
- **CO2:** To analyze various cyber threats, attacks, and vulnerabilities and recommend suitable protection mechanisms.
- **CO3:** To apply network security, application security, cryptographic techniques, and authentication mechanisms to enhance system security.
- **CO4:** To perform basic security assessment activities and implement appropriate incident response procedures for security incidents.
- **CO5:** To evaluate cybersecurity risks and apply governance, compliance, business continuity, and disaster recovery strategies to improve organizational resilience.

Course Contents

Unit No: I	Introduction To Cyber Security And Security Principles	09 Hours

Introduction to Cyber Security: Need, Importance and Challenges of Cyber Security, Cyberspace and Cyber Ecosystem, Information Security vs Cyber Security, Security Goals and Principles, CIA Triad, Cyber Threat Landscape.

Security Concepts: Assets, Threats, Vulnerabilities, Exploits, Attack Surface, Security Controls, Security Policies and Procedures.

Cyber Security Frameworks and Standards: Overview of Security Governance, Security Awareness and Training, Introduction to Cyber Security Frameworks and Best Practices.

Case Studies: Analysis of a major cyber attack and its impact on an organization.

Unit No: II	Cyber Threats, Attacks and Security Controls	09 Hours

Cyber Threats and Attacks: Malware (Virus, Worm, Trojan Horse, Ransomware, Spyware), Phishing, Social Engineering, Password Attacks, Denial-of-Service (DoS) and Distributed DoS Attacks.

Web and System Attacks: SQL Injection, Cross-Site Scripting (XSS), Man-in-the-Middle Attack, Session Hijacking, Insider Threats.

Security Controls: Administrative, Technical and Physical Controls, Access Control Mechanisms, Security Monitoring and Logging.

Emerging Threats: Cloud Security Threats, Mobile Security Threats, IoT Security Challenges. Case Studies: Investigation of a ransomware attack and mitigation strategies.

Unit No: III	Network and Application Security	09 Hours

Network Security Fundamentals: Network Security Architecture, Secure Network Design, Common Network Vulnerabilities.

Network Security Mechanisms: Firewalls, Proxy Servers, Virtual Private Networks (VPNs), Intrusion Detection and Prevention Systems (IDS/IPS), Network Segmentation.

Application Security: Secure Software Development Life Cycle (SSDLC), Secure Coding Practices, Vulnerability Assessment and Penetration Testing Concepts.

Data Security: Data Classification, Data Protection Techniques, Backup and Recovery Mechanisms. Case Studies: Security assessment of a web application and network infrastructure

Unit No: IV	Cryptography, Authentication and Incident Response	09 Hours

Cryptography Fundamentals: Principles of Cryptography, Encryption and Decryption, Symmetric and Asymmetric Cryptography.

Cryptographic Techniques: Hash Functions, Digital Signatures, Public Key Infrastructure (PKI), Digital Certificates.

Authentication and Access Management: Authentication Methods, Multi-Factor Authentication (MFA), Authorization and Identity Management.

Incident Response: Security Incident Lifecycle, Detection, Containment, Eradication, Recovery, Incident Reporting and Documentation.

Case Studies: Analysis of a security breach and incident response process

Unit No: V	Cyber Risk Management and Security Governance	09 Hours

Cyber Risk Management: Risk Concepts, Risk Identification, Risk Assessment, Risk Analysis, Risk Evaluation and Risk Prioritization. **Risk Treatment Strategies:** Risk Avoidance, Risk Mitigation, Risk Transfer and Risk Acceptance, Risk Register and Risk Monitoring. **Governance and Compliance:** Security Governance Principles, Cyber Laws and Regulations, Compliance Requirements, Ethics in Cyber Security.

Business Continuity and Disaster Recovery: Business Continuity Planning (BCP), Disaster Recovery Planning (DRP), Backup Strategies, Recovery Objectives and Testing of Recovery Plans.

Case Studies: Risk assessment and business continuity planning for an organization

Learning Resources

Text Books:

1. Charles J. Brooks, Christopher Grow, Philip Craig Jr., Donald Short, “Cyber Security Essentials”, McGraw-Hill Education.

