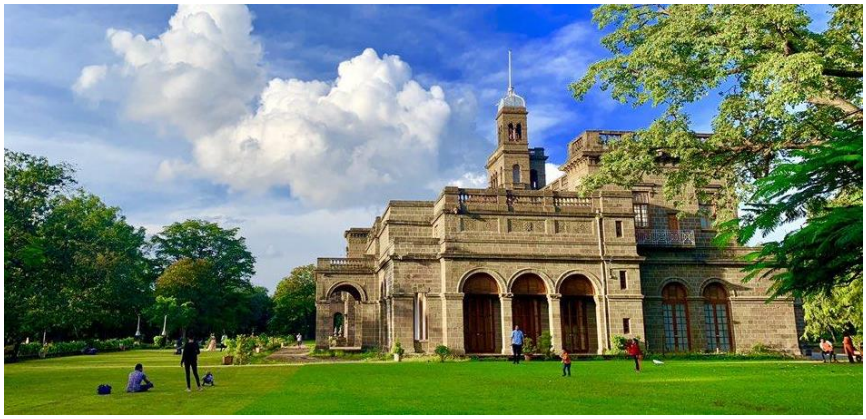
Savitribai Phule Pune University, Pune,

Maharashtra, India

सावित्रीबाई फुले पुणे विद्यापीठ

Faculty of Science and Technology



National Education Policy (NEP)-2020 Compliant Curriculum

Third Year Engineering (2024 Pattern) in

Artificial Intelligence and Machine Learning

(With effect from Academic Year 2026-27)

Contents

Savitribai Phule Pune University Third Year of Engineering (2024 Course)		
(With effect from Academic Year 2026-27)		
Table of Contents		
Sr. No.	**Title**	**Page**
	Preface	**I**
	Abbreviations	**II**
	Knowledge and Attitude Profile (WK)	**III**
	Program Outcomes	**IV**
	Program Specific Outcomes	**V**
	Course Structure	**VI**
	General Guidelines	**VII**
	Course Contents (Semester V)	
1	PCC-301-AIM – Operating System	**2**
2	PCC-302- AIM – Deep Learning	**4**
3	PCC-303- AIM – Theory of Computation	**9**
4	PCC-304- AIM – Operating System Lab 1	**12**
5	PCC-305- AIM – Deep Learning Lab 2	**15**
6	PEC-321- AIM – Elective I	**20**
7	PEC-322- AIM – Elective I Lab 3	**31**
8	MDM -33- AIM – AIoT Product Development	**41**
9	OLE-341- AIM – Open Elective	**48**
10	ELC-342- AIM – Technical Seminar	**57**
11	**Course Contents (Semester VI)**	
12	PCC-351- AIM – Computer Vision	**61**
13	PCC-352- AIM – Software Engineering with DevOps	**64**
14	PCC-353- AIM – Computer Vision Lab 1	**67**
15	PCC-354- AIM – Software Engineering with DevOps Lab 2	**70**
16	PEC-361- AIM – Elective 2	**73**
17	PEC-362- AIM – Elective 3	**90**
18	PEC-363- AIM – Elective 3 Lab 3	**103**
19	MDM-371- AIM – Multimedia Technology	**113**
20	VSE-372- AIM – Full Stack Development with MERN	**117**
21	ELC-381- AIM – Internship/OJT	**119**
22	**Team Members**	**121**

Preface

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 third year has been thoughtfully structured to strike a balance between core computing fundamentals—such as Operating System, Theory of Computation, Software Engineering & DevOps, Deep Learning, Computer Vision—and emerging domains like Data Science & Big Data Analytics, Cloud Computing, High Performance Computing, Natural language Processing, Cyber Security & Risk management. In addition, we have integrated value-added courses, skill development modules, interdisciplinary learning components as open electives, and Technical Seminar, Internship/OJT, AIoT based Product Development, Blockchain Technology, Full Stack Development with MERN, Power BI and Tableau 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 AI&ML 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 Artificial Intelligence and Machine Learning**		
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)

Program Specific Outcomes (PSO)	
A graduate of the Artificial Intelligence & Machine Learning Program will demonstrate-	
PSO1	An ability to apply the theoretical concepts and practical knowledge of Artificial Intelligence & Machine Learning 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. AI & ML 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

NEP 2020 Compliant Curriculum Structure

Third Year Engineering (2024 Pattern)

Artificial Intelligence and Machine Learning

SEMESTER V

Sr	Course Code	Course Type	Course Name	Teaching Scheme (Hrs/week)			Examination Scheme & Marks						Credit			
				Lecture	Tutorial	Practical	CCE	ESE	Termwor	Practical	Oral	Total	Lecture	Tutorial	Practical	Total
1	PCC-301 -AIM	PCC 1	Operating System	3	-	-	30	70	-	-	-	100	3	-	-	3
2	PCC-302 -AIM	PCC 2	Deep Learning	3	-	-	30	70	-	-	-	100	3	-	-	3
3	PCC-303 -AIM	PCC 3	Theory of Computation	3	-	-	30	70	-	-	-	100	3	-	-	3
4	PCC-304 -AIM	PCC Lab 1	Operating System Lab 1	-	-	2	-	-	25	25		50	-	-	1	1
5	PCC-305 -AIM	PCC Lab 2	Deep Learning Lab 2	-	-	4	-	-	25	50		75	-	-	2	2
6	PEC-321 -AIM	PEC I	Elective I	3	-	-	30	70	-	-	-	100	3	-	-	3
7	PEC-322 -AIM	PEC I Lab 3	Elective I Lab 3	-	-	2	-	-	25	-	-	25	-	-	1	1
8	MDM-33 -AIM	MDM	AIoT Product Development	-	1	4	-	-	25		50	75	-	1	2	3
9	OLE-341 -AIM	OE	Open Elective*	2	-	-	15	35	-	-	-	50	2	-	-	2
10	ELC-342 -AIM	ELC	Technical Seminar	-	-	2	-	-	-	-	25	25	-	-	1	1
Total				**14**	**1**	**14**	**135**	**315**	**100**	**125**	**25**	**700**	**14**	**1**	**7**	**22**

Programme Elective I (PEC)

PEC-321A-AIM	Cloud Computing
PEC-321B-AIM	Natural Language Processing
PEC-321C-AIM	Web Application Development
PEC-321D-AIM	Advanced Database Management System

Note: 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.

NEP 2020 Compliant Curriculum Structure

Second Year Engineering (2024 Pattern)

Artificial Intelligence and Machine Learning

SEMESTER VI

Sr	Course Code	Course Type	Course Name	Teaching Scheme (Hrs/week)			Examination Scheme & Marks						Credit			
				Lecture	Tutorial	Practical	CCE	ESE	Termwork	Practical	Oral	Total	Lecture	Tutorial	Practical	Total
1	PCC-351 -AIM	PCC 1	Computer Vision	3	-	-	30	70	-	-	-	100	3	0		3
2	PCC-352-AIM	PCC 2	Software Engineering with DevOps	2	-	-	30	70	-	-	-	100	2	0		2
3	PCC-353-AIM	PCC Lab 1	Computer Vision Lab 1	-	-	4	-	-	25	25	-	50	0	0	2	2
4	PCC-354-AIM	PCC Lab 2	Software Engineering with DevOps Lab 2	-	-	2	-	-	25	25	-	50	0	0	1	1
5	PEC-361-AIM	PEC II	Elective 2	3	-	-	30	70	-	-	-	100	3	0		3
6	PEC-362-AIM	PEC III	Elective 3	3	-	-	30	70	-	-	-	100	3	0		3
7	PEC-363-AIM	PEC III Lab 3	Elective 3 Lab 3	-	-	2	-	-	25	25	-	50	0	0	1	1
8	MDM-371-AIM	MDM	Multimedia Technology	-	1	2	-	-	25	-	25	50	0	1	1	2
9	VSE-372-AIM	VSE	Full Stack Development with MERN	-	-	2	-	-	50	-	-	50	0	0	1	1
10	ELC-381-AIM	ELC	Internship/OJT	-	-	8	-	-		-	50	50	0	0	4	4
Total				11	1	20	120	280	150	75	75	700	11	1	10	22

Programme Elective II (PEC)

PEC-361A-AIM	Quantum Computing
PEC-361B-AIM	Blockchain Technology
PEC-361C-AIM	High Performance Computing
PEC-361D-AIM	Data Science & Big Data Analytics

Programme Elective II (PEC)

PEC-362A-AIM	Power BI & Tableau
PEC-362B-AIM	Large Language Models
PEC-362C-AIM	Mobile Application Development
PEC-362D-AIM	Cyber Security & Risk Management

General Guidelines

Credit Specifications

I. Theory Courses: 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 30 marks with the specified parameters, the allocation of marks and the structure can be detailed as follows:

IX

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. N0.	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 & 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 70 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.
  - Analysing: 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 - V

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PCC-301 -AIM
Course Name: Operating System

Teaching Scheme	Credits	Examination Scheme
Theory: 3 Hrs./Wk.	3	CCE - 30 Marks ESE – 70 Marks

Prerequisite Courses, if any: Data Structures

Companion Course, if any:

Course Objectives:

1. Understand the fundamental concepts, architecture, and services provided by modern operating systems.
2. Analyze process management techniques, CPU scheduling algorithms, and synchronization mechanisms used in operating systems.
3. Apply memory management and virtual memory concepts to optimize system performance.
4. Evaluate file systems, storage management techniques, and security mechanisms used in contemporary operating systems.
5. Explore operating system concepts in AI/ML computing environments, cloud infrastructure, and high-performance computing systems.

Course Outcomes:

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

CO1: Explain the architecture, components, and functions of modern operating systems.
CO2: Analyze process scheduling, synchronization, deadlock handling, and resource allocation techniques.
CO3: Apply memory management and virtual memory algorithms to solve operating system problems.
CO4: Evaluate file systems, storage management strategies, and security mechanisms in operating systems.
CO5: Assess operating system support for AI/ML workloads, cloud computing platforms, and distributed environments.

Course Contents

Unit No: I	Introduction to Operating Systems and System Structures	9 Hours

Evolution and Functions of Operating Systems, Operating System Services, System Calls and APIs
Operating System Structures: Monolithic Systems, Layered Architecture, Microkernel Architecture, Virtual Machines
User Mode and Kernel Mode, Operating System Boot Process, Linux and Windows Architecture Overview

Exemplars /Case Studies	**Case Study 1:** Kernel components and System call interface **Case Study 2:** Process and memory management in Linux

Unit No: II	Process Management and CPU Scheduling	9 Hours
Processes and Process States, Process Control Block (PCB), Context Switching, Threads and Multithreading CPU Scheduling: FCFS, SJF, Priority Scheduling, Round Robin, Multilevel Queue Scheduling Process Synchronization, Critical Section Problem, Semaphores and Mutex Locks, Classical Synchronization Problems: Producer-Consumer, Readers-Writers, Dining Philosophers		
Exemplars/Case Studies	**Case Study 1:** CPU Scheduling in Linux Completely Fair Scheduler (CFS)	
Unit No: III	**Deadlocks and Memory Management**	**9 Hours**
Deadlocks: System Model, Necessary Conditions for Deadlock, Resource Allocation Graph, Deadlock Prevention, Deadlock Avoidance, Banker's Algorithm, Deadlock Detection and Recovery. Memory Management: Address Binding, Logical and Physical Address Space, Paging, Segmentation, Paging with Segmentation, Virtual Memory Concept		
Exemplars/Case Studies	**Case Study 1:** Google Data Center Memory Management, Memory allocation strategies **Case Study 2:** Resource optimization in large-scale systems	
Unit No: IV	**Virtual Memory, File Systems and Storage Management**	**9 Hours**
Virtual Memory: Demand Paging, Page Replacement Algorithms (FIFO, Optimal, LRU, Clock Algorithm), Thrashing, Working Set Model File Systems: File Concepts, Directory Structures, File Allocation Methods, File Access Methods, Free Space Management Storage Management: Disk Structure Disk Scheduling Algorithms (FCFS, SSTF, SCAN, CSCAN, LOOK)		
Exemplars/Case Studies	**Case Study 1:** Hadoop Distributed File System **Case Study 2:** Google File System (GFS)	
Unit No: V	**Security, Virtualization and Operating Systems for AI/ML**	**9 Hours**
Operating System Security: Authentication and Authorization, Access Control Mechanisms, Malware and Protection Techniques, Security Threats in Modern Systems Virtualization: Hypervisors, Containers and Docker, Virtual Machines, Resource Isolation OS Support for AI/ML Applications: GPU Resource Management, Parallel Processing, Distributed Computing Frameworks, Containerized AI Deployments, Edge and Cloud Computing Platforms		

Exemplars/Case Studies	**Case Study 1:** Docker for AI/ML Application Deployment **Case Study 2**: Kubernetes for Scalable AI/ML Infrastructure

Learning Resources

Textbooks:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Concepts, Wiley.
2. William Stallings, Operating Systems: Internals and Design Principles, Pearson.

Reference Books:

1. Andrew S. Tanenbaum, Modern Operating Systems, Pearson.
2. D. M. Dhamdhere, Operating Systems, McGraw Hill.
3. Remzi H. Arpaci-Dusseau, Andrea C. Arpaci-Dusseau, Operating Systems: Three Easy Pieces.

e-Books: https://pages.cs.wisc.edu/~remzi/OSTEP/?utm_source

Links to online SWAYAM/NPTEL Courses:

1. https://nptel.ac.in/courses/106105214
2. https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs47

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PCC-302-AIM
Course Name: Deep Learning

Teaching Scheme	**Credits 3**	**Examination Scheme**
Theory: 3 Hrs./Wk.		CCE - 30 Marks ESE - 70 Marks

Prerequisite Courses, if any:
Foundation of Machine Learning (PCC-252-AIM),
Probability & Statistics (PCC-253-AIM)
Data Structures & Algorithms (PCC-201-AIM)

Companion Course, if any: Deep Learning Lab 2

Course Objectives:

1. To understand the fundamental evolutionary shift from traditional machine learning to deep learning.
2. To learn the mathematical foundations, architectural building blocks of deep models.
3. To explore advanced deep learning architectures such as CNNs, RNNs, LSTMs, Autoencoders, GANs, and Transformers.
4. To analyze real-world deep learning applications using modern frameworks such as TensorFlow and PyTorch.
5. To develop deep learning models for computer vision, natural language processing, and sequence data.

Course Outcomes:

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

CO1: Explain the fundamentals of artificial neural networks and deep learning concepts.
CO2: Apply regularization and optimization techniques to train robust, deep neural architectures.
CO3: Develop convolutional neural network models for image classification and computer vision tasks.
CO4: Apply recurrent neural networks and sequence models for time-series and NLP applications.
CO5: Analyze advanced deep learning architectures including Autoencoders, GANs, and Transformers for real-world AI applications.

Course Contents

Unit No: I	Fundamentals of Neural Networks & Deep Learning	**8 Hours**

Evolution from Machine Learning to Deep Learning: Limitations of shallow networks and handcrafted features.
Biological vs. Artificial Neurons: The Perceptron model, mathematical representation, and decision boundaries.
Forward Propagation: Matrix formulation of data flow through dense layers. Backpropagation Algorithm: The chain rule of calculus for gradient computation.
Multilayer Perceptron's (MLPs): Architecture, hidden layers, and the universal approximation theorem.
Activation Functions: Linear vs. non-linear functions (Sigmoid, Tanh, ReLU, Leaky ReLU, Parametric ReLU, ELU, SoftMax)

Exemplars/ Case Studies	**Case Study 1: Implementation** of a Multi-Layer Perceptron from scratch using NumPy for binary classification. / Handwritten Digit Recognition using MNIST Dataset **Case Study 2:** Predict customer churn using a basic Feedforward Neural Network.

Unit No: II	Training Deep Neural Networks & Optimization	**8 Hours**

The Loss Function: Mean Squared Error (MSE), Binary Cross-Entropy, Categorical Cross-Entropy.
Concept of Empirical Risk Minimization and Optimization Techniques.
Gradient Descent Variants: Batch, Stochastic (SGD), and Mini-batch Gradient Descent.
The Vanishing and Exploding Gradient Problems: Causes, mathematical representation, and early mitigation strategies.
Weight Initialization Techniques: Xavier (Glorot) initialization and He (Kaiming) initialization.
Vanishing/Exploding Gradients: Graphical representation of gradient decay across deep layers.
Generalization Challenges in Deep Networks: Overfitting, underfitting, and model capacity, Early Stopping: Criteria, patience parameters, and validation error monitoring. Data Augmentation: Techniques for expanding dataset variance synthetically.
Regularization: Dropout, L1/L2 penalties Batch Normalization: Internal covariate shift, Mathematical formulation, scale gamma and shift beta parameters. Layer Normalization vs. Batch Normalization

Exemplars/Case Studies	**Case Study 1:** A healthcare startup is developing a deep neural network to classify chest X-ray images as **Normal** or **Pneumonia**. The dataset contains 50,000 labeled images. The team designs a 25-layer deep neural network but observes unstable training behavior. The training loss fluctuates significantly, validation accuracy remains low, and gradients either vanish or explode in deeper layers. The team must identify the causes of poor model performance and implement suitable optimization and regularization strategies to improve convergence and generalization. **Case Study 2:** An e-commerce company wants to automatically classify product images into 10 categories such as electronics, clothing, books, furniture, and groceries. The dataset contains 100,000 images but suffers from class imbalance and overfitting. The company develops a deep CNN with 40 layers. 98% training accuracy is achieved, however the model attains only 72% validation accuracy. Training logs reveal unstable gradients and poor generalization. Analyze the possible reasons for the same.

Unit No: III	Convolutional Neural Networks (CNNs)	**8 Hours**

Introduction to Spatial Data: Limitations of MLPs when handling high-dimensional image inputs. The Convolution Operation: Filters/kernels, feature maps, and spatial structure preservation. Hyperparameters of Convolutional Layers: Stride, Padding (Valid vs. Same), and Kernel Size. Pooling Operations: Max pooling, Average pooling, Global Average Pooling, and their mathematical intuition. CNN Architecture: LeNet Architecture, AlexNet Architecture, VGGNet Architecture, ResNet Architecture, GoogLeNet / Inception Network. Transfer Learning: Fine-tuning vs. feature extraction strategies using pre-trained weights. Feature Visualization: Understanding what CNN filters learn at shallow, mid, and deep layers. Object detection: R-CNN, Fast R-CNN, YOLO, SSD Semantic segmentation: U-Net, Mask R-CNN Challenges in CNN training		
Exemplars/Case Studies	**Case Study 1 :** Designing and training a custom CNN from scratch for the MNIST handwritten digit or CIFAR-10 classification task. **Case Study 2 :** on automated chest X-ray disease classification using a pre-trained ResNet-50 via Transfer Learning.	
Unit No: IV	Recurrent Neural Networks (RNNs)	**8 Hours**
Introduction to Sequential Data: Time-series, audio, and text dependencies. The Recurrent Neuron: Hidden states, memory retention, and temporal parameters sharing. RNN architecture and working, Vanishing Gradient in RNN, Limitations of vanilla RNNs. Gated Recurrent Units (GRU): Update gate, reset gate, and hidden state equations, Long Short-Term Memory (LSTM) networks : Architecture, Cell state vs. Hidden state. LSTM Gate Mechanics: Forget gate, Input gate, Output gate, and explicit mathematical equations. Bi-directional RNNs/LSTMs: Incorporating past and future context simultaneously. Deep/Stacked RNNs: Layering recurrent units for hierarchical temporal feature learning. Sequence-to-Sequence (Seq2Seq) Architectures: Encoder-Decoder framework concepts. The Bottleneck Problem in Seq2Seq models.		
Exemplars/Case Studies	**Case Study 1 :** Twitter Sentiment Analysis using RNN/LSTM **Case Study 2 :** Stock Price Prediction using LSTM	
Unit No: V	Deep Learning Frameworks and Advanced Applications.	**8 Hours**

Deep Learning Software Ecosystem: High-level abstractions vs. low-level tensor operations. PyTorch Core Ecosystem: Tensors, Autograd engine, nn. Module, and data loaders. TensorFlow & Keras Ecosystem: Functional API vs. Sequential API, tf.data pipelines, and eager execution. Introduction to Generative AI Paradigms: Autoencoders (AE) vs. Variational Autoencoders (VAE) structural basics. Generative Adversarial Networks (GANs): Generator-Discriminator minimax game formulation. Deep Learning for Natural Language Processing: Machine Translation and Named Entity Recognition (NER) use cases. Deep Learning for Speech & Audio: Automatic Speech Recognition (ASR) pipelines. Modern Trends: Foundation models, Large Language Models (LLMs), and Vision Transformers (ViT) conceptual horizons. Concepts of MLOps, Packaging and deploying deep learning models using REST APIs, containerization with Docker, deployment on cloud and edge devices, Continuous Integration and Continuous Deployment (CI/CD) pipelines. Ethical AI & Explainability: Bias in deep learning models and the black-box dilemma. Challenges in Deep Learning: Compute costs, massive data requirements, and adversarial vulnerabilities.	
Exemplars/Case Studies	**Case Study 1:** Building and deploying a real-time object detection pipeline using a pre-trained YOLO framework via PyTorch/TensorFlow. **Case Study 2:** Develop and deploy a containerized web service for a trained Deep Learning model using Docker and FastAPI.

Learning Resources

Textbooks:

1. Eugene Charniak, "Introduction to Deep Learning", MIT Press, 2023.
2. Francois Chollet, "Deep Learning with Python", Manning Publications, 2nd Edition, 2021.
3. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016.
4. Adrian Rosebrock, *Deep Learning for Computer Vision with Python* (3 Volumes), PyImageSearch, 2019.
5. Delip Rao and Brian McMahan, *Natural Language Processing with PyTorch: Build Intelligent Language Applications Using Deep Learning*, O'Reilly Media, 2019
6. Rajalingappaa Shanmugamani – Deep Learning for Computer Vision, Packt Publishing, 2018.
7. Charu C. Aggarwal, "Neural Networks and Deep Learning: A Textbook", Springer, 2018.
8. Eli Stevens, Luca Antiga, Thomas Viehmann, *Deep Learning with PyTorch*, Manning Publications, 2020.
9. Ashish Vaswani et al. – Attention is All You Need (Research paper)

Reference Books:

1. Michael Nielsen, Neural Networks and Deep Learning, Determination Press, 2015
2. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, O'Reilly Media, 3rd Edition, 2022.
3. Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, Dive into Deep Learning, Cambridge University Press, 2023.
4. Simon J.D. Prince, Understanding Deep Learning, MIT Press, 2023.
5. Christopher Bishop – Pattern Recognition and Machine Learning, Springer 2006.

e-Books:

1. Dive into Deep Learning (Free Digital Version): https://d2l.ai/
2. Michael Nielsen's Neural Networks and Deep Learning: http://neuralnetworksanddeeplearning.com/
3. Deep Learning Notes by Andrew Ng: https://www.deeplearning.ai/program/deep-learning-specialization/
4. https://www.deeplearningbook.org
5. https://www.manning.com/books/deep-learning-with-python (manning.com in Bing)
6. https://info.deeplearning.ai/machine-learning-yearning-book (info.deeplearning.ai in Bing)

Links to online SWAYAM/NPTEL Courses:

1. Introduction to Machine Learning (Prerequisite Revision) - Prof. Balaraman Ravindran, IIT Madras
2. Deep Learning – Prof. Mitesh Khapra, IIT Madras
3. Deep Learning - Prof Prabir Kumar Biswas, IIT Kharagpur
4. Natural Language Processing - Prof Pawan Goyal, IIT Kharagpur
5. Deep Learning for Computer Vision - Prof. Jayanta Mukhopadhyay, IIT Kharagpur
6. Deep Learning for Visual Computing – Prof. Debdoot Sheet, IIT Kharagpur

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PCC-303 -AIM
Course Name: Theory of Computation

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

Prerequisite Courses: Discrete Mathematics

Companion Course, if any: NA

Course Objectives:

1. Apply the concepts of Deterministic and Nondeterministic Finite Automata to recognize and model regular languages.
2. Analyze the equivalence between regular expressions, finite automata, and regular languages.
3. Apply the concepts of Context-Free Grammars to generate and represent Context-Free Languages.
4. Develop Push Down Automata models for solving computational problems
5. Apply the principles of Turing Machines to model and solve computational problems.

Course Outcomes:

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

CO1: Design DFA and NFA models for regular languages and validate string acceptance.

CO2: Analyze the equivalence and conversion between regular expressions, NFA, DFA, and regular languages.

CO3: Develop Context-Free Grammars for specified Context-Free Languages and generate valid strings using derivation techniques.

CO4: Construct the Pushdown Automata for Context-Free Languages and validate string acceptance.

CO5: Design Turing Machines for language acceptance and simple computational problems.

Course Contents

Unit No: I	Formal Languages and Finite Automata	09 Hours

Introduction to Finite Automata, Central Concepts of Automata Theory: Alphabets, Strings, Operations over Strings

Formal Languages: Definition, Finite representation of languages, Operations over languages: Union, Concatenation, Kleene star and Kleene plus, Decision Problems Vs Languages,

Deterministic Finite Automata (DFA)-Formal Definition, Simplified notation: State transition graph, transition table, Language of DFA, construction of DFAs for Languages and proving correctness.

Nondeterministic finite Automata (NFA), NFA with epsilon transition and language of NFA, Conversion of NFA with epsilon transitions to DFA, Introduction and construction of Mealy and Moore Machines Applications and Limitation of Finite Automata

Exemplars /Case Studies	**Case Study 1:** AI-Based Chatbot Using Finite State Machine **Case Study 2:** DFA-Based Email and Spam Detection System

Unit No: II	Regular and Non-Regular Languages	09 Hours
Regular expression (RE): Definition, Operators of regular expression and their precedence, Algebraic laws for Regular expressions, Kleene's Theorem: Equivalence Regular expressions and DFAs, Closure properties of Regular Languages (union, intersection, complementation, concatenation, Kleene closure), Decision properties of Regular Languages, Applications of Regular expressions. **Conversions:** Conversion from Regular Expression to Nondeterministic finite Automata and Deterministic Finite Automata, Conversion from Deterministic Finite Automata to Regular Expression. Myhill-Nerode theorem. **Non-Regular Languages:** Revisiting Pigeon-Hole principle, Pumping Lemma for regular Languages.		
Exemplars/Case Studies	1. Validation check for keywords, constants in C Language 2. Pattern matching examples	
Unit No: III	**Context Free Grammar**	**09 Hours**
Context Free Grammars: Definition, Examples, Derivation, Languages of CFG, Constructing CFG, Closure properties of CFLs (Union, Concatenation, Kleene closure, reversal). Derivation tree /Parse Tree, Ambiguity in CFGs, Removing ambiguity, Inherent ambiguity. **Simplification of CFG**: Eliminating of epsilon productions, unit productions, removing useless and non-reachable symbols, Normal forms for CFGs: Chomsky normal form, Greibach normal form. Decision Properties of CFLs (Emptiness, Finiteness and Membership). Applications of CFG.		
Exemplars/Case Studies	1. Grammar for Arithmetic Expressions 2. Syntax checking of English sentence	
Unit No: IV	**Push Down Automata**	**09 Hours**
Introduction, Formal definition of Push Down Automata(PDA), Acceptance by Final state, Acceptance by empty stack, Deterministic and Non-deterministic PDAs, PDA & Context Free Language, CFG to PDA construction, Pumping lemma for CFLs, non-Context Free Languages, Chomsky hierarchy. Applying PDA for Top-Down Parsing and Bottom-up Parsing.		
Exemplars/Case Studies	1. Parser for arithmetic expression of C Language 2. Parser for if-else statement of C Language	
Unit No: V	**Turing Machines**	**09 Hours**
Basic model, definition, and representation, Instantaneous Description, Language acceptance by TM. Robustness of Turing Machine model and equivalence with various variants: Two-way/One-way infinite tape TM, multi-tape TM, non-deterministic TM, Universal Turing Machines. Church-Turing Thesis and intuitive notion of Algorithm, Encoding for Turing machines and countability of set of all Turing machines. Undecidability of Halting problem. Example of undecidable problem: Post Correspondence Problem, Example of Turing unrecognizable language.		
Exemplars/Case Studies	1. Turing machine for password validation 2. Simulate any computation or program like addition, subtraction operations.	

Learning Resources
Textbooks: 1. Hopcroft J, Motwani R, Ullman, Addison-Wesley, “Introduction to Automata Theory, Languages and Computation”, Second Edition, ISBN 81-7808-347-7. 2. Michael Sipser, Course Technology, “Introduction to Theory of Computation”, Third Edition,ISBN-10: 053494728X.
Reference Books: 1. J. Martin, “Introduction to Languages and the Theory of Computation” Third edition, Tata McGraw-Hill, ISBN 0-07-049939-x, 2003. 2. Daniel I. A. Cohen, “Introduction to Computer Theory”, Wiley-Second Edition, ISBN-10 : 04711377
Links to online SWAYAM/NPTEL Courses: 1. https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs114

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PCC-304 -AIM
Course Name: Operating System Lab 1

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

Course Objectives:

1. Develop an understanding of the fundamental concepts, architecture, services, and structures of modern operating systems.
2. Examine process management techniques, CPU scheduling policies, threads, and synchronization mechanisms used in contemporary operating systems.
3. Apply memory management, virtual memory, and deadlock handling techniques to improve system performance and resource utilization.
4. Analyze file systems, storage management methods, and operating system security mechanisms for efficient and secure computing.
5. Explore virtualization, containerization, cloud-based operating systems, and their role in supporting AI/ML applications and distributed computing environments.

Course Outcomes:

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

CO1: Explain the architecture, services, system calls, and structural design of modern operating systems.

CO2: Analyze process scheduling, multithreading, synchronization, and deadlock management techniques for efficient resource allocation.

CO3: Apply memory management and virtual memory algorithms to solve operating system-related problems and evaluate their performance.

CO4: Evaluate file organization, storage management, disk scheduling, and security mechanisms in operating systems.

CO5: Assess the use of virtualization, containerization, and operating system support for cloud computing and AI/ML workloads.

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. Experiments to be conducted VS Code + Ubuntu (WSL or Virtual Machine) is the ideal setuip for conduction of assignments.

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 list of practical Assignment.
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
2. parameters such as timely conduction of practical assignment, methodology adopted for implementation of practical assignment, timely submission of assignment in the form of
3. handwritten write-up along with results of implemented assignment, attendance etc.
4. 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.
5. 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 VS Code + Ubuntu (WSL or Virtual Machine)

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

Virtual Laboratory

- https://os-scheduling-algorithms.vercel.app/
- https://hpc-tutorials.llnl.gov/posix/
- https://www.docker.com/play-with-docker/ (Assignment 10)

1. Linux Environment and File Management -- CO1

Perform file and directory operations in Linux using command-line utilities. Demonstrate file permissions, ownership management, process monitoring, and basic system administration commands.

2. Shell Scripting and System Automation: -- CO1

Develop shell scripts using decision-making statements, loops, functions, and file handling commands to automate common operating system tasks.

3. Process Creation, Scheduling, and Performance Analysis -- CO2, CO4, CO5

Develop programs to create and manage processes using system calls. Implement CPU scheduling algorithms (FCFS, SJF, Round Robin) and analyze their impact on process execution, memory utilization, and overall system performance.

4. Multithreading and Resource Utilization –CO2, CO3, CO4

Create a multithreaded application using POSIX threads to execute concurrent tasks. Measure CPU utilization, memory consumption, and execution time under different thread

configurations.
5. Inter-Process Communication and Shared Resource Management -- CO2, CO3, CO5
Implement Inter-Process Communication using pipes, shared memory, and message queues. Analyze communication efficiency, resource sharing, and memory utilization.
6. Synchronization Mechanisms for Concurrent Systems -- CO2, CO3, CO5
Develop programs using semaphores and mutex locks to solve synchronization problems such as Producer–Consumer and Readers–Writers. Evaluate the effect of synchronization on resource allocation and system performance.
7. Deadlock Detection, Avoidance, and Recovery -- CO2, CO3, CO4
Implement Banker's Algorithm and deadlock detection techniques. Analyze resource allocation patterns and evaluate system performance under different deadlock scenarios.
8. Memory Management and Page Replacement Strategies -- CO4, CO5
Simulate paging, segmentation, and virtual memory management. Implement FIFO, LRU, and Optimal page replacement algorithms and compare their effectiveness using performance metrics.
9. File System and Disk Scheduling Analysis -- CO4, CO5
Implement file allocation and disk scheduling algorithms (FCFS, SSTF, SCAN, CSCAN). Analyze storage utilization, access time, and system throughput.
10. Containerization and Resource Monitoring for AI/ML Workloads -CO1, CO2, CO5
Deploy a simple AI/ML application inside a Docker container. Monitor CPU, memory, storage, and process utilization using Linux tools and evaluate the benefits of container-based deployment.

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PCC-305-AIM
Course Name: Deep Learning Lab-II

Teaching Scheme	Credits	Examination Scheme
Practical : 4 Hrs/wk	2	TW: 25 PR : 50

Course Objectives:

1. To understand the fundamental concepts, architecture, and mathematical foundations of deep learning and neural networks.
2. To develop the ability to design, train, and optimize deep neural network models for solving real-world problems.
3. To apply advanced deep learning techniques such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transfer Learning for data-driven applications.
4. To develop hands-on expertise in implementing deep learning models using modern frameworks like TensorFlow and PyTorch, addressing common challenges like overfitting, vanishing/exploding gradients, optimization and hyperparameter tuning.
5. To enable students to apply deep learning techniques to solve real-world engineering problems across domains such as vision, NLP, and robotics.

Course Outcomes:

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

CO1: Explain the principles of artificial neural networks, deep learning architectures, activation functions, loss functions, and optimization techniques.

CO2: Design and implement deep learning models using modern frameworks to solve classification, regression, and prediction problems.

CO3: Apply CNNs, RNNs, LSTMs, and transfer learning techniques for image, text, speech, and sequential data analysis.

CO4: Evaluate and improve deep learning models using appropriate performance metrics, regularization methods, and hyperparameter tuning strategies.

CO5: Design and build end-to-end AI applications integrating deep learning solutions for domains like image recognition, speech processing, and text analysis.

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. Experiments to be conducted Part A in C++ and Part B in Python.

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. The 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 the faculty concerned and head of the department.
5. All the assignments 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
2. parameters such as timely conduction of practical assignment, methodology adopted for implementation of practical assignment, timely submission of assignment in the form of
3. handwritten write-up along with results of implemented assignment, attendance etc.
4. 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.
5. 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 them 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

PART A

1. Implementation of Backpropagation in Multilayer Neural Networks for Image Classification. MNIST Dataset. -- CO1, CO2
   Load and preprocess MNIST Handwritten digits.
   Build a deep MLP with multiple hidden layers.
   Implement backpropagation using TensorFlow/PyTorch automatic differentiation.
   Train models with: One hidden layer, Two hidden layers, Three hidden layers
   Use Softmax activation in the output layer.
   Compare performance with: Logistic Regression, Decision Tree Classifier
   Analyze: Accuracy, Loss curves, Training time
   Conclude how deep learning overcomes limitations of handcrafted features and traditional

models.
2. Implementation of Multilayer Perceptron and Comparative Study of Activation Functions. IRIS Dataset. -- CO1, CO2 Load and normalize the dataset. Build an MLP using TensorFlow/Keras or PyTorch. Implement forward propagation manually for one hidden layer. Train separate models using: Sigmoid, Tanh, ReLU, Leaky ReLU, ELU Compare: Training loss, Validation accuracy, Convergence speed, Plot activation function curves. Interpret the effect of non-linearity on learning. Conclude with advantages and disadvantages of activation functions with reference to outcome.
3. Implementation and study of Gradient Problems, Generalization, and Regularization. Fashion-MNIST Dataset. -- CO1, CO2 1) Prepare the dataset for training and validation: Download Fashion-MNIST dataset Normalize pixel values to [0,1] Split into training, validation, and test sets. 2) Train Deep MLP with Poor Initialization: Show vanishing/exploding gradients with naive initialization. Use random small weights Train for 20 epochs Plot gradient norms across layers 3) Apply Xavier and He Initialization: Improve gradient flow with proper initialization. Reinitialize weights using Xavier and He Train again and compare gradient plots Observe improved stability 4) Regularization: Prevent overfitting with dropout and penalties. Add dropout layers (p=0.5) Apply L1/L2 regularization Monitor validation accuracy 5) Use Early Stopping and Data Augmentation: Enhance generalization and prevent overfitting. Implement early stopping with patience=5 Apply random rotations and flips Compare validation error trends 6) Compare Batch vs Layer Normalization Analysis Evaluate normalization techniques for stability. Train with Batch Normalization Train with Layer Normalization Compare convergence speed and final accuracy
PART B
4. Comparative Study of CNN Architectures for Image Classification. CIFA 5. R-10 Dataset. Load and preprocess the CIFAR-10 dataset. --CO1, CO2, CO3 Design a basic CNN and evaluate classification accuracy. Experiment with: Different kernel sizes (3×3, 5×5), Strides (1, 2), Padding (Valid and Same), Max Pooling and Average Pooling Implement and compare: LeNet, AlexNet, VGG16, ResNet50 Apply Transfer Learning: Feature Extraction using pre-trained ResNet50, Fine-tuning selected layers Visualize: Convolutional filters, Feature maps from shallow and deep layers

	Analyze challenges such as overfitting and vanishing gradients.
6.	Implement Real-Time Object Detection using R-CNN, Fast R-CNN, YOLO and SSD Mobile-Net. Pascal VOC dataset. -- CO1, CO2, CO3 Download and preprocess Pascal VOC dataset. Implement object detection using: R-CNN, Fast R-CNN, YOLOv5/YOLOv8, SSD MobileNet Detect multiple objects in images and videos. Compare: Detection Accuracy (mAP), Inference Speed (FPS), Computational Cost Analyze strengths and limitations of: R-CNN, Fast R-CNN, YOLO, SSD
7.	Implement Semantic Segmentation using U-Net. Oxford-IIIT Pet dataset. . -- CO1, CO2, CO4 Download Oxford-IIIT Pet dataset. Preprocess images and segmentation masks. Implement U-Net architecture. Perform pixel-wise classification. Visualize segmented outputs. Compare predicted masks with ground truth masks. Evaluate using: IoU (Intersection over Union), Dice Coefficient, Pixel Accuracy
8.	Comparative Analysis of RNN, GRU Time-Series Forecasting stock market trends. -- CO1, CO2, CO5 Dataset : Yahoo Finance Historical Stock Prices: Download historical stock data (Open, High, Low, Close, Volume). Visualize trends and seasonality. Normalize features using Min-Max Scaling. Convert data into sliding windows of 30 previous days to predict the next day's closing price. Develop a simple RNN model with: One recurrent layer, Hidden units = 50, Dense output layer Observe: Training loss, Validation loss, Prediction accuracy Note the limitations observed during training. Implement a GRU network using: Update gate, Reset gate Analysis: Training convergence speed, Computational efficiency, Prediction performance
9.	English-to-French Neural Machine Translation using Sequence-to-Sequence Learning English French Sentence Pairs Dataset - CO1, CO2, CO5 Download the English French sentence dataset. Perform: Text cleaning, Lowercasing, Removal of special characters, Tokenization, Sequence padding Create: Input vocabulary, Output vocabulary Construct a Seq2Seq architecture consisting of: Encoder, Embedding layer, LSTM encoder, Final hidden and cell states Decoder, Embedding layer, LSTM decoder, Dense Softmax layer Train the model for sentence translation. Encoder–Decoder Framework: Input Sentence, Encoder LSTM, Context Vector, Decoder LSTM, Translated Sentence Experiment with: Short sentences (<10 words), Medium sentences (10–20 words), Long sentences (>20 words) Analyze: BLEU score degradation with increasing sentence length. Information compression limitations of the context vector. Replace the encoder with a Bi-directional LSTM. Evaluate: Translation quality improvements, Ability to capture both past and future context. Implement: Two-layer stacked LSTM encoder, Two-layer stacked LSTM decoder Compare against baseline Seq2Seq.

Evaluation Metrics: BLEU Score, Perplexity, Training Loss, Validation Loss
10. Design a deep learning model capable of classifying handwritten digits (0–9) and expose the model prediction through a REST API. . - CO1, CO2, CO4 MNIST Handwritten Digits Dataset Load the MNIST dataset. Normalize image pixel values. Create a neural network using: Keras Sequential API. Train the model for 5 epochs. Evaluate model accuracy. Save the trained model. Create a REST API using FastAPI or Flask.
11 Build a simple Named Entity Recognition pipeline to identify entities such as PERSON, ORGANIZATION, and LOCATION from news articles. -CO1, CO2, CO4, CO5 https://www.kaggle.com/datasets/abhinavwalia95/entity-annotated-corpus Load the Named Entity Recognition dataset. Perform basic preprocessing. Use a pre-trained spaCy model: en_core_web_sm Extract: PERSON, ORG, LOCATION Test the model on 5 custom news sentences. Large Language Models (LLMs), Vision Transformers (ViT) Automatic Speech Recognition (ASR) Pipeline Create a CI/CD workflow diagram showing: Data Collection, Model Development, Testing, Docker Container Creation, Cloud Deployment, Continuous Monitoring

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PEC-321A-AIM
Course Name: Cloud Computing (Elective I)

Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 3 H/W	3	CCE - 40 Marks End Sem - 60 Marks

Prerequisite Courses, if any: **Database Management System, Python Programming**

Companion Course, if any: Computer Networks, Machine Learning

Course Objectives:

1. Understand the fundamental concepts, architectures, service models, deployment models, and virtualization technologies used in cloud computing environments.
2. Explore cloud infrastructure, computing, storage, networking, and resource management services provided by major cloud platforms for scalable application deployment.
3. Develop knowledge of cloud-native technologies, containerization, orchestration, micro services, and serverless computing for modern software and AI/ML application development.
4. Utilize cloud platforms and services for big data processing, machine learning model development, deployment, and MLOps practices to support AI-driven applications.
5. Analyze cloud security, privacy, governance, and emerging cloud technologies to design secure, reliable and efficient cloud-based AI/ML solutions.

Course Outcomes:

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

CO1: Explain cloud computing architecture, service models and virtualization technologies.

CO2: Analyze cloud infrastructure services and deployment models for AI/ML applications.

CO3: Deploy cloud-native applications using containers and orchestration platforms.

CO4: Utilize cloud platforms for machine learning, big data analytics and model deployment.

CO5: Apply security, MLOps and emerging cloud technologies for scalable AI solutions.

Course Contents

Unit No: I	**Introduction to Cloud Computing and Virtualization**	**8 Hours**

Evolution of Computing: Distributed, Grid, Utility and Cloud Computing, Cloud Computing Concepts and Characteristics, Service **Models**: IaaS, PaaS, SaaS, Deployment **Models**: Public, Private, Hybrid, Community Cloud, Virtualization Concepts. **Hypervisors**: Type-I and Type-II, Virtual Machine Lifecycle Virtual Machines and Containers, Cloud Architecture and Data Centers, Benefits and Challenges of Cloud Computing.**AI/ML Infrastructure Requirements**: GPU Computing, TPU Computing, High Performance Computing (HPC).

Exemplars/Case Studies	**Case Study 1:** Netflix Cloud Migration.

Unit No: II	Cloud Infrastructure and Cloud Services	8 Hours
Compute Services: Virtual Machines, Containers, Serverless Computing, Cloud Instances. **Storage Services**: Object Storage, Block Storage, File Storage, Data Backup and Recovery. **Cloud Networking**: Virtual Networks, Subnets, IP Addressing, DNS Services. **Load Balancing**: Concepts, Types of Load Balancers, Traffic Distribution. **Auto Scaling**: Horizontal Scaling, Vertical Scaling, Scaling Policies. **Resource Provisioning**: Dynamic Resource Allocation, Resource Scheduling, Resource Monitoring. **Cloud Service Providers Overview**: AWS, Microsoft Azure, Google Cloud Platform,		
Exemplars/Case Studies	Case study on E-Commerce Scalability During Festivals. For Example: Amazon and Flipkart sale events.	
Unit No: III	**Cloud-Native Technologies and Containers**	**7 Hours**
Containerization Concepts: Containers vs Virtual Machines, Benefits of Containers. **Docker fundamentals**: Docker Architecture, Docker Images and Containers, Docker Hub, Docker Operations. **Kubernetes Fundamentals & components**: Pod, Deployment, Services, Replica sets. Microservices Architecture, Serverless Computing, Cloud Native Applications, CI/CD Basics, DevOps vs MLOps, GitHub Actions Overview.		
Exemplars/Case Studies	Spotify's Containerized Infrastructure.	
Unit No: IV	**Cloud for Big Data and Machine Learning**	**8 Hours**
Big Data Concepts: Characteristics & Challenges. **Distributed Data Processing**: Hadoop Ecosystem, HDFS, MapReduce, YARN. **Apache Spark Fundamentals**: Spark Architecture, RDD, Data Frames, Spark MLlib Cloud-based Data Analytics, Machine Learning in Cloud , AI Services in Cloud, ML Lifecycle in Cloud, Introduction to MLOps, Model Training and Deployment.		
Exemplars/Case Studies	Case study on Smart Agriculture Analytics.	
Unit No: V	**Cloud Security, MLOps and Emerging Trends**	**7 Hours**
Cloud Security Fundamentals: Security Challenges, Shared Responsibility Model, Security Best Practices. **Identity and Access Management (IAM):** Authentication and Authorization, Roles and Policies. Data Security and Privacy, Security Challenges in AI Systems, MLOps Concepts, Model Monitoring, Continuous Deployment of AI Models, Edge Computing Architecture, Fog Computing Architecture. **Multi-Cloud and Hybrid Cloud**: Benefits, Challenges, Industrial Use Cases **Emerging Trends:** Generative AI on Cloud, AI-as-a-Service, Green Cloud Computing, Cloud-Native AI Applications.		
Exemplars/Case Studies	MLOps in Autonomous Vehicle Development.	
Learning Resources		
Textbooks: 3. Rajkumar Buyya, James Broberg, Andrzej Goscinski, Cloud Computing: Principles and Paradigms. WILEY Publisher. 4. Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, Mastering Cloud Computing. McGraw Hill Publisher.		

Reference Books:

4. Thomas Erl,"Cloud Computing Concepts, Technology and Architecture",ServiceTech Press
5. Kelsey Hightower, Brendan Burns & Joe Beda ,"Kubernetes Up and Running".O'REILLY Publisher.
6. Mark Treveil & the Dataiku Team, "Machine Learning Engineering with MLOps" O'REILLY Publisher.
7. Martin Kleppmann, "Designing Data Intensive Applications". O'REILLY Publisher.

e-Books:-

1. **https://www.dbooks.org/exploring-cloud-computing-161729487x/?utm_source**
2. **https://www.intechopen.com/books/5416?utm_source**
3. **https://docs.docker.com/?utm_source**
4. **https://kubernetes.io/docs/home/?utm_source**

Links to online SWAYAM/NPTEL Courses:

3. https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs55
4. https://onlinecourses.nptel.ac.in/e-learning/preview/noc24_cs131

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PEC-321B-AIM
Course Name: Natural Language Processing

Teaching Scheme	Credits	Examination Scheme
Theory : 3 H/W	3	CCE - 40 Marks ESE - 60 Marks

Prerequisite Courses, if any: Data Structures and Algorithms, Probability and Statistics, Foundation of Machine Learning , Foundation of Artificial Intelligence

Companion Course, if any: Natural Language Processing Lab

Course Objectives:

1. To learn fundamental concepts and techniques of natural language processing (NLP).
2. To acquire knowledge of various morphological, syntactic, and semantic analysis.
3. To learn the various language modeling techniques for NLP.
4. To understand appropriate tools and techniques for processing natural languages.
5. To comprehend the real-world applications in NLP domain.

Course Outcomes:

On completion of the course, students will be able to

CO1: Describe the fundamental concepts of NLP, challenges and issues in NLP.

CO2: Analyze word level and syntactical parsing techniques.

CO3: Analyze semantic relationships and sentiment features for natural languages.

CO4: Integrate the NLP techniques for the information retrieval task

CO5: Demonstrate the use of NLP tools and techniques for real time applications of natural languages.

Course Contents

Unit No: I	Introduction	8 Hours

Introduction: History, NLP system, level of NLP, Challenges and Issues in NLP, Morphological Analysis: Introduction to Morphology, Types of Morphemes, Language Modeling: Grammar-based LM, Statistical LM, Regular Expressions, Finite State Automata, Morphological parsing with Finite State Transducers (FST) Tokenization, Detecting and Correcting Spelling Errors, Minimum Edit Distance.

Case Study	Smart Spell Checker for Student Assignment Submission

Unit No: II	Word Level and Syntactic Analysis	9 Hours

Word Level Analysis: Unsmoothed N-grams, Evaluating N-grams, Smoothing, Interpolation and Backoff – Word Classes, Part-of Speech Tagging, Rule-based, Stochastic and Transformation based tagging, Issues in PoS tagging, Hidden Markov and Maximum Entropy models.

Syntactic Analysis: Syntactic Representations of Natural Language, Parsing Algorithms, Probabilistic context-free grammar, and Statistical parsing.