2. Michael E. Whitman, Herbert J. Mattord, “Principles of Information Security”, Cengage Learning.

Reference Books:

1. William Stallings, Lawrie Brown, “Computer Security: Principles and Practice”, Pearson Education.

2. William Stallings, “Cryptography and Network Security: Principles and Practice”, Pearson Education.

3. Mark Stamp, “Information Security: Principles and Practice”, Wiley India.

4. Michael E. Whitman, Herbert J. Mattord, “Risk Management and Information Security”, Cengage Learning.

e-Books:

1. : https://onlinecourses.nptel.ac.in/noc23_cs75/preview

2 : https://onlinecourses.nptel.ac.in/noc24_cs121/preview

3 : https://onlinecourses.nptel.ac.in/noc25_cs120/preview

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

PEC 363-IT : Cyber Security and Risk Management Lab

Teaching Scheme	Credits	Examination Scheme
Practical: 2 Hrs/week	1	TW : 25 Marks PR :25 Marks

Prerequisite Courses: Operating Systems, and Python Programming concepts.

Course Objectives:

- To provide practical exposure to cybersecurity concepts, threats, vulnerabilities, and security mechanisms.
- To develop skills in using cybersecurity tools and techniques for network monitoring and security assessment.
- To implement cryptographic techniques and authentication mechanisms for securing information systems.
- To analyse cybersecurity risks and apply mitigation, business continuity, and disaster recovery strategies.

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

- CO1: To identify cybersecurity threats, vulnerabilities, and risks in computing environments.
- CO2: To utilise cybersecurity tools for network monitoring, traffic analysis, and security assessment.
- CO3: To implement cryptographic algorithms, hashing techniques, and authentication mechanisms.
- CO4: To analyze cyber security incidents and evaluate risk mitigation and recovery strategies.

Guidelines for the Instructor's Manual

The faculty member should prepare the laboratory manual for all the experiments and make it available to students and the laboratory instructor/assistant. The instructor's manual should include a prologue, university syllabus, laboratory conduct and assessment guidelines, course objectives, course outcomes, prerequisite knowledge, software and hardware requirements, detailed procedure of experiments, sample outputs, viva questions, references, and mini-project guidelines. Experiments should be conducted using Python and open-source cybersecurity tools such as Wireshark, Nmap, OpenSSL, and OWASP Juice Shop.

Guidelines for Students' Lab Journal

1. The laboratory assignments are to be submitted by students in the form of journals. The journal should consist of a prologue, a certificate, a table of contents, and a handwritten write-up of each assignment, including: Title, Objectives, Problem Statement, Course Outcome Mapping, Software and Hardware Requirements, Date of Completion, Assessment Grade/Marks and Assessor's Signature, Theory Concept, Algorithm/Procedure, Program Code (where applicable), Sample Input and Output, Observations and Conclusion

2. To support Go-Green initiatives, hard-copy printouts are discouraged. Soft copies of programs, screenshots, and reports should be maintained and submitted to the concerned faculty member.

Practical examination will be based on the Cyber Security and Risk Management theory and laboratory assignments.

3. Students are expected to know the theoretical concepts, implementation details, and practical applications related to each experiment.

4. The practical examination should be conducted only if the journal is completed in all respects and certified by the concerned faculty member and Head of Department.

5. All the assignments prescribed in the syllabus must be performed and documented.

Guidelines for Lab /TW Assessment

1. Examiners will assess the term work based on: Timely completion of practical assignments, Methodology adopted for implementation, Correctness of results, Quality of documentation, Attendance and participation, Timely submission of laboratory journal

2. Examiners will evaluate the student's understanding of cyber security concepts, tools, risk assessment techniques, and practical implementation through oral examination.

3. Appropriate knowledge of software tools, programming practices, coding standards, security concepts, and analytical skills should be assessed by the concerned faculty member(s).

Guidelines for Lab Examination

Both internal and external examiners should jointly conduct the practical examination. During assessment, examiners should give maximum weightage to: Correct understanding of the problem statement, Practical implementation skills, Use of cyber security tools, Interpretation and analysis of results, Knowledge of risk management concepts, The practical examination should focus on effective, secure, and efficient implementation of the assigned task along with proper documentation and analysis.

Guidelines for Laboratory Conduction

The instructor is expected to frame assignments by considering prerequisites, current cyber security practices, industrial relevance, and emerging trends in cyber security and risk management. The instructor may assign different case studies, risk assessment scenarios, and mini-projects to different batches of students.

Assignments should be based on real-world cyber security problems such as password security, network traffic analysis, vulnerability assessment, authentication mechanisms, cryptography, incident response, and cyber risk management.