Case Study	Part-of-Speech Tagging Using Hidden Markov Models for English Text Processing	
Unit No: III	**Semantic Analysis**	**9 Hours**
Semantic Analysis: Lexical Semantic, Relations among lexemes & their senses Homonymy, Polysemy, Synonymy, Hyponymy, WordNet, Word Sense Disambiguation (WSD), problem of sense tagging, Dictionary based approach, Latent Semantic Analysis, semantic role labeling, selection restrictions, decomposition of predicates Lexicon for sentiment, affect and connotation. Word Embeddings/ Vector Semantics: Bag-of-words, TFIDF, word2vec, doc2vec, Contextualized representations (BERT).		
Case Study	Resolving Multiple Meanings of Words Using NLP Techniques	
Unit No: IV	**Information Retrieval using NLP**	**9 Hours**
Information Retrieval: Introduction, Vector Space Model Named Entity Recognition: NER System Building Process, Evaluating NER System Entity Extraction, Relation Extraction, Reference Resolution, Coreference resolution, Cross Lingual Information Retrieval.		
Case Study	NLP-Based Information Retrieval System for E-Learning Platforms	
Unit No: V	**NLP Tools and Techniques**	**10 Hours**
Prominent NLP Libraries: Natural Language Tool Kit (NLTK), spaCy, TextBlob, Gensim etc. Linguistic Resources: Lexical Knowledge Networks, WordNets, Indian Language WordNet (IndoWordnet), VerbNets, PropBank, Treebanks. Machine Translation: Rule based techniques, Statistical Machine Translation (SMT), Cross Lingual Translation.		
Case Study	Study working of IBM Watson Natural Language Processing	
Learning Resources		
Textbooks: 1. Jurafsky, David, and James H. Martin, "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition", PEARSON Publication. 2. Manning, Christopher D., and nrich Schütze , "Foundations of Statistical Natural Language Processing", Cambridge, MA: MIT Press		
Reference Books: 1. Steven Bird, Ewan Klein, Edward Loper, "Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit", O'Reilly Publication		

2. Dipanjan Sarkar , “Text Analytics with Python: A Practical Real-World Approach to Gaining Actionable Insights from your Data”, Apress Publication ISBN: 9781484223871
3. Alexander Clark, Chris Fox, and Shalom Lappin, “The Handbook of Computational Linguistics and Natural Language Processing”, Wiley Blackwell Publications
4. Jacob Eisenstein, “Natural Language Processing”, MIT Press
5. Jacob Eisenstein, “An Introduction to Information Retrieval”, Cambridge University Press

e-Books:

1. https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf
2. https://www3.cs.stonybrook.edu/~cse521/L16NLP.pdf

Links to online SWAYAM/NPTEL Courses:

1. https://nptel.ac.in/courses/106101007
2. https://nptel.ac.in/courses/106106211

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PEC-321C-AIM
Course Name: Web Application Development

Teaching Scheme	Credits	Examination Scheme
Theory: 3 Hrs/Wk	3	CCE - 30 Marks ESE - 70 Marks

Prerequisite Courses, if any:
1. Programming languages C++, Java.
2. Object Oriented Concepts
3. Database Management System Concepts

Companion Course, if any:
1. Web Application Development Lab

Course Objectives:
1. To understand modern web application architecture and technologies.
2. To develop responsive and interactive web applications.
3. To design and implement RESTful APIs for AI applications.
4. To integrate databases and machine learning models with web applications.
5. To deploy AI-enabled web applications for real-world use cases.

Course Outcomes:
On completion of the course, students will be able to–
CO1: Develop responsive web interfaces using HTML, CSS, Bootstrap and JavaScript.
CO2: Design interactive client-side applications using modern JavaScript frameworks.
CO3: Develop backend services and REST APIs for web applications.
CO4: Integrate databases and machine learning models into web applications.
CO5: Develop and deploy AI-enabled web applications for real-world problems.

Course Contents

Unit No: I	Fundamentals of Web Technologies	9 Hours

HTML5: Structure of HTML Document, HTML Elements and Attributes, Forms and Validation, Lists, Tables, Hyperlinks, Multimedia Elements (Audio, Video, Canvas), Semantic HTML, Web Accessibility Basics
CSS3: Selectors, Box Model, Flexbox, Grid Layout, Responsive Design
Bootstrap: Grid System, Typography and Utilities, Components, Responsive Web Design
Web Architecture: Evolution of Web Technologies**,** Client-Server Architecture, Static vs Dynamic Websites, HTTP/HTTPS Protocol, REST Architecture

Exemplars /Case Studies	**Case Study 1:** College Information Portal **Case Study 2:** AI-based Student Performance Prediction Portal UI Design

Unit No: II	Client-Side Programming	9 Hours

JavaScript: Variables and Data Types, Functions and Objects, Arrays and Array Methods, Events and Event Handling, DOM Manipulation
Advanced JavaScript: ES6 Features, Template Literals, De structuring, Arrow Functions, Promises, Async/Await, Error Handling
JSON and AJAX: JSON Parsing, AJAX Concepts, Fetch API, API Consumption

ReactJS: Introduction**,** Components, Props, State, Forms, Hooks (Introduction)		
Exemplars/Case Studies	Interactive Student Performance Dashboard Interactive Data Analytics Dashboard	
Unit No: III	**Back-End Development and Web Services**	**9 Hours**
Flask Framework: Environment Setup and Virtual Environment, Flask Application Structure, Routing, Templates, Forms, Static Files Management, URL Parameters and Dynamic Routing REST APIs: REST Architecture, HTTP Methods, API Design Principles, JSON Request and Response HandlingSession Management: Cookies, Sessions, User Authentication Basics API Testing: Postman, Error Handling		
Exemplars/Case Studies	Student Recommendation System API	
Unit No: IV	**Database and AI Model Integration**	**9 Hours**
Database Connectivity: SQLite/MySQL**,** Database Connectivity using Flask**,** CRUD Operations, **MongoDB Basics:** Collections, Documents, CRUD Operations in MongoDB, Querying and Filtering Data **AI Model Integration:** Model Serialization Pickle, Joblib, Loading Trained Models, Model Input Preprocessing, Feature Engineering for Prediction Services, Handling Prediction Outputs **Prediction Services:** Web Forms to Prediction APIs, Response Handling		
Exemplars/Case Studies	House Price Prediction Web Application	
Unit No: V	**AI-Enabled Web Applications and Deployment**	**9 Hours**
Deployment Fundamentals: Hosting Web Applications, Render/Netlify/Vercel **AI Application Deployment:** Deploying ML Models, Image Upload Processing **Generative AI Integration:** LLM APIs, Chatbot Development Concepts **Security Fundamentals:** Authentication Basics, Data Privacy in AI Applications **Emerging Trends:** AI Agents, Multimodal Applications		
Exemplars/Case Studies	AI Chatbot for Academic Assistance	

Learning Resources

Textbooks:

1. Web Technologies, Uttam K. Roy, Oxford University Press, Latest Edition.
2. Internet and World Wide Web: How to Program, Paul Deitel and Harvey Deitel, Pearson Education, Latest Edition.
3. Flask Web Development, Miguel Grinberg, O'Reilly Media, 2nd Edition.
4. Building Machine Learning Powered Applications, Emmanuel Ameisen, O'Reilly Media, 1st Edition.

Reference Books:

1. Learning Web Design, Jennifer Niederst Robbins, O'Reilly Media, 5th Edition.
2. Web Development with Node and Express, Ethan Brown, O'Reilly Media, 2nd Edition.
3. MongoDB: The Definitive Guide, Kristina Chodorow, O'Reilly Media, 3rd Edition.
4. Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow, Aurélien Géron, O'Reilly Media, 3rd Edition.
5. Designing Web APIs, Brenda Jin, Saurabh Sahni, and Amir Shevat, O'Reilly Media, 1st Edition.

e-Books/E-learning References:

1. **https://developer.mozilla.org/en-US/**
2. **http://www.meanacademy.in/**
3. https://**www.w3schools.com/**
4. **https://www.tutorialspoint.com/**
5. **https://react.dev/learn**
6. **https://flask.palletsprojects.com/en/stable/**
7. **https://www.mongodb.com/docs/**
8. **https://scikit-learn.org/stable/**

Links to online SWAYAM/NPTEL Courses:

1. https://onlinecourses.swayam2.ac.in/e-learning/preview/nou24_cs09
2. https://onlinecourses.swayam2.ac.in/e-learning/preview/ntr25_ed123

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PEC-321D-AIM
Course Name: Advanced Database Management System

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

Prerequisite Courses, if any: Database Management System

Companion Course, if any:

Course Objectives:

1. To learn the concepts of parallel and distributed database systems.
2. To understand NoSQL databases and modern data storage models.
3. To study data warehousing and OLAP technologies for analytical processing.
4. To understand data mining concepts and knowledge discovery techniques.
5. To explore advanced database models and emerging database technologies for modern applications.

Course Outcomes:

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

CO1: Explain the concepts and architectures of parallel and distributed database systems.

CO2: Discuss NoSQL database models and their applications in managing large-scale data.

CO3: Describe data warehousing concepts and OLAP technologies.

CO4: Describe the knowledge discovery process and data mining functionalities for extracting useful information from data.

CO5: Describe advanced database models and emerging database technologies used in AI-oriented applications.

Course Contents

Unit No: I	**Parallel and Distributed Databases**	**8 Hours**

Introduction to Parallel Databases, Architectures for parallel databases, Parallel query evaluation, Parallelizing individual operations, Parallel query optimizations. Introduction to distributed databases, Distributed DBMS architectures, Storing data in a Distributed DBMS, Distributed catalog management, Distributed Query processing, Updating distributed data, Distributed transactions, Distributed Concurrency control and Recovery.

Exemplars /Case Studies	**Case Study 1:** Analysis of a Distributed E-Commerce Database System for order processing, distributed query execution, transaction management, and data consistency across multiple locations. **Case Study 2:** Analysis of a Parallel Database System for Large-Scale Retail Analytics focusing on parallel query evaluation, query optimization, and high-performance data processing.

Unit No: II	**NoSQL Databases**	**8 Hours**

Introduction to NoSQL Databases, Types of NoSQL Databases, Document-Oriented Databases: MongoDB, Column-Family Databases: Apache Cassandra, Semi-Structured Data and Data Exchange Formats (XML and JSON, Applications of NoSQL Databases in AI & ML

Exemplars/Case Studies	• Social Media Data Management using MongoDB • Healthcare Data Management using Apache Cassandra	
Unit No: III	**Data Warehousing**	**8 Hours**
Introduction to Data Warehousing: need, Characteristics of Data Warehouse, Advantages and Limitations, Data Warehouse Architecture and Components, Data Warehouse Modeling, OLAP and OLAP Operations, Data Warehousing for Decision Support Systems, Business Intelligence: concepts and applications		

Exemplars/Case Studies	**Case Study 1: Retail Sales Data Warehouse**
	• Integrating sales data from multiple stores into a centralized data warehouse.
	• Designing a Star Schema with:
	  o Fact Table: Sales
	  o Dimension Tables: Product, Customer, Store, Time
	• Performing OLAP operations such as Roll-up and Drill-down to analyze sales trends.	

Unit No: IV	**Data Mining and Knowledge Discovery**	**8 Hours**
Introduction to Data Mining, Knowledge Discovery in Databases (KDD), Data mining architecture, Predictive Data Mining, Descriptive Data Mining, Outlier Analysis (Overview), Data mining tools and applications		
Exemplars/Case Studies	Case Study: Analysis of Customer Purchase Data for Knowledge Discovery and Pattern Identification using the KDD Process.	
Unit No: V	**Advanced Database Models and AI Applications**	**6 Hours**
Active Databases and Triggers, Temporal Databases, Spatial Databases, Graph Databases and Knowledge Graphs, Emerging AI-Oriented Database Applications: Genome and Bioinformatics Data Management, Time-Series Databases for Sensor and IoT Data, Vector Databases		
Learning Resources		
Textbooks: 1. Silberschatz A., Korth H., Sudarshan S, Database System Concepts, McGraw Hill Publication, ISBN 0-07-120413-X, Sixth Edition. 2. Ramez Elmasri and Shamkant B. Navathe “Fundamentals of Database System”7th Edition		
Reference Books: 1. Kristina Chodorow, Michael Dirolf, “MongoDB: The Definitive Guide”, O’Reilly Publications 2. Jiawei Han, Micheline Kamber, Jian Pei, “Data Mining: Concepts and Techniques”, Elsevier 3. M. Tamer Özsu, Patrick Valduriez, “Principles of Distributed Database Systems”, Prentice Hall, 1999. 4. S. K. Singh, Database Systems: Concepts, Design and Application, Pearson Publication, ISBN-978-81-317-6092-5		
e-Books: Mario Piattini, Oscar Diaz “Advanced Database Technology and Design”- online book.		

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PEC-322A-AIM
Course Name: Cloud Computing Lab (Elective I Lab 3)

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

Course Objectives:

1. Understand the concepts of cloud computing, virtualization, and cloud service models through practical implementation.
2. Develop and manage virtualized environments using virtual machines and analyze resource utilization in cloud infrastructures.
3. Deploy and manage web applications using cloud-native technologies, containerization, and orchestration platforms.
4. Apply cloud computing platforms and tools to develop AI/ML-enabled applications for solving real-world problems.
5. Design and implement scalable cloud-based solutions using modern cloud technologies and open-source tools.

Course Outcomes:

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

CO1: Create, configure, and manage virtual machines to demonstrate virtualization concepts and efficient resource utilization.

CO2: Deploy and manage cloud-based services and web applications using Linux servers and networking concepts.

CO3: Implement containerized applications and deploy them using Docker and Kubernetes to achieve portability, scalability, and high availability.

CO4: Develop and deploy cloud-enabled AI/ML applications for data analysis, prediction, and Decision-making.

CO5: Design complete cloud-based solutions by integrating virtualization, deployment, orchestration, and AI/ML technologies for 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.

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

6. 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.
7. 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.
8. 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.

The laboratory shall be equipped with computing systems having adequate processing capability, memory, storage, and internet connectivity to support virtualization, containerization, and cloud-based experiments. Practical assignments may be implemented using Linux/Windows operating systems, Python programming language, cloud platforms, virtualization tools, container technologies, and relevant AI-ML frameworks. Students shall use appropriate development environments, databases, and software tools required for implementation and deployment of applications. Laboratory systems should support virtual machine creation, container execution, and cloud service access. Open-source software and academic cloud resources are recommended for conducting experiments and mini-projects.

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

For lab assignments use necessary tools like Python, Google Colab , Streamlit , Minikube, Docker Desktop, Apache/Nginx Web Server, Oracle Virtual Box.

1. Case study on Amazon or Microsoft Azure Cloud. --CO2
2. An organization wants to reduce hardware costs by running multiple operating systems on a single machine.
3. Create and configure virtual machines to demonstrate server virtualization and resource sharing. -CO1
4. Monitor CPU, memory, disk, and network utilization of virtual machines under different workloads and analyse resource allocation efficiency. --CO1
5. Deploy a web application on a Linux-based server and configure it to provide services to users over a network. --CO2

6.	Package a Python-based application into a Docker container and deploy it independently of the host operating system. --CO3
7.	Study the architecture and working of Kubernetes clusters for deploying containerized applications with multiple replicas to achieve scalability and high availability. Analyze the role of pods, deployments, services, and replica sets in Kubernetes. --CO3
8.	Find a procedure to transfer a files from one virtual machine to another virtual machine. --CO2
9.	Develop a cloud-based sentiment analysis platform that classifies social media posts as positive, negative, or neutral and visualizes public opinion trends. --CO2 , CO3, CO4, CO5
10.	Develop a cloud-based application that predicts student placement chances based on academic performance, technical skills, internships, and aptitude scores. --CO1, CO2, CO3, CO4, CO5

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PEC-322B -AIM
Course Name: Natural Language Processing Lab (Elective I Lab 3)

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

Course Objectives:

1. To understand fundamental NLP techniques for text preprocessing and linguistic analysis.
2. To understand significance of natural language processing in solving real-world problems.
3. To learn NLP-based models for text classification and information retrieval.
4. To learn NLP tools and libraries for natural language processing tasks and applications.

Course Outcomes:

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

CO1: Apply text preprocessing techniques for natural language data preparation.

CO2: Analyze text representations using feature extraction and embedding methods.

CO3: Implement POS tagging, semantic analysis, and text classification techniques.

CO4: Implement NLP-based solutions for sentiment analysis and information retrieval tasks.

CO5: Use NLP tools and libraries to solve real-world language processing problems.

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. Recommended Tools for the implementation of assignments: Python, NLTK etc.

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 the faculty concerned 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 them 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.
List of Assignments
PART A (All assignments are mandatory)
1. Implement tokenization techniques for sentence and word segmentation using NLTK and SpaCy.
2. Implement Hidden Markov Model (HMM) for Part-of-Speech tagging.
3. Implement TF-IDF and Bag-of-Words models for text representation.
4. Write a program for document retrieval using the Vector Space Model.
5. Perform linguistic analysis using SpaCy and Text Blob libraries.
PART B (Anyone)
1. Mini Project - Automated Part-of-Speech Tagging and Sentence Analysis Tool
2. Mini Project - Intelligent Digital Library Search System Using NLP -
3. Mini Project - Product Review Sentiment Analysis System
4. Mini Project – Intelligent Information Retrieval and Question Answering System

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PEC-322C -AIM
Course Name: Web Application Development Lab (Elective I Lab 3)

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

Course Objectives:

1. To develop responsive and interactive web applications using modern web technologies.
2. To implement client-side programming using JavaScript and ReactJS.
3. To design RESTful web services using Flask.
4. To integrate databases and machine learning models into web applications.

Course Outcomes:

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

CO1: Develop responsive web pages using HTML, CSS3, JavaScript and Bootstrap.

CO2: Build interactive client-side applications using ReactJS.

CO3: Develop REST APIs using Flask.

CO4: Integrate databases and machine learning models with web 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. Experiments to be conducted are grouped in Part A, B and C.

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
2. parameters such as timely conduction of practical assignment, methodology adopted for implementation of practical assignment, timely submission of assignment in the form of
3. handwritten write-up along with results of implemented assignment, attendance etc.
4. 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.
5. 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.
List of Assignments
Virtual Laboratory: **https://www.w3schools.com/js/default.asp** **https://spoken-tutorial.org/watch/JavaScript/Overview+of+JavaScript/English/**
PART A Web Technologies (Any 3)
1. Design a responsive personal portfolio website using HTML5, CSS3 and Bootstrap. --CO1
2. Develop a student registration form with client-side validation using JavaScript. -- CO1
3. Develop College Event Management Website using Bootstrap components. -- CO1
4. Interactive Image Gallery using JavaScript DOM Manipulation. -- CO1
PART B Client-Side Development using ReactJS (Any 3)
1. Develop a task management application using ReactJS. --CO2
2. Develop a ReactJS-based weather dashboard that consumes data from a public weather API. -- CO2
3. Develop an interactive dashboard using ReactJS that visualizes student academic performance using charts and graphs. -- CO2
4. Design and implement a Single Page Application (SPA) using React Router. -- CO2
PART C Backend Development and AI Integration (Any 3)
1. Develop a RESTful API using Flask for managing student records. The API should support Create, Read, Update, and Delete (CRUD) operations and return responses in JSON format. -- CO3
2. Develop a web application using Flask and SQLite/MySQL for managing library resources. The system should allow adding, updating, deleting, and searching books, along with maintaining issue and return records. -- CO4
3. Develop a machine learning-based web application using Flask. Train any machine learning model (e.g. regression model) for house price prediction and integrate this trained model with a web interface -- CO4
4. . Develop a web application using Flask that recommends elective courses, internships, or skill-development resources based on student interests and academic performance. Store student data in a database and display personalized recommendations through a web interface. – CO4

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PEC-322D -AIM
Course Name: Advanced Database Management System Lab (Elective I Lab 3)

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

Course Objectives:

1. To understand advanced database concepts including NoSQL databases and data warehousing.
2. To install, configure, and manage MongoDB databases.
3. To learn MongoDB query processing and data management techniques.
4. To design and develop database applications using MongoDB.
5. To design conceptual data warehouse models using Star and Snowflake schemas.

Course Outcomes:

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

CO1: Apply NoSQL database concepts and advanced database techniques for data management applications.
CO2: Install, configure, and manage MongoDB database systems.
CO3: Apply data management and query processing techniques using MongoDB.
CO4: Design and develop database applications using MongoDB and suitable front-end technologies.
CO5: Design conceptual data warehouse models using Star and Snowflake schemas for real-world applications.

Guidelines for Instructor's Manual

The faculty member should prepare the laboratory manual for all the assignments and make it available to students and laboratory instructor/assistant. The laboratory manual should include the course objectives, course outcomes, university syllabus, assessment guidelines, relevant database concepts, problem statements, and implementation guidelines.

Guidelines for Student's Lab Journal

1. The laboratory assignments shall be submitted by students in the form of a journal. The journal should include a prologue, certificate, table of contents, and write-up of each assignment containing the title, objectives, problem statement, course outcomes addressed, date of completion, assessment grade/marks and assessor's signature, relevant theory/concepts, implementation details, observations/results, and conclusion.
2. The journal should contain relevant outputs, screenshots, query results, schema designs, and mini project implementation details wherever applicable.
3. Students are expected to understand the underlying concepts and methodologies associated with each assignment.
4. The term work assessment shall be based on the timely completion and quality of assignments, maintenance of the laboratory journal, and performance in continuous assessment activities.
5. All the assignment mentioned in the syllabus must be conducted.

Guidelines for Lab /TW Assessment

1. The term work shall be assessed based on the timely completion and submission of laboratory assignments, quality of journal maintenance, attendance, and performance in continuous assessment activities.
2. Assessment shall consider the correctness of MongoDB operations, query processing techniques, data warehouse schema design, and documentation of the assignments.
3. The faculty member shall evaluate the student's understanding of MongoDB concepts, database operations, query processing techniques, data warehouse concepts, and implementation aspects associated with the completed assignments.
4. The Mini Project shall be evaluated based on database design, implementation, functionality, report quality,

and demonstration.

Guidelines for Laboratory Conduction

Assignments 1, 2, and 3 (MongoDB-based assignments) shall be performed individually by each student.

Assignment 4 (Data Warehouse Design) and Assignment 5 (Mini Project) may be carried out in groups of 3 or 4 students. The Mini Project may be developed using any suitable front-end technology; however, MongoDB shall be used as the back-end database.

List of Assignments

1. **MongoDB Database Creation and Data Manipulation** -- CO1, CO2, CO3

Create a suitable MongoDB database and perform data manipulation operations including document insertion (single and bulk insertion), document validation, document modification (update, replacement, upsert), and document deletion.

2. **MongoDB Query Processing**-- CO1, CO3

Execute at least 15 queries demonstrating data retrieval (find, findOne), query criteria and conditional operators, logical query operations (AND, OR, NOT), type-specific queries (null values, regular expressions, array queries), advanced queries ($where queries), and cursor operations (sort, limit, skip, projection).

3. **MongoDB Aggregation, MapReduce and Indexing** -- CO1, CO3

Implement data processing and optimization techniques in MongoDB using a suitable dataset and demonstrate:

- Aggregation framework
  - Data filtering and grouping
  - Data projection
  - Sorting of aggregated results
- MapReduce processing for data analysis
- Index management
  - Creation of different types of indexes
  - Deletion of indexes
- Query performance analysis
  - Comparison of query execution with and without indexes
  - Use of explain() to demonstrate the advantages of indexing

4. **Data Warehouse Design** -- CO1, CO5

Design a conceptual data warehouse model for a suitable application using Star and Snowflake schemas.

5. Mini **Project** -- CO1, CO2, CO3, CO4

Develop a database application using any suitable front-end technology and MongoDB as the back-end database. The application should demonstrate database connectivity, CRUD operations, query processing, report generation, and user interface design.

Students shall work in teams of 3 to 4 members and submit a project report including the project title, abstract, hardware and software requirements, database design, source code, user interface screenshots, results, and conclusion.

Reference Books

1. Silberschatz A., Korth H., Sudarshan S., "Database System Concepts", 6th Edition, McGraw Hill Publishers, ISBN 0-07-120413-X.
2. Kristina Chodorow, MongoDB The definitive guide, O'Reilly Publications, ISBN:978-93-5110-269- 4,2nd Edition.

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: MDM-33-AIM
Course Name: AIoT Product Development

Teaching Scheme	Credits	Examination Scheme
Tutorial : 1 hr/wk	1	TW - 25 Marks
Practical : 4 hrs/wk	2	OR - 50 Marks

Prerequisite Courses, if any: Foundation of Artificial Intelligence (PCC-203- AIM), Internet of Things (MDM -271- AIM), Design Thinking and Idea Lab (VSE-102)

Companion Course, if any:

Course Objectives:

The objectives of this course are:

1. To understand the architecture, components, and workflow of Artificial Intelligence enabled Internet of Things (AI-IoT) systems.
2. To enable students to acquire, process, and analyze real-time sensor data using IoT devices, cloud platforms, and data analytics tools.
3. To develop understanding of machine learning concepts and apply them on IoT-generated datasets for intelligent decision-making.
4. To facilitate deployment of AI models on edge devices for real-time inference and automated operation.
5. To enable students to apply design thinking approach to design, develop, and demonstrate complete AI-IoT solutions for real-world applications through product-based learning

Course Outcomes:

After successful completion of the course, students will be able to:

CO1: Configure IoT hardware and acquire real-time sensor data using Arduino-based systems.

CO2: Develop IoT applications with cloud integration and real-time dashboard visualization.

CO3: Apply data preprocessing, visualization, and analysis techniques on IoT datasets using Python tools.

CO4: Develop and evaluate machine learning models using IoT data for prediction and intelligent decision-making.

CO5: Design and implement complete AI-IoT systems integrating sensors, cloud platforms, machine learning, and edge intelligence for real-world applications.

Course Contents For Tutorial

Unit No: I	AI-IoT Problem Identification and Product Ideation	03 Hours

Introduction to AI-IoT systems, real-world applications in agriculture, healthcare, energy, parking, smart classroom and safety domains. Problem identification techniques, stakeholder analysis, user requirement gathering, feasibility analysis, product ideation, and design thinking approach.

Exemplars/Case Studies	Smart Agriculture Monitoring System using AI and IoT
Reference Books	1. Arshdeep Bahga and Vijay Madisetti, Internet of Things: A Hands-on Approach 2. Rajkumar Buyya, Internet of Things: Principles and Paradigms Tim Brown – Change by Design (Design Thinking)

Unit No: II	AI-IoT System Design and Hardware Mapping	03 Hours

AI-IoT architecture, sensor-actuator mapping, Arduino/ESP8266 systems, communication

protocols (MQTT/HTTP), cloud platforms, dashboard design, system workflow.		
Exemplars/Case Studies	Smart Energy Monitoring and Intelligent Analytics System	
Reference Books	1. Adrian McEwen, Designing the Internet of Things 2. Bahga & Madisetti – IoT: A Hands-on Approach 3. Simon Monk – Programming Arduino: Getting Started with Sketches (McGraw-Hill Education) 4. Brian Evans – Beginning Arduino Programming (Apress)	
Unit No: III	**Data Acquisition and Dataset Design**	**03 Hours**
IoT data lifecycle, dataset design, CSV structure, sampling rate, labeling, preprocessing, missing values, noise handling, and storage planning.		
Exemplars/Case Studies	Predictive Maintenance System Using Industrial Sensor Data	
Reference Books	1. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow 2. Andreas Müller, Introduction to Machine Learning with Python 3. Wes McKinney – Python for Data Analysis	
Unit No: IV	**AI Model Selection and Intelligent Decision Making**	**03 Hours**
Machine learning workflow, feature selection, classification/regression models, training/testing, evaluation metrics, confusion matrix, real-time prediction pipeline, edge AI basics.		
Exemplars/Case Studies	Edge AI Based Health Monitoring and Alert System	
Reference Books	1. Aurélien Géron – Hands-On Machine Learning 2. Andreas Müller – Introduction to Machine Learning with Python 3. Jake VanderPlas – Python Data Science Handbook	
Unit No: V	**AI-IoT Product Integration, Deployment and Documentation**	**03 Hours**
System integration, testing, validation, cost estimation, scalability, edge vs cloud trade-off, documentation, report writing, and presentation, innovation and entrepreneurship opportunities in AI-IoT.		
Exemplars/Case Studies	Smart City Infrastructure using AI-IoT Technologies	
Reference Books	1. Bahga and Madisetti, *Internet of Things: A Hands-On Approach* 2. Industry White Papers and Research Reports on AI-IoT Applications 3. Pete Warden – *TinyML: Machine Learning with TensorFlow Lite* 4. Rajkumar Buyya – *Cloud and IoT Integration Concepts*	
Learning Resources		
Textbooks 1. Bahga, A., & Madisetti, V. (2015): Internet of Things: A Hands-on Approach. Universities Press. ISBN: 978-8173719547. 2. Buyya, R., & Dastjerdi, A. V. (2016): Internet of Things: Principles and Paradigms. Morgan Kaufmann. ISBN: 978-0128053959. 3. Géron, A. (2022): Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd Edition). O'Reilly Media. ISBN: 978-1098125974. 4. Warden, P., & Situnayake, D. (2019).:TinyML: Machine Learning with TensorFlow Lite on		

Arduino *and Ultra-Low-Power Microcontrollers*. O'Reilly Media. ISBN: 978-1492052043.

Reference Books

1. McEwen, A., & Cassimally, H. Designing the Internet of Things. Wiley.
2. Raj, P., & Raman, A. C. The Internet of Things: Enabling Technologies, Platforms, and Use Cases. CRC Press.
3. Müller, A. C., & Guido, S. Introduction to Machine Learning with Python. O'Reilly Media.
4. Chollet, F. Deep Learning with Python (2nd Edition). Manning Publications.
5. Tim Brown – Change by Design: How Design Thinking Transforms Organizations (HarperBusiness)
6. Simon Monk – Programming Arduino: Getting Started with Sketches (McGraw-Hill Education) Industry White Papers on AI-IoT Applications (IBM, AWS, Microsoft Azure IoT Documentation)

e-Books

1. Artificial Intelligence: Foundations of Computational Agents: https://artint.info/
2. Practical Deep Learning for Cloud, Mobile, and Edge: https://www.oreilly.com/library/view/practical-deep-learning/9781492034858/
3. Introduction to Data Science: https://open.umn.edu/opentextbooks

Links to online SWAYAM/NPTEL Courses

1. Introduction to Internet of Things (NPTEL) – Prof. Sudip Misra, IIT Kharagpur: https://onlinecourses.nptel.ac.in/noc25_cs64/
2. Machine Learning (NPTEL) – Prof. Balaraman Ravindran, IIT Madras: https://onlinecourses.nptel.ac.in/noc25_cs81/
3. Data Analytics with Python (NPTEL) – IIT Roorkee: https://onlinecourses.nptel.ac.in
4. Python for Data Science (SWAYAM): https://swayam.gov.in
5. Introduction to Industry 4.0 and Industrial Internet of Things By Prof. V. Kamakoti and Team, IIT Madras: https://onlinecourses.nptel.ac.in
6. Design Thinking – A Primer (NPTEL) – IIT Madras (Prof. Ashwin Mahalingam & Prof.
7. Bala Ramadurai):https://onlinecourses.nptel.ac.in/noc26_mg51/

Guidelines for Instructor's Manual

1. The faculty member should prepare an Instructor's Manual covering all laboratory assignments and make it available to students and laboratory instructors/assistants.
2. The manual should include Course Objectives, Course Outcomes, University Syllabus, Laboratory Conduction and Assessment Guidelines.
3. Each assignment should contain Objective, Learning Outcome, Requirements, Implementation Steps, and Expected Results.
4. The manual should provide guidance for IoT hardware interfacing, cloud platform integration, AI/ML implementation, and AI-IoT product development.
5. The manual should include the Project Report Writing Format and relevant references.

Guidelines for Lab /TW Assessment

1. Assessment shall be based on continuous performance throughout the semester.
2. Evaluation parameters shall include:
   - Timely completion of assignments
   - Correctness of implementation
   - Sensor interfacing and cloud integration
   - Data analytics and AI/ML implementation
   - Documentation quality and presentation skills
   - Innovation and problem-solving approach

- Attendance and contribution in product development.

3. Students shall demonstrate the working of assignments and explain the adopted methodology.
4. Examiners may ask questions related to theory, implementation, cloud services, AI/ML models, and system integration.
5. Knowledge of hardware, software tools, IoT platforms, cloud services, coding standards, and testing procedures shall be assessed.

Guidelines for Student's Lab Journal

1. Students shall maintain a record of all laboratory assignments performed during the semester and submit a Project Report for the AI-IoT mini product.
2. The Project Report should include Title, Objectives, Problem Statement, System Architecture, Hardware & Software Requirements, IoT/Cloud Platform Integration, AI/ML Implementation, Results, Conclusion, and References.
3. Relevant screenshots, sensor outputs, cloud dashboards, datasets, AI/ML results, and project demonstration records should be included wherever applicable.
4. Practical Examination shall be based on the term work, assignments completed, and AI-IoT mini product developed by the students.
5. Students are expected to understand the theory, implementation methodology, and outcomes of the assignments and project work.
6. The Project Report should be completed in all respects, certified by the concerned faculty, and submitted before the practical examination.
7. All assignments and the AI-IoT mini product prescribed in the syllabus must be completed.

Guidelines for Lab /TW Assessment

1. Assessment shall be based on continuous performance throughout the semester.
2. Evaluation parameters shall include timely completion of assignments, correctness of implementation, hardware interfacing, IoT/cloud integration, AI/ML implementation, documentation quality, attendance, and participation.
3. Students shall demonstrate the working of assignments and explain the adopted methodology.
4. Examiners may ask questions related to theory, implementation, cloud services, AI/ML models, edge AI, and system integration.
5. Ideation, Critical Thinking, Innovation, problem-solving approach, project execution, and presentation skills shall also be considered during assessment.

Guidelines for Laboratory Conduction

1. The laboratory shall be conducted in a product development mode, where students progressively develop an AI-IoT based mini product throughout the semester.
2. Students shall work in groups of 2–3 members and follow a design thinking approach for problem identification, design, implementation, testing, and validation.
3. Assignments shall be carried out sequentially to support end-to-end AI-IoT product development, including data acquisition, cloud integration, analytics, and intelligent decision-making.
4. The instructor may assign different application domains such as Smart Agriculture, Smart Healthcare, Smart Energy Monitoring, Smart Parking, Smart Classroom, Environmental Monitoring, and Industrial Automation
5. Recommended Hardware: Arduino UNO R3, sensors (DHT11/DHT22, LDR, PIR, Ultrasonic, Soil Moisture, Gas/Smoke), actuators (LED, buzzer, relay, motors), and ESP8266/Bluetooth modules.
6. Recommended Software and Platforms: Arduino IDE, Python, Pandas, NumPy, Matplotlib, Scikit-learn, ThingSpeak, Adafruit IO, and Blynk.
7. The use of open-source software, IoT platforms, cloud services, and AI/ML tools is encouraged.

Guidelines for Practical Examination

1. The practical examination shall be jointly conducted by internal and external examiners.

2. The examination shall be based on the laboratory assignments and the AI-IoT mini product developed during the semester.
3. Maximum weightage shall be given to the successful implementation, demonstration, testing, and validation of the developed AI-IoT solution.
4. Examiners may ask questions related to problem identification, system design, sensor interfacing, IoT/cloud platform integration, data analytics, machine learning, edge deployment, and product implementation.
5. Evaluation shall be based on the functionality of the developed product, technical understanding, innovation, documentation, presentation, and overall contribution of the student to the project.

List of Assignments:

Assignment 1
Title: Arduino Hardware Setup and Sensor Data Acquisition.
Objective: To configure Arduino hardware platforms and collect real-time sensor data.

Task Description:

- Connect Arduino UNO and install Arduino IDE.
- Interface sensors such as DHT11/DHT22, LDR, Ultrasonic, PIR, or Soil Moisture sensor.
- Acquire real-time sensor readings.
- Display sensor values using Serial Monitor.
- Store sensor observations in CSV format.
- Collect at least 100 sensor observations for analysis.

Learning Outcome (LO1):

Configure Arduino UNO and acquire real-time data from sensors such as DHT11, LDR, Ultrasonic, PIR or

Soil Moisture.

Assignment 2
Title: Arduino-Based IoT Cloud Connectivity and Dashboard Creation
Objective: To establish communication between IoT devices and cloud platforms.

Task Description:

- Connect Arduino-based IoT device to a Wi-Fi network using ESP8266 or equivalent module.
- Transmit sensor data to IoT cloud platforms such as ThingSpeak, Adafruit IO, or Blynk.
- Create cloud dashboards for real-time monitoring.
- Visualize sensor readings using charts and gauges.
- Monitor live sensor data remotely.

Learning Outcome (LO2):
Interface Arduino with communication modules, transmit sensor data to cloud platforms, and visualize data through interactive dashboards.

Assignment 3
Title: Sensor Data Storage, Preprocessing and Analytics using Python
Objective: To perform storage, preprocessing and visualization of IoT data.

Task Description:

- Import IoT sensor datasets into Python.
- Store datasets in CSV or database format.
- Handle missing values and remove outliers.
- Perform statistical analysis (mean, median, standard deviation).
- Generate graphs and visualizations using Pandas and Matplotlib.
- Interpret trends and patterns in sensor data.

Learning Outcome (LO3):
Store, preprocess, analyze and visualize IoT datasets using Python tools and libraries.

Assignment 4
Title: Machine Learning Model Development for IoT Data.
Objective: To develop predictive machine learning models using IoT-generated datasets.

Task Description:
- Select suitable IoT datasets.
- Perform feature selection and data preparation.
- Train machine learning models such as Decision Tree, Random Forest, or Logistic Regression.
- Evaluate model performance using appropriate metrics.
- Compare the performance of different machine learning algorithms.
- Predict outcomes such as occupancy detection, irrigation requirement, fire risk, or device state.

Learning Outcome (L04):
Build and evaluate machine learning models using Arduino-generated sensor datasets.

Assignment 5
Title: Real-Time AI Prediction using live sensor Data
Objective: To integrate trained machine learning models with live sensor streams for real-time prediction.
Task Description:
- Connect live sensor data streams to the trained AI model.
- Perform real-time prediction using incoming sensor data.
- Display prediction results through dashboards or monitoring interfaces.
- Analyze prediction confidence, accuracy, and response latency.
- Validate model performance under real-world conditions.

Learning Outcome (LO5):
Integrate trained AI models with live sensor data and perform real-time intelligent prediction.

Assignment 6
Title: Edge AI Deployment and Device Automation.
Objective: To deploy AI models on embedded IoT devices and implement intelligent automation.
Task Description:
- Deploy lightweight AI models on Arduino-compatible edge devices.
- Integrate sensors, actuators, and decision-making logic.
- Trigger LEDs, buzzers, relays, or motors based on AI predictions.
- Measure system response time and resource utilization.
- Compare edge-based and cloud-based processing approaches.
- Evaluate the effectiveness of automated decision-making.

Learning Outcome (LO6):
Implement AI-driven automation and decision-making using Arduino-based IoT devices.

Assignment 7
Title: AI-IoT Mini Product Development and Integration.
Objective: To design, develop, and demonstrate a complete AI-IoT solution integrating sensing, cloud services, analytics, and intelligence.
Task Description:
Design and implement an AI-IoT based mini product such as
- Smart Agriculture Monitoring System
- Smart Health Monitoring System
- Smart Classroom Monitoring System
- Smart Waste Management System
- Smart Energy Monitoring System
- Smart Parking System
- Predictive Maintenance System etc.

Design and implement an AI-IoT based mini product such as

- Sensor data acquisition layer
- Communication and IoT cloud layer
- Data storage and analytics layer
- AI/ML prediction layer
- Dashboard, alert, and automation features
- Final demonstration and validation

Learning Outcome (LO7):
Design, develop, and demonstrate a complete AI-enabled IoT solution integrating sensors, cloud services, analytics, and intelligent decision-making.

Assignment 8
Title: AI-IoT Product Documentation and Project Report
Objective: To prepare professional technical documentation for the developed AI-IoT product.

Task Description:

Prepare a project report containing:

- Problem Statement
- Objectives
- System Architecture
- Hardware Components
- Software Tools and Technologies
- IoT Platform Integration
- AI/ML Model Description
- Experimental Results and Performance Analysis
- Challenges and Limitations
- Future Scope
- Conclusion
- References

Prepare and deliver a technical presentation demonstrating the developed AI-IoT solution.

Learning Outcome (LO8):
Prepare professional technical documentation, project reports, and presentations for AI-IoT systems.

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: **ELC-342-AIM**
Course Name: **Technical Seminar**

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

Prerequisites: Software Engineering, Community Engagement Project,

Course Objectives:

Technical Seminar should make the student attain skills like:

1. To gather literature in a focused area related to Artificial Intelligence and Machine Learning.
2. To analyze and summarize the literature to identify state-of-the-art techniques and technologies.
3. To evaluate existing approaches and identify research gaps and future directions.
4. To formulate a potential project problem statement based on literature findings.
5. To prepare and present technical content in a scientific and professional manner.

Course Outcomes:

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

CO1: Apply appropriate research tools and databases to collect technical literature in the selected domain.

CO2: Analyse research papers to identify methodologies, datasets, tools, experimental setup, and outcomes.

CO3: Compare and analyze various techniques and approaches reported in the literature.

CO4: Evaluate existing work to identify research gaps and opportunities for future investigation.

CO5: Develop a technical seminar report and presentation by synthesizing information from multiple research sources and formulate a project problem statement.

Guidelines for Seminar Selection and Presentation

- Students shall identify an area or topic in Artificial Intelligence and Machine Learning based on emerging technologies, recent research trends, industrial requirements, and consultation with the institute guide.
- Seminar topics shall preferably be aligned with the proposed Final Year Project domain.
- Students must review sufficient literature including reference books, journal articles, conference papers, patents, white papers, magazines, technical reports, and web resources relevant to the selected topic.
- Seminar topics should be based on recent developments and research trends. The guide shall approve the topic after evaluating novelty, technical depth, applicability, and scope for future work.
- Research articles may be referred from IEEE Xplore, ACM Digital Library, Springer, Elsevier ScienceDirect, Scopus, arXiv, ResearchGate, Google Scholar, and other recognized digital libraries.
- Each student shall review a minimum of eight to ten recent research papers.
    - Students shall prepare a comparative analysis highlighting:
    - Problem Statement
    - Methodology
    - Dataset Used
    - Performance Metrics
    - Strengths and Limitations
    - Research Gap
    - Student shall present the study as an individual seminar of 15–20 minutes duration followed by a Question–Answer session.
    - Guide shall ensure that the literature survey and review process is conducted systematically.

- Attendance of all students during seminar presentations is mandatory.

Suggested Timeline Throughout the Semester

Week 01:
Introduction to technical papers, literature survey techniques, research databases, and plagiarism awareness.
Week 02:
Selection of seminar topic and collection of relevant research papers.
Week 03–04:
Detailed study of selected research papers and preparation of paper summaries.
Week 05:
Review of literature findings by Guide/Panel.
Week 06–07:
Comparative analysis of existing approaches and identification of alternative techniques.
Week 08–09:
Identification of research gaps, future directions, and formulation of tentative project problem statement.
Week 10:
Preparation of seminar report.
Week 11:
Preparation and delivery of seminar presentation.
Week 12:
Report verification, evaluation, and submission.

Guidelines for Seminar Report

1. Each student shall submit a seminar report of approximately 15–20 pages in the prescribed format duly certified by the Guide and Head of the Department.
2. The report shall contain:
    a. Title Page
    b. Seminar Certificate
    c. Abstract and Keywords
    d. Acknowledgement
    e. Table of Contents
    f. List of Figures and Tables
    g. Introduction
    h. Literature Survey
    i. Comparative Analysis of Existing Approaches
    j. Research Gap Identification
    k. Proposed Project Problem Statement
    l. Future Scope and Expected Outcomes
    m. Conclusion
    n. References (IEEE Format)
    o. Plagiarism Report
3. Students are encouraged to use open-source tools for report writing, citation management, and plagiarism checking.

Guidelines for Term Work Assessment

1. A panel constituted by the Seminar Coordinator (including the Guide) shall evaluate the seminar activity.
2. Evaluation shall be based on the following rubric:
    i. Relevance of Topic – 05 Marks

ii. Depth of Literature Survey –	05 Marks
iii. Comparative Analysis and Research Gap Identification –	05 Marks
iv. Seminar Report –	05 Marks
v. Presentation and Technical Discussion –	05 Marks
	Total: 25 Marks

Reference Book:

- Andrea J. Rutherford, Basic Communication Skills for Technology, Pearson Education Asia.
- Lesikar, Lesikar's Basic Business Communication, Tata McGraw Hill.

Textbook:

- Sharon J. Gerson and Steven M. Gerson, Technical Writing: Process and Product, Pearson Education.

Reference Resources:

1. IEEE Xplore Digital Library
2. ACM Digital Library
3. SpringerLink
4. Elsevier ScienceDirect
5. Google Scholar
6. ResearchGate
7. arXiv

SYLLABUS
SEMESTER - VI

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PCC-351 -AIM
Course Name: Computer Vision

Teaching Scheme	Credits	Examination Scheme
Theory: 3 Hrs/Wk	3	CCE - 30 Marks ESE - 70 Marks

Prerequisite Courses:
Students should know
1. vectors, linear algebra (i.e., matrix operations, solution of linear equations)
2. Programming language (e.g., C, Matlab, Python etc)

Companion Course, if any:

Course Objectives:
1. Understand the principles of image formation, image representation, and computer vision systems.
2. Apply image processing, segmentation, and feature extraction techniques for visual data analysis.
3. Analyze object recognition, motion analysis, and three-dimensional scene understanding methods.
4. Explore modern computer vision architectures for object detection, segmentation, and multimodal vision tasks.
5. Develop AI-driven solutions for real-world applications involving image and video data.

Course Outcomes:
On completion of the course, students will be able to–
CO1: Apply image preprocessing and enhancement techniques for visual data analysis.
CO2: Apply segmentation and feature extraction techniques for image analysis and recognition.
CO3: Analyze object recognition methods and interpret three-dimensional scene information.
CO4: Apply motion analysis, object detection, and tracking techniques for video understanding.
CO5: Evaluate advanced AI-driven computer vision techniques and develop intelligent visual applications.

Course Contents

Unit No: I	Fundamentals of Computer Vision and Image Processing	9 Hours

Introduction to Computer Vision, Applications of Computer Vision in AI Systems, Human Vision vs Computer Vision, Image Formation and Acquisition, Digital Image Representation, Sampling and Quantization, Color Spaces: RGB, HSV, LAB, Histogram Processing and Histogram Equalization, Image Enhancement Techniques, Edge Detection: Sobel, Prewitt, Canny, Morphological Operations

Exemplars /Case Studies	**Automatic Defect Detection in Manufacturing** • Image acquisition from conveyor belts • Image enhancement and edge detection for identifying defective products

Unit No: II	Image Segmentation and Feature Extraction	9 Hours

Image Segmentation Fundamentals, Thresholding Methods, Region-Based Segmentation, Edge-Based Segmentation, Watershed Segmentation, Connected Component Analysis, Contour Detection and Shape Analysis, Hough Transform: Line Detection, Circle Detection, Feature Detection and Description: SIFT, SURF, ORB, Feature Matching Techniques		
Exemplars/Case Studies	**Automatic Number Plate Detection System** • Vehicle image segmentation • Character boundary extraction • Line and shape detection using Hough Transform	
Unit No: III	**Feature Detection and Matching**	**9 Hours**
Object Recognition Fundamentals, Template Matching, Face Detection and Recognition, Haar Cascade Classifiers, Scene Understanding, Camera Models, Perspective Projection, Stereo Vision, Depth Estimation, Introduction to 3D Reconstruction, Applications of 3D Vision in Robotics, AR and Autonomous Systems.		
Exemplars/Case Studies	**Depth Estimation for Autonomous Vehicles** • Stereo camera setup • Distance measurement of nearby objects • Lane and obstacle perception	
Unit No: IV	**Motion Analysis, Object Detection and Tracking**	**9 Hours**
Motion Detection Techniques, Background Subtraction, Optical Flow, Object Tracking Fundamentals, Multi-Object Tracking, Human Activity Recognition, Object Detection Frameworks: R-CNN, Fast R-CNN, Faster R-CNN, YOLO, Video Analytics Fundamentals, Intelligent Surveillance Systems.		
Exemplars/Case Studies	**Smart Traffic Monitoring System** • Vehicle detection using YOLO • Traffic density estimation • Tracking and counting moving vehicles	
Unit No: V	**Advanced AI-Based Computer Vision**	**9 Hours**
Semantic Segmentation, Instance Segmentation, Vision Transformers (ViT), Self-Supervised Learning for Vision, Vision-Language Models, Image Captioning, Visual Question Answering (VQA), Generative AI for Images, Diffusion Models (Concepts), Applications: Autonomous Vehicles, Medical Image Analysis, Smart Agriculture, Industrial Inspection, Ethical and Privacy Issues in Computer Vision		
Exemplars/Case Studies	**AI-Based Medical Image Diagnosis** • Tumor detection from MRI/CT images • Segmentation of affected regions • Explainable AI for clinical decision support	
Learning Resources		

Textbooks:

1. Simon J. D. Prince, "Computer Vision: Models, Learning, and Inference", Cambridge University Press, 2012
2. Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", 3rdEdition, Pearson, ISBN: 978-81- 317-2695-2

Reference Books:

1. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer, 2nd Edition, 2022.
2. Mark Nixon and Alberto S. Aguado, Feature Extraction and Image Processing for Computer Vision, 4th Edition, Academic Press, 2020.
3. Jan Erik Solem, Programming Computer Vision with Python: Tools and Algorithms for Analyzing Images, O'Reilly Media, 2012.
4. Joe Minichino and Joseph Howse, Learning OpenCV 4 Computer Vision with Python, Packt Publishing, 2020.
5. Sudha Challa, Mark Morelande, Darko Musicki, and Robin Evans, Fundamentals of Object Tracking, Cambridge University Press, 2011.
6. Rajalingappaa Shanmugamani, Deep Learning for Computer Vision: Expert Techniques to Train Advanced Neural Networks Using TensorFlow and Keras, 2nd Edition, Packt Publishing, 2022.

e-Books:

1. **https://viso.ai/deep-learning/object-tracking/**
2. **https://www.pearson.com/en-us/subject-catalog/p/digital-image-processing/P200000003224/9780137848560**
3. **https://docs.pytorch.org/tutorials/**
   https://www.tensorflow.org/tutorials/

Links to online SWAYAM/NPTEL Courses:

1. https://onlinecourses.nptel.ac.in/e-learning/preview/noc21_cs93
2. https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_ee31
3. https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs158
4. https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs104

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PCC-352-AIM
Course Name: Software Engineering with DevOps

Teaching Scheme	Credits	Examination Scheme
Theory: 2 Hrs./Wk.	2	CCE - 30 Marks ESE - 70 Marks

Prerequisite Courses, if any: Object Oriented Programming

Companion Course, if any:

Course Objectives:

1. To learn the principles of Software Engineering.
2. To learn and understand methods of capturing, specifying, visualizing and analyzing software requirements.
3. To know design principles for software project development.
4. To use Git and GitHub for source code management and team collaboration.
5. To understand cloud computing platforms such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform.