All laboratory assignments should preferably be conducted using open-source software and tools.

List of Assignments

1. Cyber Security Risk Assessment: CO1, CO5

Identify assets, threats, vulnerabilities, and risks in a given organizational scenario. Prepare a risk assessment report and recommend suitable mitigation strategies.

2. Password Strength Checker: CO2, CO3

Implement a Python program to evaluate password strength based on length, complexity, and character combinations. Generate recommendations for creating secure passwords.

3. Classical Cryptography Techniques: CO3, CO4

Implement encryption and decryption using:

- Caesar Cipher
- Playfair Cipher
- Vigenère Cipher

4. Hashing and File Integrity Verification: CO3, CO4

Implement a Python program to generate and compare hash values (MD5/SHA-256) for file integrity verification.

5. Network Packet Analysis using Wireshark: CO2, CO3

Capture and analyze network packets using Wireshark. Identify protocols, source and destination addresses, ports, and packet characteristics.

6. Network Discovery and Port Scanning using Nmap: CO2, CO3

Perform host discovery and port scanning using Nmap. Analyze open ports, services, and potential security implications.

7. User Authentication System: CO3, CO4

Develop a Python-based user authentication system implementing user registration, login validation, password verification, and role-based access control concepts.

8. Cyber Risk Management and Business Continuity Planning: CO5

Prepare a Risk Register, Risk Analysis Matrix, and a basic Business Continuity and Disaster Recovery Plan for a selected organization.

Text Books:

1. Charles J. Brooks, Christopher Grow, Philip Craig Jr., Donald Short, "Cyber Security Essentials", McGraw-Hill Education.
2. Michael E. Whitman, Herbert J. Mattord, "Principles of Information Security", Cengage Learning.

Reference Books:

1. William Stallings, Lawrie Brown, "Computer Security: Principles and Practice", Pearson Education.
2. Mark Stamp, "Information Security: Principles and Practice", Wiley India.

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

MDM-331- IT : Multimedia & Animation

Teaching Scheme	Credits	Examination Scheme
Theory : 01 Hours/Week	01	Term Work: 25 Marks Oral : 25 Marks

Prerequisite Courses: Basic Computer Graphics Concepts, Foundational Mathematics (Linear Algebra & Truncation), Fundamental Programming Logic (Python or C++), Computer Networks.

Companion Course: Multimedia & Animation Lab

Course Objectives:

- To understand the fundamentals of digital image formation, sensing mechanisms, and the mathematical principles of raster data manipulation such as layer masking, histograms, and pixel adjustments.
- To teach the geometric representation of visual data, mathematical properties of Bezier curves, resolution independence constraints, and path boolean operations.
- To familiarize students with the 12 basic principles of physical motion animation alongside digital audio sampling theorems, multi-track mixing, and sound cancellation mechanics.
- To introduce timeline compositing, video synchronization frameworks, and the geometric mathematics behind 3D spatial viewport coordination and mesh polygonal edge extrusions.
- To instill core competencies in UV unwrapping, PBR shader node map configurations, Inverse Kinematics (IK) skeletal constraint hierarchies, and virtual camera motion paths.

Course Outcomes:

- CO-1: Apply non-destructive editing techniques, layer configurations, and spatial adjustments to manipulate pixel grids and build seamless digital raster composites.
- CO-2: Design and generate fully scalable vector graphic layouts, custom curves, and typography portfolios utilizing area geometry and constraint-based tracing nodes.
- CO-3: Implement frame-by-frame and shape-tweening animation sequences while executing algorithmic noise filters, dynamic adjustments, and crossfades on multi-channel audio tracks.
- CO-4: Build optimized hard-surface 3D models with clean topologies while synchronizing multi-layered raw video footage elements according to targeted audio vectors.
- CO-5: Analyze and construct comprehensive end-to-end multimedia asset portfolios by synthesizing skeletal animations, material textures, studio lighting rigs, and cinematic rendering optimization profiles.