Course Outcomes:

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

CO1: Identify various software application domains and classify software applications.
CO2: Analyze software requirements by applying various modeling techniques.
CO3: Translate the requirement models into design models.
CO4: Explain the fundamentals of DevOps, and the role of collaboration between development and operations teams.
CO5: Create, manage, and deploy Docker images and container for developing scalable and portable applications.

Course Contents

Unit No: I	Introduction to Software Engineering	(06 Hours)

Software Engineering Fundamentals: Nature of Software, Software Engineering Practice, Software Process, Software Myths. Process Models: A Generic Process Model, Linear Sequential Development Model, Iterative Development Model, The incremental Development Model. Agile software development: Agile manifesto, agility principles, Agile methods, myth of planned development, Introduction to Extreme programming and Scrum. Agile Practices: test driven development, pair programming, continuous integration in DevOps, Refactoring.

Exemplars /Case Studies	**Case Study 1:** Online Hospital Management System (HMS) : A multi-specialty hospital wants to automate patient registration, appointment booking, billing, pharmacy management, and medical records. **Case Study 2:** E-Commerce Shopping Platform: A startup wants an online shopping platform like Amazon where users can browse products, place orders, make payments, and track deliveries.

Unit No: II	Requirements Engineering & Analysis	(06 Hours)

Requirements Engineering: User and system requirements, Functional and non-functional requirements, requirements engineering (elicitation, specification, validation, negotiation) prioritizing requirements (Kano diagram), requirement traceability matrix (RTM). **Software Requirements Specification (SRS):** software requirements Specification document, structure of SRS, writing a SRS, structured SRS for online shopping, **Requirements Analysis**: Analysis Model, data modeling, scenario-based modeling, class-based modeling, Flow oriented modeling, behavioral modeling-Introduction to UML diagrams.		
Exemplars/ Case Studies	1. College Library Management System 2. Smart Waste Management System Using IoT and AI	
Unit No: III	**Design Engineering**	**(06 Hours)**
Design Engineering Fundamentals: Design Process & quality, Design Concepts, Design Model, Pattern-based Software Design. **Architectural Design**: Design Decisions, Views, Patterns, Application Architectures, **Component level Design**: component, designing class-based components, conducting component-level design, **User Interface Design**: The golden rules, Interface Design steps & Analysis, Design Evaluation.		
Exemplars/Case Studies	1. AI-Powered Smart Traffic Management System. 2. Smart Farming & Crop Prediction System.	
Unit No: IV	**Introduction to DevOps Fundamentals**	**(06 Hours)**
Introduction to DevOps: DevOps culture and lifecycle, Continuous Integration (CI), Continuous Delivery/Deployment (CD), DevOps pipelines and automation. Version **Control System**: What is Git & GitHub? , Branching and Merging, Jenkins for CI/CD, Source Code Management.		
Exemplars/Case Studies	**1.** Smart E-Learning Platform an EdTech company wants to continuously deliver new features such as AI quizzes, online exams, and video lectures without affecting existing users. **2.** Smart Farming Advisory System Farmers need real-time weather recommendations and crop advisory services.	
Unit No: V	Containerization using Cloud and Monitoring Security	**(06 Hours)**
Containerization: Docker fundamentals, Docker images and containers, Kubernetes basics, Cloud platforms (AWS, Azure, GCP), Infrastructure as Code (Terraform, Ansible) **Monitoring and Security :** DevSecOps concepts, Monitoring tools (Prometheus, Grafana), Logging and alerting, Security in CI/CD pipelines, Reliability and performance monitoring.		
Exemplars/Case Studies	1. Cloud-Native E-Commerce Platform. 2. Smart Traffic Management System.	
Learning Resources		
Textbooks: 1. Roger Pressman, "Software Engineering: A Practitioner's Approach", McGraw Hill, ISBN 0-07- 337597-7 2. Gene Kim, Jez Humble, Patrick Debois, John Willis: "The DevOps Handbook: How to Create		

World-Class Agility, Reliability, and Security in Technology Organizations", IT Revolution Press, IT Revolution Press.

3. Bob Hughes, Mike Cotterell and Rajib Mall: Software Project Management – Fifth Edition, Tata McGraw Hill, New Delhi. - (for Unit 1, 3, 5)

Reference Books:

1. Pankaj Jalote, Software Engineering: A Precise Approach, Wiley India, ISBN: 9788-1265-2311-5
2. JIM Arlow, Ila Neustadt, UML 2 and the Unified Process, Pearson, Second Edition, ISBN:9788131700549 Tom Pender, UML 2 Bible, Wiley India, ISBN: 9788126504527. (For Unit 2)
3. James P Lewis, "Project Planning, Scheduling & Control", McGraw Hill, 5th Edition, 2011.

Links to online SWAYAM/NPTEL Courses:

1. Software Engineering (NPTEL-SWAYAM) by Prof. Rajib Mall, IIT Kharagpur

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PCC-353 AIM
Course Name: Computer Vision Lab 1

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

Course Objectives:

1. Implement image processing and enhancement techniques using OpenCV.
2. Apply segmentation and feature extraction methods for image analysis.
3. Develop object recognition and 3D vision applications.
4. Implement motion analysis, object detection, and tracking systems.
5. Build AI-based computer vision applications using state-of-the-art models.

Course Outcomes:

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

CO1: Apply image acquisition, preprocessing, enhancement, and transformation techniques for visual data analysis.

CO2: Implement image segmentation, feature extraction, and shape analysis techniques for object identification.

CO3: Develop computer vision applications for object recognition, face recognition, and three-dimensional scene understanding.

CO4: Apply motion analysis, object detection, tracking, and video analytics techniques for dynamic scene interpretation.

CO5: Design AI-enabled computer vision solutions using deep learning models for classification, detection, and segmentation tasks.

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.

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
2. parameters such as timely conduction of practical assignment, methodology adopted for

implementation of practical assignment, timely submission of assignment in the form of
3. handwritten write-up along with results of implemented assignment, attendance etc.
4. 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.
5. 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

Virtual Laboratory

- https://cse19-iiith.vlabs.ac.in/List%20of%20experiments.html

1. **Image Acquisition and Basic Image Operations ---CO1**
   Write a program in Python to perform basic image processing operations including image loading, display, color conversion, resizing, cropping, and geometric transformations. --
   Library: OpenCV, NumPy **Dataset/Input:** User-selected digital images
2. **Histogram Processing and Image Enhancement—CO1**
   Write a program in Python to analyze image histograms and perform contrast enhancement using histogram equalization and brightness adjustment techniques -
   Library: OpenCV, Matplotlib **Dataset/Input:** Grayscale and color images
3. **Edge Detection and Morphological Operations** –CO1, CO2
   Write a program in Python to detect image edges using Sobel, Prewitt, and Canny operators and perform morphological operations such as erosion and dilation.
   Library: OpenCV, NumPy **Dataset/Input:** Real-world images
4. **Image Segmentation** –CO2
   Write a program in Python to perform image segmentation using thresholding, adaptive thresholding, and watershed segmentation techniques.
   Library: OpenCV, Scikit-image **Dataset/Input:** Natural scene images
5. **Feature Detection and Matching---CO2, CO3**
   Write a program in Python to extract image features using ORB/SIFT and perform feature matching between two related images.
   Library: OpenCV **Dataset/Input:** Multiple views of the same object
6. **Shape Detection using Hough Transform---CO2,CO3**
   Write a program in Python to detect straight lines and circular objects using Hough Line

	Transform and Hough Circle Transform. **Library:** OpenCV, **Dataset/Input:** Road images, traffic signs, industrial images
7.	**Face Detection and Recognition---CO3** Write a program in Python using OpenCV and pre-trained face recognition models to detect and recognize human faces from images or video streams. – **Library:** OpenCV, Face Recognition Library **Dataset/Input:** Face image datase
8.	**Stereo Vision and Depth Estimation—CO3** Write a program in Python to generate disparity maps and estimate object depth from stereo image pairs. **Library:** OpenCV, NumPy **Dataset/Input:** Stereo image dataset
9.	**Motion Detection and Background Subtraction—CO4** Write a program in Python to detect moving objects in videos using frame differencing and background subtraction techniques. **Library:** OpenCV **Dataset/Input:** Video sequences
10.	**Object Detection using YOLO---CO4,Co5** Develop a real-time object detection system using Python and YOLO to identify and localize multiple objects in images and videos. **Library:** OpenCV, Ultralights **Dataset/Input:** COCO Dataset or custom dataset
11.	**Multi-Object Tracking---CO4,CO5** Develop a Python application using YOLO and tracking algorithms to track multiple moving objects in a video sequence. **Library:** OpenCV, Ultralytics YOLO **Dataset/Input:** Traffic surveillance videos
12.	**Semantic Segmentation using Deep Learning—CO2,CO5** Develop a semantic segmentation system using a pre-trained deep learning model such as U-Net or DeepLabV3 for pixel-level image classification. – CO2, CO5 **Library:** TensorFlow/Keras or PyTorch **Dataset/Input:** Medical images or Cityscapes dataset
13.	**Vision Transformer-Based Image Classification—CO5** **D**evelop an image classification system using a pre-trained Vision Transformer (ViT) model and compare its performance with a CNN-based classifier. **Library:** PyTorch, Hugging Face Transformers, **Dataset/Input:** CIFAR-10 or ImageNet subset

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PCC-354-AIM
Course Name: Software Engineering with DevOps Lab 2

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

Course Objectives:

1. To explore the need of DevOps and DevOps life cycle as software engineering practice.
2. To apply the concept of Continuous Integration Continuous Delivery (CICD).
3. To learn the concept of Containerization and Container Orchestration.
4. To learn the basics of Infrastructure as code.

Course Outcomes:

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

CO1: Make use of Version Control system to efficiently track changes, manage code versions, and collaborate on software development.
CO2: Apply Continuous Integration (CI) and Continuous Deployment (CD) to automate code integration, deployment.
CO3: Utilize containerization and Container orchestration tools to package, deploy, and manage applications.
CO4: Apply Configuration Management, Infrastructure as Code (IaC), and Cloud Services for automated provisioning, deployment, and management of IT infrastructure.

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. Experiments shall be conducted using appropriate programming languages, software platforms, hardware configurations, and relevant tools. The selected technologies should support the achievement of the prescribed learning objectives and outcomes.

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 the faculty concerned and head of the department.
5. All the assignments 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
2. parameters such as timely conduction of practical assignment, methodology adopted for implementation of practical assignment, timely submission of assignment in the form of

3. handwritten write-up along with results of implemented assignment, attendance etc.
4. 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.
5. 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 them 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. Develop Version Control using Git and Github -- CO1
   a. Use git commands to understand the version control system.
   b. Branching and Merging
2. Study Jenkins Architecture and Plugins with Jenkins Installation and Setup. -- CO1, CO2, CO3
3. Integrate Jenkins with a Git repository. Configure Jenkins to pull code from GitHub. --CO1, CO2
4. Automating Build Using Jenkins: Set up a Build Job that compiles a Java project and configure triggers to run the build on every commit. - CO1, CO2
5. Study and Explore the Docker architecture, Docker commands, and Docker file . -CO1, CO2
6. Containerize a simple Python Application in Docker. –CO4
7. Study the concept of Container orchestration using Kubernetes and explore Kubernetes architecture.
8. Explore Kubernetes objects (Pod, Deployment, Service, Namespace, Config Map, Secret, Persistent Volume) and Kubernetes services (Cluster IP, Node Port, Load Balancer, External Name)
9. Install Mini Kube. Create and deploy a simple **Nginx** pod in a Kubernetes cluster.
10. Explore the different services of AWS. (EC2, S3, AWS Lambda, RDS, ELB, ECS)
11. Create EC2 instance in AWS.

Reference Books:

1	J Pro Devops with Google Cloud Platform_ With Docker, Jenkins, and Kubernetes
2	Practical guide to testing in DevOps.
3	DevOps for Developers: Michael Hüttermann
4	Practical DevOps: Joakim Veronal

E-resources:

1. **https://www.edureka.co/blog/devops-tutorial**
2. **https://www.guru99.com/devops-tutorial.html**

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PEC-361A-AIM
Course Name: Quantum Computing

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

Prerequisite Courses, if any: College level Physics (recommended but not essential), Basic linear algebra and probability theory

Companion Course, if any: NA

Course Objectives:

1. Mathematically represent single-qubit states using Dirac notation
2. Use tensor products and density matrices to analyze multi-qubit systems
3. Construct reversible quantum logic circuits using universal gates and calculate the statistical expectation values of measurements
4. Study important quantum algorithms like Deutsch-Jozsa algorithm, Grover's search algorithm, Shor's algorithm
5. Study some simple quantum cryptographic protocols and quantum error correcting codes

Course Outcomes:

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

CO1: Formulate mathematical models of single-qubit states and visualize quantum phenomena using the Bloch sphere.

CO2: Analyse multi-qubit systems using density matrices to distinguish states and verify quantum entanglement.)

CO3: Design reversible quantum logic circuits and compute the statistical expectation values of measurements.

CO4: Evaluate the mathematical execution and theoretical speedup of foundational quantum algorithms.

CO5: Develop understanding towards simple quantum cryptographic protocols and error correcting codes.

Course Contents

Unit No: I	Introduction to Quantum Computation	9 Hours

Classical vs. Quantum mechanics: The double-slit experiment and wave-particle duality. The concepts of superposition, interference, and observation (the measurement problem). A conceptual introduction to the Schrödinger equation.

The Quantum Computing Paradigm: Classical bits vs. qubits, Introduction to Dirac notation (bra-ket notation), The standard basis (computational basis) states, Representing qubits: Superposition, probability amplitudes, and the Born rule. Visualizing single qubits using the Bloch sphere.

Exemplars /Case Studies	Case Study 1: Based on the Mathematics of interference (probability vs. probability amplitudes) Case Study 2: Based on the geometry of the Bloch sphere

Unit No: II	**Qubits**	**9 Hours**
Single-Qubit Gates: The Pauli matrices (X, Y, Z), the Hadamard (H) gate, and phase gates (S, T). Representing two-qubit and multi-qubit states using tensor products. Bipartite systems and quantum entanglement. EPR pairs and Bell states. Density matrices, distinguishing between pure states and mixed states.		
Exemplars/Case Studies	Case Study 1: Based on basis transformation Case Study 2: Based on the algebra of entanglement and density matrices	
Unit No: III	**Measurement and Quantum Circuits**	**9 Hours**
The Measurement Problem: Observables, expectation values, and wave function collapse. Multi-Qubit Gates: Controlled operations, CNOT and Toffoli (CCNOT) gates. Quantum Circuits: Reading and designing quantum circuit diagrams, and the concept of reversibility in computation, Universal Gates, The No-Cloning Theorem		
Exemplars/Case Studies	Case Study 1: Based on the Mathematics of measurement (observables and expectation values) Case Study 2: Based on CNOT gates and the no-cloning theorem	
Unit No: IV	**Quantum Algorithms**	**10 Hours**
Deutsch's Algorithm and the Deutsch-Jozsa Algorithm. Grover's Algorithm: Geometry of amplification and reflections, Grover iteration and search bounds, Applications to NP search problems, Variants of Grover's algorithm. Quantum Fourier Transform (QFT): definition and properties, efficient circuit for QFT, period finding. Integer factorization using QFT (Shor's algorithm), Order-finding problem, Discrete logarithms, Impact on RSA and cryptographic systems		
Exemplars/Case Studies	Case Study 1: The geometry of amplitude amplification (Grover's algorithm) Case Study 2: Shor's algorithm and period finding	
Unit No: V	**Quantum Information & Error Correction**	**8 Hours**
Quantum Information Theory: Classical Shannon entropy versus quantum Von Neumann entropy. Quantum Cryptography: The BB84 Quantum Key Distribution (QKD) protocol. Quantum Teleportation: Moving quantum states using classical channels and shared entanglement. Decoherence: The physics of environmental noise and why quantum states degrade. Quantum Error Correction: Bit-flip codes, phase-flip codes, and Shor's 9-qubit code.		
Exemplars/Case Studies	Case Study 1: The BB84 Protocol Case Study 2: Hiding Information in Entanglement (Shor's 9-Qubit Code)	
Learning Resources		
Textbooks: 1. M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, 10th Anniversary ed. Cambridge, U.K.: Cambridge Univ. Press, 2010.		

2. P. Kaye, R. Laflamme, and M. Mosca, An Introduction to Quantum Computing. Oxford, U.K.: Oxford Univ. Press, 2007.

Reference Books:

1. Y. Kitaev, A. Shen, and M. N. Vyalyi, Classical and Quantum Computation. Providence, RI, USA: American Mathematical Society, 2002.
2. G. Strang, Introduction to Linear Algebra, 6th ed. Wellesley, MA, USA: Wellesley-Cambridge Press, 2023.
3. N. S. Yanofsky and M. A. Mannucci, Quantum Computing for Computer Scientists. Cambridge, U.K.: Cambridge Univ. Press, 2008.

Links to online SWAYAM/NPTEL Courses:

1. https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs89 [accessed on 3 June 26]

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PEC-361B-AIM
Course Name: Blockchain Technology

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

Prerequisite Courses, if any:

6. Data Structures and Algorithms
7. Computer Networks
8. Database Management Systems
9. Cryptography and Network Security (basic concepts)

Companion Course, if any: NIL

Course Objectives:

1. To develop an understanding of blockchain architecture, distributed ledger technologies, and their applications.
2. To analyze cryptographic techniques and consensus mechanisms used in blockchain systems.
3. To design smart contracts and decentralized applications using blockchain platforms.
4. To evaluate enterprise blockchain frameworks and their integration with emerging technologies.
5. To assess the applicability of blockchain solutions for real-world problems across various domains.

Course Outcomes:

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

CO1: Interpret the architecture, components, and operational principles of blockchain systems and distributed ledgers.

CO2: Analyze cryptographic techniques, consensus algorithms, and security mechanisms employed in blockchain networks.

CO3: Implement smart contracts using contemporary blockchain development platforms and frameworks.

CO4: Evaluate decentralized and enterprise blockchain solutions for diverse application domains. (L5)

CO5: Design blockchain-based solutions to address real-world challenges considering scalability, security, and sustainability requirements.

Course Contents

Unit No: I	Introduction to Blockchain Technology	8 Hours

Distributed Systems and Distributed Ledger Technology (DLT), Challenges in Distributed Systems, Trust Models and Decentralization, Evolution of Blockchain Technology, Blockchain Generations (Blockchain 1.0, 2.0, 3.0), Blockchain Architecture and Components, Transactions, Blocks and Block Structure, Hash Pointers and Merkle Trees, Public-Key Infrastructure in Blockchain, Public, Private and Consortium Blockchains, Permissioned and Permissionless Networks, Blockchain Ecosystem and Stakeholders, Cryptocurrency Fundamentals and Bitcoin Overview, Benefits, Limitations, and Emerging Applications of Blockchain Technology.

Exemplars/Case Studies	Case Study 1: Ethereum Ecosystem: Analysis of Blockchain Architecture, Smart Contracts, Decentralized Applications (DApps), Consensus Evolution, and Real-World Web3

	Applications. Case Study 2: Interactive Exploration of Blockchain Fundamentals using Anders Brownworth Blockchain Demo: Visualization of Hashing, Block Formation, Blockchain Linking, Distributed Ledger Concepts, Transaction Validation, and Immutability in Blockchain Networks.	
Unit No: II	**Cryptographic Foundations and Consensus Mechanisms**	**8 Hours**
Cryptographic Foundations of Blockchain, Cryptographic Hash Functions and Properties, Digital Signatures, Public Key Cryptography, Elliptic Curve Cryptography (ECC), Hash Pointers and Merkle Trees, Cryptographic Security Requirements, Consensus in Distributed Systems, Byzantine Generals Problem, Byzantine Fault Tolerance (BFT), Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS), Practical Byzantine Fault Tolerance (PBFT), Proof of Authority (PoA), Security Threats and Attack Vectors, Double Spending, Sybil Attack, 51% Attack, Blockchain Security Challenges.		
Exemplars/Case Studies	Case Study 1: Comparative Analysis of Consensus Mechanisms in Bitcoin and Ethereum: Evaluation of Proof of Work (PoW) and Proof of Stake (PoS) with respect to security, scalability, decentralization, and energy efficiency. Case Study 2: Analysis of Blockchain Security Incidents: Investigation of Double-Spending Attacks, Sybil Attacks, and 51% Attacks, along with mitigation strategies adopted in modern blockchain networks.	
Unit No: III	**Smart Contracts and Ethereum Ecosystem**	**8 Hours**
Introduction to Ethereum and Ethereum Architecture, Ethereum Accounts and Wallets, Ethereum Virtual Machine (EVM), Ether and Gas Mechanism, Smart Contracts: Concepts and Life Cycle, Solidity Programming Fundamentals, Data Types, Functions, Modifiers, Events, Inheritance and Interfaces, Development and Deployment of Smart Contracts, Token Standards (ERC-20, ERC-721, ERC-1155), Decentralized Applications (DApps), Testing and Debugging Smart Contracts, Introduction to Web3 and Blockchain Development Tools.		
Exemplars/Case Studies	Case Study 1: Development and Analysis of an ERC-20 Token Smart Contract using Ethereum and Solidity. Case Study 2: Crypto Kitties and NFT Ecosystem: Analysis of ERC-721 Token Standard, Digital Asset Ownership, and Smart Contract-Based Marketplace Operations.	
Unit No: IV	**Blockchain and Decentralized Applications**	**7 Hours**
Enterprise Blockchain Fundamentals, Permissioned Blockchain Networks, Hyperledger Fabric Architecture, Peers, Ordering Service, Channels and Membership Service Provider (MSP), Chaincode and Smart Contracts in Hyperledger Fabric, Blockchain-as-a-Service (BaaS), Decentralized Applications (DApps) Architecture, Web3 Frameworks and APIs, Blockchain Integration with Cloud Computing, Internet of Things (IoT), Artificial Intelligence, Enterprise Use Cases and Deployment Challenges.		
Exemplars/Case Studies	Case Study 1: Hyperledger Fabric-Based Supply Chain Management System: Analysis of Permissioned Blockchain Architecture, Chaincode, and Multi-Stakeholder Collaboration. Case Study 2: TradeLens Platform: Application of Enterprise	

	Blockchain for Global Logistics, Supply Chain Visibility, and Secure Information Sharing	
Unit No: V	**Blockchain Applications, Web3 and Future Trends**	**7 Hours**
Blockchain Applications in Finance, Healthcare, Supply Chain Management, Digital Identity, E-Governance, Education and Intellectual Property Management, Decentralized Finance (DeFi), Non-Fungible Tokens (NFTs), Decentralized Autonomous Organizations (DAOs), Web3 Ecosystem and Token Economy, Blockchain Scalability Challenges, Layer-2 Solutions, Cross-Chain Interoperability, Blockchain and Artificial Intelligence, Blockchain for Sustainable Development, Regulatory and Ethical Considerations, Emerging Research Trends and Future Directions.		
Exemplars/Case Studies	Case Study 1: Decentralized Finance (DeFi) Platforms: Analysis of Lending, Borrowing, Yield Farming, and Decentralized Exchanges using Smart Contracts. Case Study 2: Blockchain and Artificial Intelligence Integration: Investigation of Trustworthy AI, Secure Data Sharing, Decentralized AI Marketplaces, and Federated Learning Applications.	

Learning Resources

Textbooks:

1. Gaur, N., Ramakrishna, V., and Singh, M. Hands-On Blockchain with Hyperledger: Building Decentralized Applications with Hyperledger Fabric and Composer, Packt Publishing.
2. Bahga, A., and Madisetti, V. Blockchain Applications: A Hands-On Approach, 1st Edition, VPT, 2017.
3. Drescher, D. Blockchain Basics: A Non-Technical Introduction in 25 Steps, 1st Edition, Apress.

Reference Books:

1. Antonopoulos, A. M. Mastering Bitcoin: Programming the Open Blockchain, 2nd Edition, O'Reilly Media.
2. Antonopoulos, A. M., & Wood, G. Mastering Ethereum: Building Smart Contracts and DApps, 1st Edition, O'Reilly Media.
3. Gaur, N., Ramakrishna, V., & Singh, M. Hands-On Blockchain with Hyperledger: Building decentralized applications with Hyperledger Fabric and Composer, 1st Edition, Packt Publishing.
4. Bashir, I. Mastering Blockchain: A Deep Dive into Distributed Ledgers, Consensus Protocols, Smart Contracts, DApps, Cryptocurrencies, Ethereum, and More, 3rd Edition, Packt Publishing, 2020.

e-Books:

1. European Blockchain Observatory & Forum. Blockchain for Beginners: A EUBOF Guide, Open-Access Digital Publication.
2. Kumar Saurabh and Ashutosh Saxena. Blockchain Technology: Concepts and Applications, Wiley Technical E-Library Edition.
3. https://ethereum.org/en/whitepaper/
4. https://www.bitcoin.com/satoshi-archive/whitepaper/
5. Hyperledger Tutorials - https://www.hyperledger.org/use/tutorials
6. Ethereum Development Resources - https://ethereum.org/en/developers

Links to online SWAYAM/NPTEL Courses:

1. Blockchain and its Applications (IIT Kharagpur)
2. Blockchain Architecture, Design and Use Cases (IIT Kharagpur)

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PEC-361C-AIM
Course Name: High Performance Computing

Teaching Scheme	Credits	Examination Scheme
Theory: 03 H/W		CCE – 30 Marks ESE- 70 Marks

Prerequisite Courses, if any: Data Structures & Algorithms (PCC-201-AIM), Object Oriented Programming (PCC-202- AIM), Operating System (PCC-301 -AIM).

Companion Course, if any:

Course Objectives:

1. To introduce the fundamental concepts of parallel computing, parallel architectures, and programming paradigms used in modern computing systems.
2. To understand the fundamental concepts and models of parallel algorithm design.
3. To familiarize students with MPI based communication operations and parallel programming concepts.
4. To develop skills for performance analysis of parallel algorithms.
5. To equip students with the fundamentals of CUDA programming and GPU acceleration for HPC applications.

Course Outcomes:

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

CO1: Understand the fundamentals of parallel computing, parallel architectures, and programming paradigms.

CO2: Apply different Parallel programming paradigm and Decomposition Techniques.

CO3: Illustrate data communication operations on various parallel architecture

CO4: Analyze the performance of parallel systems.

CO5: Apply GPU computing and CUDA programming techniques for High-Performance Computing applications.

Course Contents

Unit No: I	Fundamentals of Parallel Computing and Parallel Architectures	09 Hours

Introduction: Motivating Parallelism, Scope of Parallel Computing. **Dichotomy of Parallel Computing Platforms:** Implicit Parallelism, (Flynn's Classifications, Distributed Memory Architecture, Shared Memory Architecture, Hybrid Architecture), Physical Organization of Parallel Platforms, Communication Costs in Parallel Machines. Levels of parallelism. Interconnection Networks. **Architecture:** N-wide superscalar architectures, multi-core, multi-threaded.

Exemplars /Case Studies	**Case Study 1: Evolution** of Multi-core and Many-core Systems **Case Study 2:** Evolution of Many-core Systems

Unit No: II	Parallel Algorithm Design	09 Hours

Principles of Parallel Algorithm Design: Preliminaries, Decomposition Techniques, Characteristics of Tasks and Interactions, Mapping Techniques for Load Balancing, Methods for Containing Interaction Overheads, Parallel Algorithm Models: Data, Task, Work Pool and Master Slave model. Complexities: Sequential and Parallel Computational Complexity, Anomalies in Parallel Algorithms.

Exemplars/Case Studies	Case Study 1: Parallel Sensor Data Aggregation System Case Study 2: Monte Carlo Simulation (π Estimation) Case Study 3: Parallel Programming Case studies https://www.cs.cmu.edu/afs/cs/academic/class/15418-s21/www/lectures/09_casestudies.pdf	
Unit No: III	**Parallel Programming and Communication**	**09 Hours**
Basic Communication Operations: One-to-All Broadcast and All-to-One Reduction, All-to-All Broadcast and Reduction, All-Reduce and Prefix-Sum Operations, **Parallel Programming Paradigm**: Shared memory programming, message passing programming model, MPI programming Environment, **Communication using MPI:** Scatter, Gather, Broadcast, Blocking and non -blocking MPI, All-to-All Personalized Communication, Circular Shift, Improving the speed of some communication operations.		
Exemplars/Case Studies	Matrix Multiplication using MPI Parallel K-Means Clustering using MPI	
Unit No: IV	**Analytical Modeling of Parallel Programs**	**09 Hours**
Sources of Overhead in Parallel Programs, **Performance Measures and Analysis:** Amdahl's and Gustafson's Laws, Speedup Factor and Efficiency, Cost and Utilization, Execution Rate and Redundancy, The Effect of Granularity on Performance, Scalability of Parallel Systems, Minimum Execution Time and Minimum Cost, Optimal Execution Time, Asymptotic Analysis of Parallel Programs. **Matrix Computation:** Matrix-Vector Multiplication, Matrix-Matrix Multiplication		
Exemplars/Case Studies	Performance analysis of Parallel Sorting Algorithms. (Bubble sort, Quick sort and Merge sort) Performance analysis of Parallel Searching Algorithms. (BFS and DFS)	
Unit No: V	**CUDA Architecture and High-Performance Computing Applications**	**09 Hours**
Introduction to GPU Architecture, CPU vs GPU, Scope of Parallel Computing, Introduction to CUDA C and CUDA Programming Model, CUDA Kernels, Thread Hierarchy, Error Handling, CUDA Memory Model and Memory Management, Communication, Synchronization and CUDA-C Programming, Kubernetes, GPU Applications and Parallel Computing for AI/ML.		
Exemplars/Case Studies	Vector Addition Using CUDA Matrix Multiplication using CUDA C	
Learning Resources		

Textbooks:

1. Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar, "Introduction to Parallel
2. Computing", 2nd edition, Addison-Wesley, 2003, ISBN: 0-201-64865-2
3. Seyed H. Roosta, “Parallel Processing and Parallel Algorithms Theory and Computation‖”, Springer-Verlag 2000, ISBN 978-1-4612-7048-5 ISBN 978-1-4612-1220-1
4. John Cheng, Max Grossman, and Ty McKercher, “Professional CUDA C Programming”,
5. John Wiley & Sons, Inc., ISBN: 978-1-118-73932-7

Reference Books:

1. Kai Hwang,, "Scalable Parallel Computing", McGraw Hill 1998.
2. George S. Almasi and Alan Gottlieb, "Highly Parallel Computing", The Benjamin and Cummings Pub. Co., Inc Jason sanders, Edward Kandrot, “CUDA by Example”, Addison-Wesley, ISBN-13: 978-0-13-138768
3. Pacheco, Peter S., “An Introduction to Parallel Programming”, Morgan Kaufmann Publishers ISBN 978-0-12-374260-5
4. Rieffel WH.EG, Polak, “Quantum Computing: A gentle introduction”, MIT Press, 2011 ,I SBN 978-0-262-01506-6.
5. Ajay D. Kshemkalyani , Mukesh Singhal, “Distributed Computing: Principles, Algorithms, and Systems”, Cambridge March 2011, ISBN: 9780521189842

e-Books:

1. **http://prdrklaina.weebly.com/uploads/5/7/7/3/5773421/introduction_to_high_performance_computing_for_scientists_and_engineers.pdf**
2. **https://www.vssut.ac.in/lecture_notes/lecture1428643084.pdf**

Links to online SWAYAM/NPTEL Courses:

1. https://nptel.ac.in/courses/106108055
2. https://www.digimat.in/nptel/courses/video/106104120/L01.html

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PEC-361D-AIM
Course Name: Data Science & Big Data Analytics

Teaching Scheme	Credits	Examination Scheme
Theory: 3 Hrs./Wk.	3	CCE - 30 Marks ESE - 70 Marks

Prerequisite Courses, if any:
PCC-252- AIM - Foundation of Machine Learning
PCC-255- AIM - Foundation of Machine Learning Lab
PCC-302 -AIM PCC 2 Deep Learning

Companion Course, if any:

Course Objectives:
1. To understand the need of Data Science and Big Data to handle enormous amounts of data.
2. To understand computational statistics in Data Science
3. To learn Data Analytics using Python programming
4. To understand and apply the Analytical concept of Big data using Python.
5. To understand the application and impact of Big Data.

Course Outcomes:
On completion of the course, students will be able to–
CO1: Analyze needs and challenges for Data Science Big Data Analytics
CO2: Understand, apply and analyze needs, challenges and techniques for big data visualization.
CO3: Apply the lifecycle of Big Data analytics to real world problems
CO4: Design and implement Big Databases using the Hadoop ecosystem
CO5: Evaluate the ethical, privacy, security, and societal implications associated with data science and big data applications.

Course Contents

Unit No: I	Foundations of Data Science and Big Data	8 Hours

Introduction to Data Science and Big Data, Real-World Applications of Big Data, Roles and Responsibilities of a Data Scientist. Data Explosion and the Digital Universe: data generated from IoT devices, social networks, mobile applications, sensors, enterprise systems. Shared-Everything and Shared-Nothing Architectures.

Data Volume: Managing Massive Datasets ranging from terabytes to petabytes and exabytes, necessitating distributed storage solutions such as Hadoop Distributed File System (HDFS).
Data Velocity: Real-Time Data Generation and Processing, e.g. stock market analysis, online fraud detection, and autonomous systems.

Big Data Infrastructure Components: distributed storage systems, cluster computing platforms, NoSQL databases, stream processing engines, cloud computing services, and data integration tools to support scalable analytics.

Challenges Associated with Big Data: data quality assurance, security and privacy concerns, heterogeneous data integration, scalability limitations, fault tolerance, high computational costs, and regulatory compliance requirements.

Exemplars /Case Studies	**Case Study 1: Managing** Big Data Challenges in an Online Retail Platform Define Data Science and Big Data in the context of Shop Smart. Identify examples of the 3Vs of Big Data: Volume, Velocity, Variety Explain why the company is experiencing Data Explosion. Discuss the limitations of using a traditional Data Warehouse for this scenario. Suggest a suitable Big Data Processing Architecture (e.g., Hadoop ecosystem or Spark-based architecture) and justify your choice. Compare Shared-Everything and Shared-Nothing Architectures and recommend the appropriate architecture for Shop Smart. Identify at least three challenges the company may face while implementing Big Data solutions. **Case Study 2:** Leveraging Data Science and Big Data for Urban Traffic Optimization Explain how this scenario represents a Big Data application. Identify the characteristics of Big Data observed in this case using the 3Vs framework. Recommend appropriate Big Data Infrastructure for handling streaming data and explain your reasoning. Describe how Data Warehouse Re-engineering could support historical traffic analysis. Suggest suitable Big Data Learning Approaches (batch learning vs. online learning) and justify your choice. Discuss the challenges associated with implementing such a system, including: Data integration, Data quality, Scalability, Privacy concerns.

Unit No: II	Big Data Processing Architectures and Next-Generation Data Warehouses	**8 Hours**

Big Data Processing Architectures: Distributed architectures such as batch processing frameworks (MapReduce), in-memory processing systems (Apache Spark), and stream processing platforms (Apache Flink and Apache Storm) to handle diverse analytical workloads efficiently.

Data Warehouse Concepts and Evolution: Application in Decision Support Systems (DSS), Business Analytics, Data Analytics, Management Information Systems (MIS), Enterprise Intelligence, Performance Management, Strategic Information Systems etc.

Re-Engineering the Traditional Data Warehouse: Incorporating data lakes, cloud-native architectures, real-time data ingestion mechanisms, and support for semi-structured and unstructured data formats.

Big Data Learning Approaches and the AI Hierarchy: Descriptive, Diagnostic, Predictive, and Prescriptive learning approaches.

Exemplars/Case Studies	**Case Study 1:** Building an Integrated Healthcare Analytics Platform for Proactive Patient Care. Why are stream processing platforms important in healthcare environments? What advantages do cloud-native architecture provide to healthcare organizations? Differentiate between predictive and prescriptive analytics using examples from this case. How can unstructured healthcare data be managed in a re-engineered data warehouse environment? Which components of this system support enterprise intelligence and performance management? **Case Study 2:** Predictive Maintenance in Manufacturing Plants

	Compare the roles of batch processing and stream processing in predictive maintenance. How does cloud-native architecture enhance manufacturing analytics capabilities? Why is prescriptive analytics valuable in maintenance decision-making? What benefits do data lakes provide over traditional data warehouses in IIoT environments?	
Unit No: III	Distributed Data Processing Frameworks	**8 Hours**
Big Data distributed storage, parallel processing, and analytics frameworks like Hadoop, Spark, Hive, Pig, and NoSQL databases. Architecture and working principles of the Google File System (GFS) and its influence on modern distributed storage systems. Hadoop architecture, including its core components and the role of distributed computing in Big Data processing. Functioning of Hadoop Distributed File System (HDFS), role of Name Node, Secondary Name Node, and Data Nodes. Hadoop shell commands file management operations such as creating directories, uploading files, retrieving files, and monitoring HDFS. Hadoop ensures fault tolerance and data reliability. Hadoop MapReduce programming paradigm, including the concepts of map tasks, reduce tasks, jobs, and task scheduling mechanisms. Hadoop cluster environment SSH access and modify essential Hadoop configuration files. NoSQL databases and textual ETL processing techniques for managing and transforming structured, semi-structured, and unstructured data. Hadoop ecosystem components such as MapReduce, Pig, and Hive, and evaluate their roles in data processing, querying, and analytical applications.		
Exemplars/Case Studies	**Case Study 1:** Real-Time Fraud Detection in Banking. A multinational bank faces increasing online fraud attempts and needs a Big Data distributed storage and parallel processing solution to analyze millions of transactions per second. The system detects fraudulent activity in real time, reducing financial losses by a certain percentage and improving customer trust through proactive alerts. Design distributed storage systems to solve large-scale problems in banking. **Case Study 2:** Personalized Content Recommendation in Media Streaming: A global streaming platform wants to deliver personalized recommendations to millions of users by leveraging Big Data analytics frameworks. The platform achieves a 25% increase in user engagement, reduces churn, and enhances customer satisfaction by delivering highly personalized recommendations. These case studies show how distributed storage, Hadoop ecosystem tools, MapReduce, and NoSQL databases come together to solve large-scale problems in banking and media streaming.	
Unit No: IV	Data Analytics Lifecycle Framework	**8 Hours**
Data Analytics Lifecycle Framework: Discovery, Data Preparation, Model Planning, Model Building, Communication of Results, and Operationalization. Analytical Approaches in Data Science: Statistical Analysis, Exploratory Data Analysis, Machine Learning to extract insights and support decision-making.		

Python as a Data Analytics Tool: Python provides powerful libraries (such as Pandas, NumPy, Matplotlib, and Scikit-learn) for data manipulation, analysis, visualization, understanding patterns and trends within datasets, and predictive modeling. Data Ingestion from Multiple Sources: Python enables importing data from diverse sources including CSV, JSON, HTML, Excel files, MongoDB, MySQL, and SQLite databases, facilitating comprehensive data integration. Data cleaning: Removing duplicate records, correcting inconsistent data formats (e.g., date formats, text case), Detecting and treating outliers, Validating and correcting erroneous data entries. Handling missing values: Listwise deletion (removing records with missing values), Pairwise deletion (using available data for analysis), Replacing missing values with a constant (e.g., "Unknown", 0), Identifying missing data patterns Missing Completely at Random, Missing At Random, Missing Not at Random treatment. Data imputation: Mean/Median imputation for numerical attributes, Mode imputation for categorical attributes, k-Nearest Neighbors (KNN) imputation, Regression or Multiple Imputation techniques. Data Transformation: Normalization (Min-Max Scaling), Standardization (Z-score scaling), Encoding categorical variables (One-Hot Encoding, Label Encoding), Aggregation and feature construction (creating new derived attributes)	
Exemplars/Case Studies	**Case Study:** Predicting Employee Attrition Using HR Analytics Background : Identify key factors influencing employee attrition. Predict whether an employee is likely to leave the organization. Provide actionable recommendations to reduce turnover. IBM HR Analytics Employee Attrition Dataset Dataset URL: https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset 1. Discovery : Understand business objectives related to employee retention, Identify relevant variables such as age, job role, salary, overtime, job satisfaction, and work-life balance. Define success criteria (e.g., achieving prediction accuracy above 80%). 2. Data Preparation: Import HR data from CSV files using Pandas, explore dataset structure and variable types, integrate additional employee satisfaction survey data stored in Excel format. 3. Data Cleaning: Remove duplicate employee records, standardize text values (e.g., "Sales", "sales", "SALES"), Detect outliers in monthly income using boxplots, Validate impossible entries such as negative working years. 4. Handling Missing Values: Listwise deletion for records with

	excessive missing values, replace missing categorical values with "Unknown", Analyze missingness patterns MCAR MAR, MNAR 5. Data Imputation: Mean/Median imputation for Monthly Income, Mode imputation for Job Satisfaction category, KNN imputation for correlated numerical features such as Age and Years at Company. 6. Data Transformation: Standardization (Z-score scaling) for continuous variables, One-Hot Encoding for Department and Job Role. 7. Feature engineering: Years Since Last Promotion Ratio, Income Per Year Experience 8. Model Planning: Statistical Analysis: Correlation analysis and Chi-square tests, Exploratory Data Analysis (EDA): Attrition trends across departments, 9. Machine Learning Models: Logistic Regression, Decision Tree Classifier, Random Forest Classifier 8. Model Building: Split data into training and testing sets (80:20), Train classification models, Evaluate using Accuracy, Precision, Recall, F1-score, and ROC-AUC. 9. Communication of Results: Create visualizations using Matplotlib, Attrition by department, Feature importance plots, ROC curves for model comparison. 10. Present recommendations such as: Review overtime policies, improve career advancement opportunities, Enhance employee engagement programs. 11. Operationalization: Deploy the predictive model within the HR management system, generate monthly attrition risk reports, Monitor model performance periodically.	
Unit No: V	Big Data Technologies, Applications and Impact	**8 Hours**

Social Media Analytics: Analyze data from social media platforms to understand customer sentiments, brand perception, market trends, and user engagement patterns for informed decision-making.

Mobile Analytics: Monitor and evaluate mobile application usage data, app performance, user behavior, retention rates, and engagement metrics to improve user experience and business performance.

Autonomous Vehicles: Self-driving cars rely on real-time big data processing from sensors, cameras, and GPS. Machine learning models analyze traffic patterns, detect obstacles, and predict driver behavior, Integration of IoT data streams ensures safe navigation and adaptive route planning.

Precision Healthcare and Predictive Medicine: Analyze electronic health records (EHRs), genomic data, medical imaging, and wearable device data to predict disease risks, enable personalized treatment plans, support early diagnosis, and improve patient outcomes.

Smart Cities and Intelligent Transportation Systems: Big data generated from IoT sensors, GPS devices, traffic cameras, and public transportation systems is analyzed to optimize traffic flow, reduce congestion, improve public safety, enhance energy management, and support urban planning. Predictive analytics helps in demand forecasting and efficient resource allocation.

Predictive Maintenance in Industry 4.0: Manufacturing industries leverage sensor data from IoT devices, machine logs, and operational records to predict equipment failures before they occur. Advanced analytics and machine learning models help reduce downtime, optimize maintenance schedules, and improve operational efficiency.

Recommendation Systems and Customer Personalization: E-commerce platforms, streaming services, and digital businesses employ advanced analytics techniques such as collaborative filtering, deep learning, and behavioral analysis to provide personalized product recommendations, targeted marketing campaigns, and enhanced customer experiences.

Learning Resources

Textbooks:

1. David Dietrich, Barry Hiller, “Data Science and Big Data Analytics”, EMC education services, Wiley publication, 2012, ISBN0-07-120413-X
2. 2. DT Editorial Services, Big Data, Black Book, DT Editorial Services, ISBN: 9789351197577, 2016 Edition.
3. EMC Education Services, “Data Science and Big Data Analytics- Discovering, analyzing Visualizing and Presenting Data”

Reference Books:

1. Chirag Shah, “A Hands-On Introduction To Data Science”, Cambridge University Press, (2020), ISBN : ISBN 978-1-108-47244-9
2. Wes McKinney, “Python for Data Analysis”, O' Reilly media, ISBN: 978-1-449-31979-3
3. Trent Hauk, “Scikit-learn Cookbook”, Packt Publishing, ISBN: 9781787286382
4. Jenny Kim, Benjamin Bengfort, “Data Analytics with Hadoop”, OReilly Media, Inc., ISBN: 9781491913703
5. Venkat Ankam,“Big Data Analytics”, Packt Publishing, ISBN: 9781785884696
6. Seema Acharya, Subhashini Chellappan, “Big Data And Analytics”, Wiley publication, ISBN: 9788126579518
7. Mitzenmacher and Upfal, Probability and Computing: Randomized Algorithms and Probabilistic Analysis, Cambridge University press, ISBN :521835402 hardback.
8. Dana Ron, Algorithmic and Analysis Techniques in Property Testing, School of EE.
9. Graham Cormode, Minos Garofalakis, Peter J. Haas and Chris Jermaine, Synopses for Massive Data: Samples, Histograms, Wavelets, Sketches, Foundation and trends in databases, ISBN :10.1561/1900000004.
10. Alex Holmes, Hadoop in practice, Dreamtech press, ISBN:9781617292224.
11. Ambiga Dhiraj, Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today’s Business, Wiley CIO Series.
12. ArvindSathi, Big Data Analytics: Disruptive Technologies for Changing the Game, IBM Corporation, ISBN:978-1-58347-380-1.
13. Li Chen, Zhixun Su, Bo Jiang, Mathematical Problems in Data Science, Springer, ISBN :978-

3-319-25127-1.
14. Philip Kromer and Russell Jurney, Big Data for chips, O'Reilly, ISBN :9789352132447.
15. Mueller Massaron, Python for Data science, Wiley, ISBN :9788126557394.
16. Benoy Antony, Konstantin Boudnik, Cheryl Adams,,Professional Hadoop, Wiley India, ISBN :9788126563029
17. Judith Hurwitz, Alan Nugent, Big Data For Dummies, Wiley India, ISBN : 9788126543281

e-Books:

1. An introduction to data Science : https://docs.google.com/file/d/0B6iefdnF22XQeVZDSkxjZ0Z5VUE/edit?pli=1
2. Hadoop Tutorial : https://www.tutorialspoint.com/hadoop/hadoop_tutorial.pdf?utm_source=7_&utm_medium=affiliate&utm_content=5f34cd37cdf1050001b09537&utm_campaign=Admitad&utm_term=761c575424fc4a6b48d02f72157eb578
3. Learning with Python; How to think like a computer scientist: http://openbookproject.net/thinkcs/python/english3e/
4. Python for everybody: http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf
5. Scikit Learn Tutorial https://scikit-learn.org/stable/

Links to online SWAYAM/NPTEL Courses:

1. NPTEL Python for Data Science, Prof. Ragunathan Rengasamy (IIT Madras)
2. NPTEL Big Data Computing, Prof. Rajiv Misra (IIT Patna)

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PEC-361A-AIM
Course Name: Power BI & Tableau

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

Prerequisite Courses, if any: **Programming Fundamentals, Database Management Systems.**

Companion Course, if any: Machine Learning, Data Analytics.

Course Objectives:

1. To understand the concepts of Business Intelligence (BI), data analytics, and data visualization.
2. To develop skills in data preparation, transformation, and modeling using Power BI and Tableau.
3. To design interactive dashboards and visual reports for decision-making.
4. To apply advanced analytics and AI-driven features available in BI tools.
5. To develop real-world business intelligence solutions using industry datasets.

Course Outcomes:

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

CO1: Explain the concepts of Business Intelligence, data analytics, data visualization principles, and the role of Power BI and Tableau in data-driven decision making.