Course Contents

Unit I	2D Digital Imaging & Raster Graphics Engineering	(03 Hours)

Fundamentals of Digital Image Formation: Sensing mechanisms, digital sampling and spatial/amplitude quantization boundaries. Continuous vs. discrete color spaces (RGB, CMYK, HSV, and Alpha channels).
Mathematical Processing of Raster Matrices: Understanding intensity histograms, global mapping curves, gray-level transformations, and non-destructive layer masking principles.
Technical Asset Pipeline: Mechanics of localized cloning, healing brush algorithms, and frequency separation techniques used in production-level matte painting and digital composition.
Case Studies: Cinematic Virtual Production Environment

Unit II	Vector Graphics and Brand Asset Design	(03 Hours)

Geometric Representation of Visual Data: Resolution independence and vector scaling principles. Theoretical mechanics of Bezier curves, control anchor points, and mathematical vector path manipulation.
Constructive Area Geometry: Computational processing of path boolean operations (Union, Difference, Exclusion, Intersection, Division). Typography geometry: Text-on-path anchoring and font kerning math.
Graphic Optimization Frameworks: High-fidelity file formatting constraints (SVG, EPS, PDF layout models). Translating structural vector paths into crisp raster pixels (Rasterization algorithms).
Case Studies : Dynamic UI Vector System for Automated Smart Vehicles

Unit III	Traditional 2D Animation Foundations & Audio Engineering	(03 Hours)

Physics of Motion in 2D Space: Deconstructing the 12 Principles of Animation (Squash and Stretch, Anticipation, Staging, Timing). Mechanics of onion-skinning, keyframing, and automatic shape-tweening/morphing interpolation.
Digital Audio Representation & Physics: Sampling rate constraints (Nyquist-Shannon Theorem), bit depth, and multi-track processing matrices. Acoustic properties of digital signals.
Audio Signal Processing: Algorithmic fundamentals of ambient noise profiling/cancellation, parametric EQ curves, dynamic range compression transformations, and stereo panning.
Case Studies : Multi-Channel Broadcast Audio Restoration Engine

Unit IV	Non-Linear Video Editing & 3D Spatial Mesh Modeling	(03 Hours)

Video Composition and Timeline Engineering: Frame rates, non-linear video clip trimming, temporal sync parameters, and alpha channel overlays. Frame interpolation and transitions.
3D Viewport Coordination and Mesh Topology: Working with XYZ coordinate spaces. Mesh creation algorithms: Polygon vs. NURBS models, loop cuts, edge extrusions, and subdivision surface smoothing mathematics Hard-Surface Prototyping: Constructive 3D Boolean modifiers, beveling mechanics, and optimization rules for maintaining clean, quad-dominant clean topology.
Case Studies : 3D Product Ad Campaign with Advanced PBR and IK Rigging

Unit V	Advanced 3D Pipelines: Texturing, Rigging, & Camera Animation	(3 Hours)

Surface Property Map Ingestion: Principles of UV Unwrapping seams. PBR (Physically Based Rendering) texture map routing: Diffuse, Roughness, Metallic, and Normal/Bump map shaders. 3-Point Studio lighting physics. Skeletal Kinematics & Rigging: Bone hierarchies, armatures, vertex weight painting parameters, and Forward vs. Inverse Kinematics (IK) mathematical constraint solvers

Camera Motion Tracking & Render Analytics: Path constraints for virtual camera flight tracking. Comparative review of raster vs. path-tracing rendering engines. Synthesis of the multimedia production pipeline.

Case-Study: 3D Product Ad Campaign with Advanced PBR and IK Rigging

Learning Resources

Textbooks:

1. "Multimedia: Making It Work" by Tay Vaughan (McGraw Hill Education, 9th/10th Edition)
2. "Fundamentals of Multimedia" by Ze-Nian Li, Mark S. Drew, and Jiangchuan Liu (Springer / PHI Learning)

Reference Books:

1. "Multimedia Communication Systems: Techniques, Standards, and Networks" by K.R. Rao, Zoran S. Bojkovic, and Dragorad A. Milovanovic (Pearson Education)
2. "Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods (Pearson Education, 4th Edition)
3. "Blender 3D By Example" by Florent Renaud and Romain Caudron
4. "Principles of Multimedia" by Ranjan Parekh (McGraw Hill Education, 2nd Edition)

E-Books:

1. https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs178
2. https://nptel.ac.in/courses/117105081
3. https://nptel.ac.in/courses/117105081
4. https://nptel.ac.in/courses/127101013

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

MDM-331-IT : Multimedia & Animation Lab

Teaching Scheme	Credits	Examination Scheme
Practical : 2 Hours/Week	01	Term Work: 25 Marks Oral-25 Marks

Companion Course: Multimedia & Animation

List of Experiments

Group A: 2D Raster & Vector Graphics (Compulsory)

1.Raster Graphics Manipulation and Digital Matte Composition

Import multiple raw photographic slices to create a seamless digital matte background composite. Perform image adjustments using color histograms, levels/curves, layer masking, and the clone/healing stamp repair tools

2. Vector Graphics Design and Branding Assets

Design a fully scalable vector corporate branding portfolio (Logo, Icon, and Visiting Card layout) utilizing custom Bezier curve manipulation, path boolean operations (Union, Difference, Intersection), and text-on-path formatting constraints.

Group B: 2D Animation, Audio & Video Editing

3.2D Animation, Audio & Video Editing

Design a smooth 2D animation sequence depicting a classic "bouncing ball" or a basic character walking cycle. Establish custom motion path guides and apply automatic shape-morphing interpolation.

4. Digital Audio Editing, Mixing, and Sound Engineering

Import fragmented voiceover clips and background music tracks. Perform ambient noise reduction/cancellation, execute audio compression transformations, insert custom crossfades, and export a

master audio file mixed cleanly across stereo channels

5. Non-Linear Video Editing and Motion Graphics Compositing

Import various raw visual footage layers. Slice, re-arrange, and sync the clips according to an audio beat track, overlaying lower-thirds animated text assets using keyframe transformation nodes

Group C: 3D Modeling, Rigging & Animation (Any 3)

6. 3D Hard-Surface Mesh Prototyping

Build a detailed 3D low-poly or hard-surface model (e.g., a mechanical gear assembly, a drone frame, or a stylized building) utilizing loop cuts, edge extrusions, bevels, and subdivision surface smoothing operations.

7. 3D Texturing, Lighting Setup, and Shader Node Graphs

Take the 3D mesh model generated in Experiment 6 and unwrap its UV coordinates. Generate and apply Diffuse (Color), Roughness, and Normal/Bump maps via shader nodes. Establish a standard 3-Point Studio Lighting rig (Key Light, Fill Light, and Rim Light).

8. 3D Skeletal Rigging and Inverse Kinematics (IK)

Construct an internal bone armature for a character or mechanical mesh. Bind the skeleton using automated vertex weights, fix deformation glitches via weight paint brush adjustments, and configure Inverse Kinematics
(IK) constraints for joint articulation.

9. Camera Path Animation and Render Optimization

Design a path constraint for a virtual camera to create a fluid, cinematic fly-through sequence around a 3D scene. Animate the camera along the path using smooth Bezier curve handles and export the finalized clip to a
compressed MP4 video container.

Group D: Multimedia Capstone Production (Compulsory Mini-Project)

1. Unified Multimedia Mini-Project Portfolio Development

Working in micro groups with selected problem statement and execute the entire content creation life cycle:

- Draft structural script and story board layout
- Synthesize audio elements
- Generate visual graphics, 2D vector or 3D Animation

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

VSE372-IT : Full Stack Development with MERN

Teaching Scheme	Credits	Examination Scheme
Practical : 2 Hrs/wk	1	TW : 50 Marks

Prerequisite: Good understanding of Programming and Problem-Solving concepts

Course Objectives:

- Understand the architecture and components of modern full-stack web applications using the MERN stack.
- Develop server-side applications and RESTful APIs using Node.js and Express.js.
- Design and manage NoSQL databases using MongoDB and Mongoose.
- Build responsive and interactive Single Page Applications (SPAs) using React.
- Integrate frontend, backend, and database components to develop, secure, and deploy full-stack web applications.

Course Outcomes:

- **CO1:** Apply modern JavaScript (ES6+) features and Node.js concepts to develop server-side applications.
- **CO2:** Design and develop RESTful web services using Express.js, middleware, and API integration techniques.
- **CO3:** Create and manage MongoDB databases using Mongoose for data modeling, validation, and database operations.
- **CO4:** Develop responsive and interactive Single Page Applications using React components, hooks, state management, and routing.
- **CO5:** Build, test, secure, and deploy complete MERN-stack applications by integrating React, Express, Node.js, and MongoDB.

Guidelines for Instructor's Manual

Lab Assignments: Following is a list of suggested laboratory assignments for reference. Laboratory Instructors may design a suitable set of assignments for their respective courses at their level. Beyond curriculum assignments, the mini-project is also included as a part of laboratory work. The Inclusion of few optional assignments that are intricate and/or beyond the scope of curriculum will surely be the value addition for the students and it will satisfy the intellectuals within the group of the learners and will add to the perspective of the learners. For each laboratory assignment, it is essential for students to draw/write/generate flowchart, algorithm, test cases, mathematical model, Test data set and comparative/complexity analysis (as applicable).