CO2: Apply data preparation, transformation, integration, and data modeling techniques using Power BI and Tableau to create structured datasets for analysis.

CO3: Design interactive dashboards and visual reports using appropriate visualization techniques to communicate business insights effectively.

CO4: Analyze business data using advanced analytical features, DAX calculations, forecasting methods, and AI-assisted capabilities available in Power BI and Tableau.

CO5: Develop business intelligence solutions for real-world applications by utilizing cloud-based BI services, security mechanisms, real-time analytics, and emerging AI-powered BI technologies.

Course Contents

Unit No: I	**Fundamentals of Business Intelligence and Data Visualization**	**9 Hours**

Introduction to Business Intelligence: BI Concepts, Architecture, Process and components. **Data Analytics Fundamentals**: Descriptive Analytics, Diagnostic Analytics, Predictive Analytics, Prescriptive Analytics. **Introduction to Data Visualization**: Importance of Visualization, Data Visualization Principles, Data Storytelling, Visual Perception and Human Cognition. **Introduction to Power BI and Tableau**: Features and Architecture, Comparison of Power BI and Tableau, Applications in Industry. **Data Sources**: Excel, CSV, Databases, Web Data Sources and Cloud Data Sources.

Exemplars/Case Studies	Case Study 1: Netflix User Analytics Dashboard.

Unit No: II	Data Preparation and Data Modeling	9 Hours
Data Collection and Integration, Data Cleaning and Transformation, **Power Query Editor**: Query Transformation, Merge Queries, Append Queries, **Data Modelling**: Relationships, Star Schema, Snowflake Schema. **Calculated Fields and Measures**: DAX Fundamentals, Calculated Columns, Measures. **Tableau Data Preparation**: Tableau Prep Basics, Data Blending, Joins and Unions.		
Exemplars/Case Studies	Case Study on Retail Sales Data Analysis.	
Unit No: III	**Dashboard Design and Visualization Techniques**	**9 Hours**
Visualization Components: Charts, Graphs, Tables, Maps. **Power BI Visualizations**: Bar Charts, Line Charts, Pie Charts, Tree Maps, KPI Cards.**Tableau Visualizations**: Heat Maps, Scatter Plots, Geographic Maps, Histograms. **Dashboard Design Principles**: Layout Design, Color Theory, User Experience. **Interactive Dashboards**: Filters, Slicers, Drill Down, Drill Through.**Storytelling with Data**: Business Insights, Dashboard Narratives.		
Exemplars/Case Studies	Case Study on COVID-19 Global Dashboard.	
Unit No: IV	**Advanced Analytics and AI Integration**	**9 Hours**
Advanced DAX Functions, Tableau Calculations, Predictive Analytics, AI Features in Power BI, **AI Features in Tableau**: AI Visuals, Key Influencers, Smart Narratives. **Integration with Python and R:** Data Analysis using Python, Visualization Extensions. Business Forecasting Applications.		
Exemplars/Case Studies	Case study on Sales Forecasting Dashboard.	
Unit No: V	**Business Intelligence Applications and Emerging Trends**	**9 Hours**
BI Applications, Cloud-Based BI, Data Governance and Security, Real-Time Analytics, **AI-Powered Business Intelligence**: Generative AI in BI, Natural Language Querying, Conversational Analytics. **Emerging Trends**: Augmented Analytics, Self-Service BI, Data Fabric, Data Mesh.		
Exemplars/Case Studies	Case study on Smart City Analytics Dashboard.	

Learning Resources

Textbooks:

1. Chris Sorensen "Microsoft Power BI Data Analyst" 3rd Edition.
2. Cole knaflic "Storytelling with Data", WILEY Publisher.
3. Ryan Sleeper "Practical Tableau" O'Reilly Publisher.

Reference Books:

1. Steve Wexler, Jeffre Shaffer, Andy Cotgreave "The Big Book of Dashboards" WILEY Publisher.
2. Larrisa T. MoSs, Shaku Atre "Business Intelligence Roadmap".
3. Patrick Boily, Stephen Davies, Jen Schellinck "The Practice of Data Visualization".

e-Books: **1. https://link.springer.com/book/10.1007/978-1-4471-7527-8?utm_source** **2. https://researchportal.murdoch.edu.au/esploro/outputs/book/Decoding-Data-A-Complete-Guide-to/991005875624907891?utm_source** **3. https://link.springer.com/book/10.1007/978-3-031-28674-2?utm_source** **4. https://www.youtube.com/watch?v=e6QD8lP-m6E**
Links to online SWAYAM/NPTEL Courses: 1. https://onlinecourses.swayam2.ac.in/e-learning/preview/imb26_mg110 2. https://onlinecourses.swayam2.ac.in/e-learning/preview/imb25_mg221

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PEC-362B-AIM
Course Name: Large Language Models

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

Prerequisite Courses, if any: Deep Learning (PCC-302 -AIM), Natural Language Processing (PEC-321B-AIM)

Companion Course, if any: LLM Laboratory (PEC-363-AIM)

Course Objectives:

1. To understand the fundamentals of Natural Language Processing and language representation techniques.
2. To learn the architecture and working of Transformer-based models.
3. To understand the evolution and capabilities of Large Language Models.
4. To apply fine-tuning, prompt engineering, and RAG techniques for AI applications.
5. To evaluate and deploy responsible and effective LLM-based and multimodal AI systems.

Course Outcomes:

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

CO1: Explain NLP concepts, text preprocessing techniques, and word embeddings.

CO2: Apply Transformer architecture and attention mechanisms for NLP tasks

CO3: Analyze the architecture and working of Large Language Models.

CO4: Develop applications using fine-tuning, prompt engineering, and RAG techniques.

CO5: Evaluate the performance, deployment, and ethical aspects of LLM-based and multimodal AI systems

Course Contents

Unit No: I	**Foundations of Natural Language Processing (NLP)**	**7 Hours**

Introduction to Natural Language Processing (NLP) and its real-world applications. Basic text preprocessing techniques such as tokenization, stop-word removal, stemming, and lemmatization. Methods for representing text using vector space models and word embeddings including Word2Vec, GloVe, and FastText. Fundamentals of language modeling and sequence representation used in modern NLP systems.

Exemplars/Case Studies	Case Study : Sentiment Analysis of Product Reviews

Unit No: II	**Transformer Architecture and Attention Mechanisms**	**8 Hours**

Overview of traditional sequence modeling approaches and their limitations. Introduction to Encoder-Decoder architecture and the concept of attention. Study of self-attention, multi-head attention, and positional encoding. Understanding the Transformer architecture and its role as the foundation of modern language models.

Exemplars/Case Studies	Machine Translation using Transformers	
Unit No: III	**Large Language Models and Their Evolution**	**8 Hours**
Introduction to foundation models and Generative AI. Study of popular language model architectures such as GPT, BERT, RoBERTa, T5, and BART. Understanding different training approaches including autoregressive and masked language modeling. Overview of scaling laws, emergent capabilities, and recent developments in Large Language Models.		
Exemplars/Case Studies	Comparative Analysis of GPT and BERT	
Unit No: IV	**Fine-Tuning and Prompt Engineering**	**7 Hours**
Introduction to the **Hugging Face ecosystem**, including Transformers, Datasets, Tokenizers, Model Hub, and Inference APIs for accessing and deploying pre-trained language models. Techniques for adapting pre-trained language models to specific tasks through fine-tuning and transfer learning. Introduction to Parameter-Efficient Fine-Tuning (PEFT), LoRA, and model quantization. Prompt engineering methods including zero-shot, few-shot, and chain-of-thought prompting. Concepts of Retrieval-Augmented Generation (RAG) and the use of vector databases for enhancing model responses with external knowledge.		
Exemplars/Case Studies	Domain-Specific Chatbot for Higher Education	
Unit No: V	**Multimodal AI, Evaluation and Responsible AI**	**8 Hours**
Introduction to multimodal AI systems capable of processing text, images, and audio. Methods for evaluating and benchmarking Large Language Models. Understanding challenges such as hallucinations, bias, and fairness in AI systems. Principles of responsible AI, including safety, privacy, and ethical considerations. Overview of tools and frameworks for developing and deploying LLM-based applications, along with emerging trends in AI agents and intelligent systems.		
Exemplars/Case Studies	Multimodal AI, Evaluation and Responsible AI	
Learning Resources		
Textbooks: 1. Tanmoy Chakraborty, "Introduction to Large Language Models", Wiley India, 1st Edition, 2025. ISBN : 9789363864740 2. Dan Jurafsky and James H. Martin, Speech and Language Processing, 2nd edition, Pearson Press, 2008.		
Reference Books: 1. Jacob Eisenstein, Natural Language Processing, First edition,The MIT Press, 2019. 2. Jay Alammar and Maarten Grootendorst, "*Hands-On Large Language Models: Language Understanding and Generation", O'Reilly* 3. Dan Jurafsky and James H. Martin, "Speech and Language Processing (3rd Edition)"		
e-Books: **1. https://tjzhifei.github.io/resources/NLTK.pdf** **2. https://github.com/codewithdark-git/Building-LLMs-from-**		

scratch/blob/main/Foundations%20of%20LLM.pdf
Links to online SWAYAM/NPTEL Courses: 1. https://onlinecourses.nptel.ac.in/e-learning/preview/noc25_cs45

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PEC-362C-AIM
Course Name: Mobile Application Development (Elective 3)

Teaching Scheme	Credits	Examination Scheme
Theory: 3 Hrs/Wk	3	CCE - 30 Marks ESE - 70 Marks

Prerequisite Courses, if any:

Companion Course, if any:

Course Objectives:

1. Understand the concepts, architecture, platforms, and lifecycle of mobile application development.
2. Apply UI/UX design principles to develop responsive and user-friendly mobile applications.
3. Develop mobile applications utilizing device resources, local storage, and communication services.
4. Integrate cloud platforms, web services, and AI/ML capabilities into mobile applications.
5. Evaluate, test, optimize, secure, and deploy mobile applications using industry-standard practices.

Course Outcomes:

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

CO1: Explain mobile computing concepts, architectures, operating systems, and development methodologies.
CO2: Design mobile user interfaces and navigation systems following established UI/UX principles.
CO3: Develop mobile applications incorporating storage mechanisms, sensors, and device services.
CO4: Integrate APIs, cloud services, and AI/ML models into mobile applications.
CO5: Test, secure, optimize, and deploy mobile applications for practical applications.

Course Contents

Unit No: I	Introduction to Mobile Application Development	9 Hours

Evolution of mobile computing, mobile operating systems and architectures, Android architecture overview, iOS architecture overview, native applications, hybrid applications, cross-platform applications, mobile application development lifecycle, development environment setup, Flutter framework overview, Dart programming fundamentals, mobile application market trends, AI-enabled mobile applications and future directions.

Exemplars /Case Studies	**Case Study 1: WhatsApp Mobile Architecture and Cross-Platform Communication Framework** **Case Study 2: Instagram Mobile Application Evolution from Native Development to Scalable Mobile Platform**

Unit No: II	Mobile UI/UX Design and Application Components	9 Hours

Principles of mobile UI/UX design, Material Design guidelines, Flutter widgets and widget hierarchy, layout management, responsive design techniques, navigation and routing, forms and user input handling, state management concepts, accessibility standards, usability evaluation methods, user engagement strategies, interface prototyping techniques.		
Exemplars/Case Studies	**Case Study 1: User Experience Design Analysis of Google Maps Mobile Application** **Case Study 2: User Interface and Engagement Strategies in Spotify Mobile Application**	
Unit No: III	**Data Management and Device Integration**	**9 Hours**
Data persistence in mobile applications, shared preferences, SQLite database, file handling operations, local storage management, sensors in mobile devices, camera integration, GPS and location-based services, notifications and alerts, background services, multimedia integration, mobile security fundamentals, permissions management.		
Exemplars/Case Studies	**Case Study 1: Ola/Uber Real-Time Ride Tracking System using GPS and Location Services** **Case Study 2: Google Fit Mobile Application and Sensor-Based Activity Monitoring**	
Unit No: IV	**Backend Integration and AI-Enabled Mobile Applications**	**9 Hours**
RESTful web services, HTTP methods and communication protocols, JSON data exchange, API integration techniques, Firebase services, authentication and authorization mechanisms, cloud databases, cloud storage integration, AI and machine learning in mobile applications, TensorFlow Lite framework, on-device machine learning, recommendation systems, conversational AI and chatbot integration.		
Exemplars/Case Studies	**Case Study 1: AI-Based Personalized Recommendation Engine in Swiggy/Zomato** **Case Study 2: Google Lens and On-Device Image Recognition using TensorFlow Lite**	
Unit No: V	**Testing, Security, Optimization and Deployment**	**9 Hours**
Mobile application testing methodologies, unit testing, widget testing, integration testing, debugging techniques, performance profiling, memory optimization, battery optimization strategies, secure coding practices, application security mechanisms, version control using Git, CI/CD pipelines for mobile applications, application publishing on Play Store and App Store, emerging trends in wearable computing, augmented reality applications, edge AI applications.		
Exemplars/Case Studies	**Case Study 1: Deployment, Security, and Scalability of Aarogya Setu Mobile Application** **Case Study 2: Pokémon GO: Mobile Application Testing, Optimization, and Large-Scale Deployment**	
Learning Resources		

Textbooks:

1. Barry Burd, Flutter For Dummies, 2nd Edition, Wiley Publications, 2023.
2. Dawn Griffiths and David Griffiths, Head First Android Development, 3rd Edition, O'Reilly Media, 2024.

Reference Books:

1. Thomas Künneth, Flutter and Dart Cookbook: Developing Full-Featured Apps for the Mobile and Web, Packt Publishing, 2023.
2. Wei-Meng Lee, Beginning Android Application Development, 6th Edition, Wiley Publications, 2022.
3. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition, O'Reilly Media, 2022.

e-Books: https://docs.flutter.dev/

1. https://developer.android.com/

Links to online SWAYAM/NPTEL Courses:

1. https://onlinecourses.swayam2.ac.in/e-learning/preview/nou21_ge41
2. https://onlinecourses.swayam2.ac.in/e-learning/preview/ntr25_ed126
3. https://swayam-plus.swayam2.ac.in/courses/course-details?id=P_SMARTBRIDGE_06&utm

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PEC-362D-AIM
Course Name: Cyber Security & Risk Management (Elective III)

Teaching Scheme	Credits	Examination Scheme
Theory: 3 Hrs./WK	3	CCE - 30 Marks ESE - 70 Marks

Prerequisite Courses, if any:

Companion Course, if any: Cyber Security & Risk Management Lab

Course Objectives:

1. To introduce the fundamental concepts of cybersecurity, cyber threats, vulnerabilities, and security frameworks.
2. To develop understanding of cryptographic techniques, authentication mechanisms, and secure communication protocols.
3. To familiarize students with network security technologies and mechanisms for protecting organizational networks and systems.
4. To provide knowledge of cyber risk management, security architecture principles, and security governance frameworks.
5. To explore emerging cybersecurity challenges in cloud computing, IoT/OT systems, AI-enabled environments, and applicable cyber laws and ethics.

Course Outcomes:

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

CO1: Explain cybersecurity concepts, threat landscapes, attack methodologies, and security frameworks used to protect information assets.

CO2: Apply cryptographic algorithms, key management techniques, digital signatures, and PKI mechanisms to secure data and communications.

CO3: Analyze and implement network security controls including firewalls, IDS/IPS, VPNs, segmentation, wireless security, and DNS security mechanisms.

CO4: Assess cybersecurity risks and recommend appropriate security architectures, design principles, and risk treatment strategies for organizational environments.

CO5: Evaluate security challenges and mitigation approaches in cloud, IoT/OT, and AI-driven systems while complying with cybersecurity laws, privacy regulations, and ethical practices.

Course Contents

Unit No: I	INTRODUCTION TO CYBER SECURITY & THREAT LANDSCAPE	6 Hours

Foundations of Cybersecurity: Definition, goals: Confidentiality, Integrity, and Availability; extended model: Authenticity, Non-repudiation, and Accountability,

Cybersecurity vs. Information Security vs. Network Security: conceptual differentiation;

The NIST Cybersecurity Framework (CSF 2.0) — Identify, Protect, Detect, Respond, Recover

Classification of threats: passive vs. active attacks; insider vs. outsider threats

Malware taxonomy: viruses, worms, trojans, ransomware, spyware, adware, rootkits, fileless malware, Social engineering: phishing, spear-phishing, vishing, smishing, pretexting, baiting, tailgating

Attack surface concept: internal, external, social/human surfaces Vulnerability vs. threat vs. exploit vs. risk		
Exemplars /Case Studies	**Case Study 1:** WannaCry ransomware (2017) **Case Study 2:** Colonial Pipeline attack (2021)	
Unit No: II	**CRYPTOGRAPHY & NETWORK SECURITY**	**8 Hours**
Classical ciphers: Caesar, Vigenere, transposition, cryptanalysis; Symmetric key cryptography: DES, 3DES, AES-128/192/256, block cipher modes (ECB, CBC, CFB, OFB, CTR, GCM) Asymmetric (public-key) cryptography: RSA algorithm — key generation, encryption, decryption; Diffie-Hellman key exchange; ECC Hash functions: SHA-1, SHA-2 (SHA-256, SHA-512), SHA-3/SHAKE; collision resistance, pre-image resistance; HMAC Key management: key generation, distribution, storage, revocation, escrow; Key Management Interoperability Protocol (KMIP) Digital signatures: RSA-SHA256, ECDSA, Public Key Infrastructure (PKI): X.509 certificates, Certificate Authority (CA), Registration Authority		
Exemplars/Case Studies	**Case Study 1:** WhatsApp End-to-End Encryption (Meta & Signal Protocol) **Case study 2:** TLS/SSL in E-Commerce (Amazon, Banking, UPI Platforms)	
Unit No: III	**Network Security Mechanisms**	**6 Hours**
Firewalls: packet filtering, stateful inspection, application-layer gateways, next-generation firewalls (NGFW); rule design principles Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS): signature-based, anomaly-based, specification-based; Snort/Suricata overview Virtual Private Networks (VPN): IPSec (AH, ESP, IKEv2), SSL-VPN, WireGuard; Network segmentation: DMZ architecture, VLAN-based segmentation, micro-segmentation Wireless security: WPA3, 802.1X EAP, rogue AP detection; Bluetooth/BLE security; 5G network security considerations DNS security: DNSSEC, DNS over HTTPS (DoH), DNS over TLS (DoT); common DNS attacks		
Exemplars/Case Studies	**Case Study:** Target Corporation Data Breach (2013)	
Unit No: IV	**CYBER RISK MANAGEMENT FRAMEWORKS, SECURITY ARCHITECTURE**	**6 Hours**

Risk management concepts: threat × vulnerability × asset value = risk; risk appetite, risk tolerance, residual risk Risk management process: asset identification and valuation, threat identification, vulnerability assessment, likelihood and impact estimation Risk treatment strategies: Avoidance, Mitigation, Transfer (cyber insurance), Acceptance; Residual risk documentation Security design principles: least privilege, separation of duties, defense-in-depth, fail-safe defaults, economy of mechanism, open design Zero Trust Architecture, Security Reference Architectures: SABSA (Sherwood Applied Business Security Architecture), OSA (Open Security Architecture)		
Exemplars/Case Studies	**Case Study:** Colonial Pipeline Ransomware Attack (2021)	
Unit No: V	**EMERGING THREATS, CLOUD SECURITY, AI & CYBER LAW**	**6 Hours**
Cloud Security: CSA (Cloud Security Alliance), Cloud Controls Matrix (CCM), Cloud-specific threats, Container and Kubernetes security IoT & OT/ICS Security: IoT attack surface: device firmware vulnerabilities, default credentials, insecure update mechanisms, protocol weaknesses (MQTT, CoAP, Zigbee) Operational Technology (OT) / Industrial Control Systems (ICS) security: SCADA, PLC vulnerabilities, Critical infrastructure protection AI/ML in Cybersecurity: AI for defense: anomaly detection using ML Adversarial AI: adversarial examples, model poisoning/evasion attacks Generative AI threats: deepfake attacks, AI-generated phishing Cyber Laws & Ethics: National Cyber Security Policy 2013 (India), Digital Personal Data Protection (DPDP) Act 2023, Ethics in cybersecurity		
Exemplars/Case Studies	Case study: Deepfake Attacks, Protocol Weaknesses: MQTT, COAP	
Learning Resources		
Textbooks: 1. Cryptography and Network Security: Principles and Practice (8th Edition) , William Stallings, Pearson Education 2. Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Nina Godbole and Sunit Belapure, Wiley Publication		
Reference Books: 1. Introduction to Modern Cryptography (3rd Edition), Jonathan Katz and Yehuda Lindell, CRC Press 2. Information Security Principles and Practice (2nd Edition), Mark Stamp, Wiley Publication 3. Zero Trust Networks: Building Secure Systems in Untrusted Networks (2nd Edition), Evan Gilman and Doug Barth, O'Reilly Media 4. Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems, David J. Tyson and Eric D. Knapp, Elsevier 5. Computer Security: Art and Science (2nd Edition), Matt Bishop, Addison-Wesley		
e-Books: 1. The NIST Cybersecurity Framework 2.0 (Official Publication), National Institute of Standards and Technology (NIST)		

2. Threat Actor Taxonomy & Malware Analysis Handbook, Cybersecurity and Infrastructure Security Agency (CISA) **3.** Cloud Security Alliance (CSA) Cloud Controls Matrix (CCM) v4
1. Links to online SWAYAM/NPTEL Courses: 2. https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs57 3. https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs18

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PEC-363A-AIM
Course Name: Power BI & Tableau Lab(Elective 3 Lab 3)

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

Course Objectives:

1. Understand the process of data acquisition, integration, cleaning, and transformation using Business Intelligence tools.
2. Develop skills in data modeling, relationship creation, and analytical report generation using Power BI and Tableau.
3. Design interactive dashboards and visualizations to communicate business insights effectively.
4. Apply advanced analytical techniques such as DAX calculations, forecasting, and Python integration for data-driven decision-making.
5. Build end-to-end Business Intelligence solutions for real-world applications using modern BI tools and datasets.

Course Outcomes:

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

CO1: Import, clean, transform, and integrate data from multiple sources using Power BI and Tableau.

CO2: Create data models and establish relationships among datasets to support analytical reporting.

CO3: Develop interactive dashboards and visualizations using appropriate charts, filters, slicers, and drill-down features.

CO4: Perform advanced analytics using DAX functions, forecasting techniques, and Python integration to extract actionable business insights.

CO5: Design and implement a complete Business Intelligence solution for a real-world problem by integrating data preparation, modeling, visualization, and analytics techniques.

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.

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
2. parameters such as timely conduction of practical assignment, methodology adopted for implementation of practical assignment, timely submission of assignment in the form of
3. Handwritten write-up along with results of implemented assignment, attendance etc.
4. 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.
5. 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 them 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.

Note:

Students may use publicly available large-scale datasets from Kaggle, UCI Repository, Government, Open Data Portals, or other authentic sources for performing laboratory experiments and mini projects.

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

(Using Power BI Desktop)

1. An educational institution aims to improve student performance by analyzing academic records. Develop a Business Intelligence dashboard to evaluate student academic performance by importing and analyzing Student data containing marks, attendance, and assignment scores. Identify factors influencing student Success, determine top-performing and low-performing students, and generate interactive dashboards that provide meaningful insights to support academic decision-making and student performance improvement.
   -- CO1, CO2, CO3

2. A retail organization maintains customer and sales data collected from multiple sources. The dataset contains missing values, duplicate records, inconsistent formats, and invalid entries. Clean and transform the data using Power Query to improve data quality and prepare it for business analysis.
   -- CO1

3. An e-commerce company stores customer, order, and product information in separate tables. Create appropriate relationships and design an efficient data model to support analytical reporting. --CO2

(Using Power BI Desktop/Tableau Desktop & Python- Any SIX)
4. Develop an interactive dashboard using different visualization techniques such as bar charts, line charts, pie charts, maps, KPIs, cards, tree maps, and slicers to effectively communicate business insights and performance indicators. --CO3
5. An organization stores customer transaction data in a relational database. Establish a connection between Power BI/Tableau and the database, retrieve customer purchasing information, and create visualizations to identify customer segments, purchasing patterns, and high-value customers for business decision-making. --CO1, CO2, CO3
6. Design an interactive business dashboard that enables users to dynamically explore sales and customer information using filters, slicers, drill-down, and drill-through features. --CO3
7. Develop business metrics such as total sales, profit margin, year-to-date sales, and customer retention rate using DAX calculations to support decision-making. --CO4
8. Analyze historical sales data and generate future sales forecasts using built-in forecasting techniques available in Power BI or Tableau. --CO4
9. Integrate Python scripts with Power BI to perform advanced data analysis and generate customized Visualizations for business insights. --CO4
10. An organization maintains sales, customer, and product information in different file formats and database systems. Import and connect datasets from multiple sources, including relational databases, into Power BI, perform data integration and modeling, and prepare a unified repository for analysis. Design an interactive dashboard and publish it on Power BI Service to facilitate cloud-based reporting and decision-making **It's a mandatory assignment)** --CO1, CO2, CO3
(Mini Project – Any ONE)
1. Design and implement a complete Business Intelligence solution for a selected domain such as Retail, Healthcare, Banking, Education, Agriculture, or Social Media Analytics. The solution should include data acquisition, data cleaning, modeling, visualization, dashboard development, and business insight generation. --CO1, CO2, CO3, CO4, CO5
2. Develop a Business Intelligence dashboard to analyze Electric Vehicle (EV) adoption trends, market growth, and regional distribution using historical EV registration and sales data. The dashboard should provide insights into EV penetration, manufacturer performance, and growth patterns to support data-driven decision-making in the sustainable transportation sector. --CO1,CO2, CO3, CO4, CO5

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PEC-363B-AIM
Course Name: Elective-3 Lab-3 (Large Language Models)

Teaching Scheme	Credits	Examination Scheme
Practical : 02 Hrs/week	1	TW : 25 PR : 25

Course Objectives:

1. To understand the fundamentals of Large Language Models (LLMs) and the Hugging Face ecosystem.
2. To apply tokenization, embeddings, and prompt engineering techniques for NLP tasks.
3. To develop LLM-based applications using pre-trained models.
4. To implement fine-tuning and Retrieval-Augmented Generation (RAG) techniques.
5. To evaluate the performance of LLMs using standard evaluation metrics.