Guidelines for Student's Lab Journal
Program codes with sample output of all performed assignments are to be submitted as softcopy. Use of DVD or similar media containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journals may be avoided. Submission of journal/ term work in the form of softcopy is desirable and appreciated
Guidelines for Lab /TW Assessment
Term work is continuous assessment that evaluates a student's progress throughout the semester. Term work assessment criteria specify the standards that must be met and the evidence that will be gathered to demonstrate the achievement of course outcomes. Categorical assessment criteria for the term work should establish unambiguous standards of achievement for each course outcome. They should describe what the learner is expected to perform in the laboratories or on the fields to show that the course outcomes have been achieved. It is recommended to conduct an internal monthly practical examination as part of continuous assessment.
Guidelines for Laboratory Conduction
The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. Beyond curriculum assignments and mini-project may be included as a part of laboratory work. The instructor may set multiple sets of assignments and distribute among batches of students. For each laboratory assignment, it is essential for students to draw/write/generate flowchart, algorithm, test cases, mathematical model, Test data set and comparative/complexity analysis (as applicable). Batch size for practical and tutorials may be as per guidelines of authority.
List of Assignments
1: JavaScript ES6+ Programming –CO1 Develop a web application demonstrating Arrow Functions, Template Literals, Destructuring, Classes, Modules, Promises, and Async/Await for processing student/course data.
2: Node.js Fundamentals– CO1
Create a Node.js application using built-in modules (fs, path, os, events) to perform file handling and event-driven operations.
3: Express.js Web Server and Routing–CO2
Develop an Express.js application implementing routing, middleware, request handling, and error handling for a Student Information System.

4: RESTful API Development**– CO2**
Design and implement REST APIs (GET, POST, PUT, DELETE) for managing student, product, or employee records. Test APIs using Postman.
5 : MongoDB and Mongoose Integration–CO3
Create a MongoDB database and implement CRUD operations using Mongoose schemas, validation, and query methods.
6. React Components and Hooks–CO4
Develop a React application using functional components, props, state management, and hooks (useState, useEffect) to create an interactive dashboard.
7. Single Page Application using React Router**– CO4**
Build a SPA with multiple pages (Home, About, Login, Dashboard) using React Router and navigation components.
8: Frontend–Backend Integration**–CO2,CO3,CO4,CO5**
Integrate React frontend with Express.js backend using Axios/Fetch API. Implement data submission and retrieval from MongoDB.
9. Authentication and Authorization using JWT**–CO2,CO3,CO4,CO5**
Develop user registration and login modules with password encryption, JWT-based authentication, and protected routes.
10: Mini Project: Full Stack MERN Application**–CO1,CO2,CO3,CO4,CO5**
Develop and deploy a complete MERN application (e.g., Library Management, Online Examination, Event Management, E-Commerce, College ERP) incorporating CRUD operations, authentication, and responsive UI.
Reference Books: 1. Ethan Brown, *Web Development with Node and Express*, O'Reilly Media. 2. Vasan Subramanian, *MERN Stack Web Development*, Packt Publishing. 3. Ted Hagos, *Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node*, Apress. 4. Alex Banks and Eve Porcello, *Learning React: Modern Patterns for Developing React Apps*, O'Reilly Media. 5. Shannon Bradshaw, Kristina Chodorow, and Michael Dirolf, *MongoDB: The Definitive Guide*, O'Reilly Media. 6. Mario Casciaro and Luciano Mammino, *Node.js Design Patterns*, Packt Publishing.

Savitribai Phule Pune University

Third Year of Information Technology (2024 Course)

ELC381-IT: Internship/On Job Training

Teaching Scheme	Credits	Examination Scheme
Practical : 8 Hrs/Week	4	OR - 50 Marks

Course Objectives:

- 1. To expose students to real-world industry practices.
- 2. To bridge the gap between academic learning and practical implementation.
- 3. Develop professional competency, ethics, communication, and teamwork skills.
- 4. To encourage self-learning and problem-solving abilities.
- 5. Encourage innovation, entrepreneurship, and research aptitude.

Course Outcomes:

- CO1: Apply theoretical knowledge to solve real-world engineering problems.
- CO2: Demonstrate technical competency in tools/technologies used in industry.
- CO3: Exhibit professional ethics and teamwork.
- CO4: Prepare technical reports and deliver effective presentations on industrial training experience.
- CO5 Analyze industrial processes and suggest feasible improvements or innovations.

Guidelines

1. Students should opt for a internship/OJT that would provide them to gain ample field knowledge in the relevant field of engineering such that theoretical knowledge gained in the class can be applied to solve the practical/ field problem.

2. Students must have to opt for technical internship after VI semester and before VII semester, preferably during summer break.

3. Undergoing a training program / Course at a particular organization for specified duration is NOT considered as summer internship

4. However student can attend such programs mentioned in above to learn new tools for short duration that would help for solving the problem undertaken in the internship

5. Students should take a challenging task, may be a small portion, and apply the knowledge gained to solve it.

6. Internship can also involve data collection from different sources, including generating experimental data, collection of data from field etc. The data may be analyzed later on.

7. Different central and state government organizations, CSIR labs, premier institutions like IITs and IIMs, DRDO, public sector undertaking organizations, top IT companies may be considered for internships.

8. Student need to submit Synopsis, Permission letter and offer letter to Internship coordinator before proceeding to internship.

9. Internship completion will be considered only after submission of valid documents at the end

of internship like Completion certificate, Report and presentation of work done, feedback from industry etc. 10. Student will appear for term work evaluation where he/she will present the work done before mentor(s) at the end of internship.

Suggested Internship Activities

Students are expected to perform the following activities during internship:

• Phase I – Orientation and Requirement Study
– Understanding organization structure
– Study of workflow and operational processes
– Requirement analysis and project allocation
– Understanding tools and technologies used
• Phase II – Technical Learning and Development
– Coding and implementation
– Database design and integration
– Software testing and debugging
– API integration and deployment
– Use of version control systems
– Documentation practices
• Phase III – Project Execution
– Module development
– Testing and validation
– Performance optimization
– Client interaction (if applicable)
– Team collaboration
• Phase IV – Documentation and Presentation
– Preparation of internship report
– Preparation of project demonstration
– Final presentation and viva voce

Internship Structure

The internship may be carried out in any one of the following domains:

- Software Development
- Artificial Intelligence and Machine Learning
- Data Science and Analytics
- Cloud Computing and DevOps
- Cyber Security
- Web and Mobile Application Development
- IoT and Embedded Systems
- Networking and System Administration

- Automation and Robotics Software
- Research and Development
- Entrepreneurship and Startup Projects
- Government/NGO Technical Projects

Learning Resource

Text Books:

1. AICTE Internship policy : AICTE Internship Policy: Guidelines & Procedures
2. AICTE Internship Portal : https://internship.aicte-india.org

TE – Information Technology 2024 Pattern

National Education Policy (NEP)-2020 Compliant Syllabus

Team Members Behind Curriculum Structure Design

Members-BoS IT

Dr. Aditya Abhyankar	Dr. G. V. Garje
Dr. Lalit Patil	Dr. Rajendra Sonar
Dr. Madhavi Pradhan	Dr. R. Venkateswaran
Dr. Poonam Lambhate	Dr. Maya Ingle
Dr. Abhijit Patankar	Dr. Abhijat Vichare
Dr. Madhuri Jawale	Dr. Surendra Mahajan
Dr. Deepak Vidhate	Dr. Abhijit Patankar

Course Coordinator -TE IT

Dr.Pratima Patil

Coordinator,TE IT Curriculum and Syllabus Development

Team Members for Course Content Design

Dr. Pratima Patil	Dr. Minakshi Atre	Dr. Amita Jajoo	Dr. Rahat Khan
Mrs. Sarawasti Nagtilak	Mrs.Gulnaaz Thakur	Mrs.Priyanka Kendre	Mrs. Yogita Deogharkar
Mrs Mangal Kale	Mr. Lakhan Sagale	Mrs. Sunita Chavan	Mr. Anandkumar Rao
Mrs Pooja Vaikar	Mrs. Sumaira Shaikh	Mrs.Shweta Joshi	Mr. Nikhil Vaidya

Chairman

Dr. Sudeep D. Thepade

Chairman– Board of Studies

Information Technology

Savitribai Phule Pune University, Pune

Dean

Prof. Dr. Raosaheb Latpate

Dean – Science and Technology

Savitribai Phule Pune University, Pune