Course Outcomes:

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

CO1: Explain the concepts of LLMs, tokenization, embeddings, and the Hugging Face ecosystem

CO2: Apply prompt engineering techniques for various NLP tasks

CO3: Develop LLM-based applications such as chatbots using pre-trained models

CO4: Implement fine-tuning and RAG-based solutions for domain-specific applications

CO5: Evaluate and compare LLM performance using appropriate metrics

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. Experiments to be conducted using relevant open-source tools.

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
2. parameters such as timely conduction of practical assignment, methodology adopted for implementation of practical assignment, timely submission of assignment in the form of

3. handwritten write-up along with results of implemented assignment, attendance etc. 4. 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. 5. 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 them among batches of students. It is appreciated if the assignments are based on real world problems/applications. All the assignments should be conducted on GPU-enabled computing resources and the necessary software frameworks for the development, fine-tuning, and deployment of Large Language Model (LLM) applications.
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 (Any five assignments from Sr. No. 1 to 7 using any open-source tool)
1. Use the Hugging Face platform and pre-trained models to perform text classification, sentiment analysis, text summarization, and text generation
2. Implement different tokenization techniques and generate text embeddings to analyze word and sentence representations
3. Apply Zero-Shot, Few-Shot, and Chain-of-Thought prompting techniques and compare their performance on NLP tasks
4. Develop a chatbot using a pre-trained LLM to answer queries in a selected application domain.
5. Fine-tune a pre-trained language model using PEFT/LoRA for a domain-specific task and evaluate its performance
6. Build a RAG-based question-answering system using external documents and a vector database
7. Evaluate and compare LLM outputs using metrics such as BLEU, ROUGE, and human assessment

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PEC-363C- AIM
Course Name: Mobile Application Development (Elective 3 Lab 3)

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

Course Objectives:

1. Develop mobile applications using modern development tools and frameworks.
2. Design interactive and responsive user interfaces.
3. Implement data storage and device integration functionalities.
4. Integrate cloud services, APIs, and AI functionalities into mobile applications.
5. Test, debug, and deploy mobile applications effectively.

Course Outcomes:

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

CO1: Configure and utilize mobile application development environments.

CO2: Design and implement responsive mobile user interfaces.

CO3: Develop applications utilizing local storage and device hardware resources.

CO4: Integrate cloud services, APIs, and AI/ML functionalities into mobile applications.

CO5: Test, debug, optimize, and deploy mobile 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.

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 the faculty concerned and head of the department.
5. All the assignments 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
2. parameters such as timely conduction of practical assignment, methodology adopted for implementation of practical assignment, timely submission of assignment in the form of
3. handwritten write-up along with results of implemented assignment, attendance etc.
4. 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.

5. 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 Visual Studio Code (VS Code) with Flutter SDK and Android Emulator

The mini project may be carried out either **individually or in a group of minimum 2 and maximum 3 students**. The instructor shall ensure that each student/group undertakes a **unique project based on a different application domain or problem statement**, avoiding duplication of topics. Projects should encourage the application of mobile app development concepts, backend integration, cloud services, and AI/ML features wherever appropriate.

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

Virtual Laboratory

https://appinventor.mit.edu/
https://firebase.google.com/

1. Development Environment Setup and First Mobile Application-- CO1

Install and configure Flutter SDK, Dart SDK, Android Studio, VS Code, and Android Emulator. Create, build, and execute a simple "Hello AIML" mobile application demonstrating the mobile application development workflow.

2. Personal Portfolio Mobile Application – CO2

Design and develop a mobile application displaying personal information, educational details, skills, and contact information using Flutter widgets, layouts, images, and navigation components.

3. Student Registration and Validation System -- CO2

Develop a student registration application incorporating multiple input fields, form validation, dropdown menus, radio buttons, checkboxes, and navigation between screens.

4. To-Do List Application Using Local Storage/Cloud –CO3, CO4

Create a To-Do List mobile application that allows users to add, edit, delete, and mark tasks as completed. Store the task information using Shared Preferences.

5. Weather Information Application Using REST API – CO4

Develop a mobile application that retrieves real-time weather information from a public REST API, processes JSON responses, and displays weather forecasts in a user-friendly format.
6. Cloud-Based Authentication System - CO3, CO4
Implement Firebase Authentication and Cloud Firestore in a mobile application to provide user registration, login, profile management, and secure cloud-based data storage.
7. AI-Enabled Mobile Application Testing and Deployment—CO3, CO4, CO5
Develop a mobile application integrating a TensorFlow Lite model. Perform unit testing, widget testing, debugging, performance analysis, and optimization of a developed application. Maintain source code using Git and prepare the application for deployment
Mini Projects (Students can perform any one of below Mini Project)
1. AI Chatbot for Student Assistance -- CO2, CO4, CO5
Develop an AI-powered chatbot capable of answering frequently asked academic queries and providing student support services.
2. Smart Expense Tracker -CO3, CO4, CO5
Design a mobile application that records expenses, categorizes transactions, and provides AI-based spending insights and recommendations.

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: PEC-363-AIM
Course Name: Cyber Security & Risk Management Laboratory

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

Course Objectives:

1. To gain a fundamental knowledge of Cyber Threats and Cyber Vulnerabilities.
2. To understand the security concerns in Network
3. To impart theoretical and mathematical principles of classical ciphers, symmetric block ciphers, public-key cryptography, and Public Key Infrastructure (PKI) workflow
4. Design and deploy custom network intrusion detection signatures.
5. To provide hands-on experience in setting up secure operating system environments (Kali Linux) and configuring defensive perimeters using firewalls and browser security parameters

Course Outcomes:

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

CO1: Deploy Kali Linux, configure network firewalls, and harden web applications against hijacking threats.

CO2: Execute network scanning using Nmap/Zenmap and automate database flaw identification using exploitation tools.

CO3: Construct custom Snort rules to parse live packet payloads and isolate network threats.

CO4: Build operational Python client-server scripts utilizing Caesar ciphers, RSA encryption, and digital signature authentication.

CO5: Implement safe hybrid security pipelines combining Diffie-Hellman key exchanges with symmetric DES encryption.

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. 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
2. 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.

3. 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.
4. 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 them among batches of students. It is appreciated if the assignments are based on real world problems/applications.

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. Install Kali Linux and utilize various security tools for the defence of the system
2. Implement a penetration testing tool that automates the process of detecting and exploiting SQL Injection flaws and taking over of Database servers
3. Identify active network hosts, open ports, and potential vulnerabilities within a local sandboxed environment using NMAP/Zenmap
4. Implement a client and a server on different computers using python. Perform the communication between these two entities by using RSA cryptosystem.
5. Implement a client and a server on different computers using python. Perform the authentication of sender between these two entities by using RSA digital signature cryptosystem.
6. Implement a client and a server on different computers using python. Perform the encryption of message of sender between these two entities by using DES Algorithm and use Diffie Hellman method for exchange of keys.
7. Design, implement, and analyse a software application in Python to simulate the classical substitution-based cryptographic technique known as the Caesar Cipher.
8. Study of the features of NSA in providing network security and set firewall with proper configurations
9. Configure security parameters in any one of web browser and study of different types of vulnerabilities for hacking a websites / Web Applications
10. Use the snort intrusion detection package to analyze traffic and create a signature to identify problem traffic.

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: MDM-371-AIM
Course Name: Multimedia Technology

Teaching Scheme	**Credits**	**Examination Scheme**
Tutorial : 1 hr/wk	1	TW – 25 Marks
Practical : 2 hrs/wk	1	Oral - 25 Marks

Prerequisite Courses, if any: Object-Oriented Programming, Data Structures and Algorithms, Database Management Systems

Companion Course, if any: Multimedia Technology Practical

Course Objectives: The objectives of this course are:

1. To understand the fundamentals of multimedia systems and digital media representation.
2. To develop practical skills in image, audio, and video editing techniques.
3. To explore multimedia compression, storage, and transmission methods.
4. To apply AI and machine learning concepts in multimedia applications.
5. To design interactive multimedia projects using modern tools and platforms.

Course Outcomes: After successful completion of the course, students will be able to:
CO1: Explain multimedia components, architectures, and media representation techniques.
CO2: Perform basic image, audio, and video editing using multimedia tools.
CO3: Apply multimedia compression and storage techniques for efficient media handling.
CO4: Develop AI-assisted multimedia applications for classification, enhancement, and content generation.
CO5: Design and present an interactive multimedia project integrating multiple media elements.

Course Contents for Tutorial

Unit No: I	**Fundamentals of Multimedia Systems**	**03 Hours**

Introduction to Multimedia, Multimedia Components: Text, Graphics, Audio, Video, Animation: Multimedia Hardware and Software Requirements, Multimedia Data Representation, Modern Multimedia File Formats: JPEG, PNG, SVG, WebP, HEIF, RAW, MP3, WAV, FLAC, AAC, MP4, MKV, WebM, GIF, AVIF, Applications of Multimedia in AI, Healthcare, Education, Entertainment, and Smart Systems, Multimedia Authoring Tools and Platforms.

Exemplars/Case Studies	**Case Study 1: AI**-generated music, Music recommendation systems: Spotify **Case Study 2:** AI in Entertainment: Netflix / YouTube https://brollyacademy.com/applications-of-ai-in-education/

Unit No: II	**Digital Image Processing and Graphics**	**03 Hours**

Fundamentals of Digital Images, Image Formats and Color Models, Latest Image Formats: WebP, AVIF, HEIF, RAW Image Formats, Image Enhancement Techniques, Basics of Computer Graphics, Image Transformation and Filtering, Introduction to AI-based Image Processing.

Exemplars/Case Studies	Image editing using Canva/GIMP/Photoshop

Unit No: III	**Audio and Video Processing Techniques**	**03 Hours**

Fundamentals of Digital Audio and Video, Modern Audio and Video Formats: MP3, AAC, FLAC, WAV, MP4, MKV, MOV, WebM, H.264, H.265 $HEVC$, AV1, Audio Sampling and Encoding, Video Formats and Frame Processing, Audio and Video Compression Techniques, Streaming and Multimedia Communication, AI-based Speech and Video Applications

Exemplars/Case Studies	Intelligent Video Streaming and Compression in YouTube and Netflix

	Platforms	
Unit No: IV	**Multimedia Compression and Intelligent Multimedia Systems**	**03 Hours**
Need for Compression, Lossless and Lossy Compression, Compression Standards: JPEG, MPEG, MP3, H.264, H.265 (HEVC), Multimedia Storage and Transmission, Introduction to Content-Based Multimedia Retrieval, AI in Multimedia Analytics and Recommendation Systems		
Exemplars/Case Studies	• Zoom AI-Based Video Conferencing and Real-Time Multimedia Compression • YouTube AI Recommendation and Content-Based Video Retrieval System	
Unit No: V	**Interactive Multimedia and Emerging Trends**	**03 Hours**
Interactive Multimedia Systems, Virtual Reality (VR) and Augmented Reality (AR), Multimedia in Metaverse and Gaming, Generative AI in Multimedia, Ethical Issues in AI-generated Media, Future Trends in Intelligent Multimedia Systems		
Exemplars/Case Studies	**AI avatar and multimedia content generation** • https://www.sprinklr.com/blog/ai-social-media-content-creation/ • https://www.synthesia.io/	

Learning Resources

Textbooks:

1. Fundamentals of Multimedia by Ze-Nian Li, Mark S. Drew, Jiangchuan Liu, Springer, Second Edition
2. Multimedia: Making It Work by Tay Vaughan, McGrawHill, 8th Edition

Reference Books:

1. Multimedia Systems by Ralf Steinmetz & Klara Nahrstedt
2. Multimedia Technology and Applications by David Hillman

e-Books

1. **https://drive.uqu.edu.sa/_/mskhayat/files/MySubjects/20178FS%20Multimedia%20Systems/Fundamentals_of_multimedia_2e.pdf**
2. **https://eopcw.com/find/downloadFiles/385**

Links to online SWAYAM/NPTEL Courses

1. https://onlinecourses.swayam2.ac.in/e-learning/preview/ntr26_ed118
2. https://nptel.ac.in/courses/117105083

List of Tutorials:

1. AI-based Multimedia Advertisement
2. Smart Educational Video Creator
3. Interactive Product Presentation
4. AI-assisted Image Enhancement System
5. Multimedia Portfolio Website
6. Voice-based Multimedia Assistant
7. Short Documentary with AI Narration
8. AI avatar/video generation using RunwayML or Synthesia demos
9. Speech recognition using Google Speech API
10. AI subtitle generation
11. Emotion detection from voice samples
12. Face detection using OpenCV
13. Object tracking in videos

14. https://blog.tensorflow.org/2023/10/simulated-spotify-listening-experiences-reinforcement-learning-tensorflow-tf-agents.html

Current Industry Focus is for the following tutorials:

- AI-driven multimedia systems
- Web media optimization
- Streaming platforms
- Generative AI content pipelines
- Modern digital content creation ecosystems.

List of Tools:

- Canva
- GIMP / Adobe Photoshop
- Audacity
- OpenShot / Filmora / Adobe Premiere Rush
- Blender
- Google Teachable Machine
- TensorFlow MediaPipe (basic demo)
- RunwayML / Generative AI Tools
- https://artlist.io/
- https://deepmind.google/models/lyria/

PART A (Using recommended tools)

1. Study of Multimedia Tools and File Formats---CO1

Understand multimedia software ecosystem
Explore multimedia file properties
Tools recommended:
Canva, GIMP, VLC Media Player

2. Digital Image Editing and Enhancement-----CO1,CO2

- Brightness/contrast adjustment
- Background removal
- Color correction
- Image filtering

Tools recommended:
GIMP / Photoshop

3. AI-based Image Generation---CO3

- Generate AI artwork
- Perform image upscaling
- AI image restoration

Tools recommended:
DALL·E, Canva AI, Leonardo AI

PART B (Using Python)

4. Audio Recording and Noise Reduction---CO2

- Record audio samples
- Remove background noise

- Apply audio effects
- Normalize audio

Tools recommended: Audacity

5. Speech-to-Text and Subtitle Generation—CO3

- Convert speech into text
- Generate subtitles
- Analyze transcription accuracy

Tools recommended:
Google Speech API / Whisper AI

6. Video Editing and Multimedia Storytelling--- CO1, CO3, CO4

- Video trimming
- Transition effects
- Audio synchronization
- Title and caption addition

Tools recommended:
OpenShot / Filmora

7. Face Detection and Object Recognition-- CO3, CO4

- Detect faces in images/videos
- Perform object tracking
- Analyze AI detection performance

Tools recommended:
Python OpenCV

8. Multimedia Compression Analysis—CO5

- Compare compressed/uncompressed media
- Evaluate storage optimization
- Analyze quality degradation

Formats:
JPEG, PNG, WebP, H.264, AV1

9. AI-based Interactive Multimedia Application-- CO4, CO5

- Create chatbot-integrated multimedia interface
- Add voice interaction
- Create interactive presentation

Tools recommended:
Streamlit / Canva / AI APIs

10. Mini Project: Intelligent Multimedia System-- CO1, CO2, CO3, CO4, CO5

Example Projects

- Smart AI video summarizer
- AI poster generator
- Emotion-aware media player
- AI educational content creator
- Multimedia recommendation system

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: VSE-372-AIM
Course Name: Full Stack Development with MERN

Teaching Scheme	Credits	Examination Scheme
Practical : 2 Hrs./wk.	1	TW : 50

Course Objectives:

1. To introduce students to modern full-stack web application development using the MERN stack.
2. To develop interactive user interfaces using React.js components and state management.
3. To design and implement RESTful APIs using Node.js and Express.js.
4. To integrate MongoDB databases with web applications for data management.
5. To develop AI-ML oriented web applications using full-stack development principles.

Course Outcomes:

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

CO1: Develop interactive front-end applications using React.js components and hooks.
CO2: Implement server-side functionalities and REST APIs using Node.js and Express.js.
CO3: Design and integrate MongoDB databases for data storage and retrieval.
CO4:Analyse and integrate front-end, back-end, and database components to develop full-stack applications.
CO5: Develop AI-ML oriented web applications using MERN stack technologies.

Guidelines for Instructor's Manual

The faculty member should prepare the laboratory manual for all assignments and make it available to students and laboratory instructors. The manual should include objectives, outcomes, implementation guidelines, sample datasets, test cases, evaluation criteria, and references.

Guidelines for Student's Lab Journal

1. Students shall maintain a laboratory journal containing objectives, problem statements, implementation details, outputs, observations, and conclusions.
2. Source code and screenshots of implemented assignments shall be attached.
3. The journal shall be certified by the concerned faculty member.
4. All assignments mentioned in the syllabus must be completed.
5. Term work assessment shall be based on laboratory performance and journal submission.

Guidelines for Lab /TW Assessment

1. Continuous assessment shall be based on implementation, understanding of concepts, and timely completion of assignments.
2. Proper documentation, coding standards, debugging practices, and result analysis shall be considered during evaluation.
3. Students shall demonstrate the execution and functionality of developed applications.
4. Mini-project implementation and presentation shall be considered for assessment.

Guidelines for Laboratory Conduction

The laboratory shall be equipped with computing systems having adequate processing capability, memory, storage, and internet connectivity. Practical assignments shall be implemented using React.js, Node.js, Express.js, MongoDB, JavaScript, and supporting development tools. Students should use appropriate IDEs, database tools, and version-control systems for application development

and testing. Open-source software and datasets are recommended for experimentation and mini-project implementation. All the assignments should be conducted on multicore hardware and 64-bit open-source software.

List of Assignments

1. Study of MERN stack architecture, development workflow, and application structure. --CO1
2. Develop reusable React components using props, state, and event handling. --CO1
3. Design a single-page application using React Router for client-side navigation. --CO1
4. Create and validate data-entry forms using React and JavaScript validation techniques. --CO1, CO2
5. Develop RESTful APIs using Node.js and Express.js for data processing operations. --CO2
6. Design and implement MongoDB collections and perform CRUD operations using Mongoose. --CO3
7. Integrate React frontend with Node.js backend and MongoDB database. --CO2, CO3, CO4
8. Implement user authentication and session management in a MERN application. --CO2, CO4
9. Develop a Dataset Management Portal for storing, searching, and managing AI/ML datasets. --CO1, CO2, CO3, CO4, CO5
10. Develop an AI/ML Project Repository System for maintaining project information, documentation, and resources using the MERN stack. --CO1, CO2, CO3, CO4, CO5

Learning Resources

Textbooks:

1. Greg Lim, Beginning MERN Stack Development.
2. Shama Hoque, Full Stack React Projects.

Reference Books:

1. Ethan Brown, Web Development with Node and Express.
2. Alex Banks and Eve Porcello, Learning React.
3. Brad Dayley, Node.js, MongoDB and Angular Web Development.
4. Simon Holmes, Getting MEAN.

Software Requirements:

The laboratory should be equipped with Windows or Linux operating systems and appropriate development tools for MERN stack application development. Students shall use Visual Studio Code as the integrated development environment. Node.js and Express.js shall be used for server-side application development, while React.js shall be used for front-end development. MongoDB Community Edition shall be used for database management and data storage. Git and GitHub may be used for version control, source code management, and collaborative development activities.

Savitribai Phule Pune University
Third Year of Artificial Intelligence and Machine Learning (2024 Course)
Course Code: **ELC-381-AIM**
Course Name: **Internship/OJT**

Teaching Scheme	Credits	Examination Scheme
Practical: 8Hrs./wk	4	TW: 50

Prerequisites: Programming and Problem Solving, Data Structures and Algorithms, Fundamentals of Artificial Intelligence and Machine Learning.

Course Objectives:

Internship should make the student attain skills like:

1. To provide opportunities for students to gain professional experience through Internship/OJT.
2. To apply academic knowledge and technical skills in real-world industrial, research, or entrepreneurial environments.
3. To familiarize students with contemporary tools, technologies, and development practices.
4. To develop professional skills such as teamwork, communication, leadership, and problem-solving.
5. To apply internship learning towards project work, innovation, and career development.
6. To nurture professional ethics, workplace discipline, and social responsibility.

Course Outcomes:

On completion of the Internship/OJT, learner will be able to–

CO1: Apply academic knowledge and technical skills in professional environments.
CO2: Analyze industrial problems, workflows, and technical requirements.
CO3: Utilize appropriate tools, technologies, and platforms to accomplish assigned tasks.
CO4: Evaluate technical solutions considering professional, societal, and ethical responsibilities.
CO5: Develop professional documentation, reports, and presentations based on internship experience.

Guidelines

Internship/OJT provides practical exposure to industrial, research, innovation, entrepreneurial, government, or social environments. The objective is to bridge the gap between academic learning and professional practice.
Duration: Internship/OJT shall be completed after Semester V and before commencement of Semester VI.
Minimum Duration: 4 Weeks
Internship Work Identification: Students may undergo Internship/OJT through:
• Industry Internship
• AI/ML, Data Science, Cloud, Cyber Security, Software Development Projects
• Research Internship
• Startup and Entrepreneurship Activities
• Innovation and Product Development
• Open-Source Development
• Government, NGO, or Rural Internship
• Hackathons, Innovation Challenges, and Technical Competitions
Internship Diary / Workbook:
Students shall maintain an Internship Diary/Workbook containing:

- Daily activities performed
- Technologies and tools used
- Learning outcomes achieved
- Observations and reflections
- The Diary/Workbook shall be certified by the Industry Supervisor/Faculty Mentor.

Guidelines for Internship Report

1. Each student shall submit a Internship report in the prescribed format duly certified by the Internal /External Mentor and Head of the Department.
2. **The report shall contain**: The Internship Report shall include:

- Cover Page
- Internship Completion Certificate

- Organization Profile
- Acknowledgement
- Table of Contents
- Introduction
- Objectives of Internship
- Work Carried Out
- Tools and Technologies Used
- Results and Learning Outcomes
- Conclusion
- References
- Supervisor Feedback
- Attendance Record

1. **Internship Work Evaluation:**
   Assessment shall be carried out by the Faculty Mentor/Internship Coordinator based on internship performance, diary/workbook, report, and presentation

Guidelines for TW Assessment (50 Marks)

3.	Internship Diary / Workbook –	10 Marks
4.	Quality of Work Performed –	15 Marks
5.	Internship Report –	10 Marks
6.	Industry Supervisor Feedback –	10 Marks
7.	Presentation and Technical Discussion –	05 Marks
	Total: 50 Marks	

Post Internship Presentation:

Students shall deliver a presentation based on the internship work completed. The presentation shall cover:

- Organization and Project Overview
- Objectives and Scope
- Work Performed
- Technologies Used
- Outcomes and Learning Experience
- Challenges Faced and Solutions
- Future Scope

Reference Book:

- Ian Sommerville, Software Engineering, Pearson.
- Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow.

Textbook:

- C.R. Kothari and Gaurav Garg, Research Methodology: Methods and Techniques.

Reference Resources:

1. AICTE Internship Portal
2. NPTEL Employability Enhancement Courses
3. GitHub Open-Source Projects
4. Kaggle Learning Resources

Savitribai Phule Pune University
TE - Artificial Intelligence and Machine Learning (2024 Course)
National Education Policy (NEP)-2020 Compliant Syllabus

Team Members Behind Curriculum Structure Design

Members-BoS IT

Dr. Sudeep D. Thepade	Dr. Goraksha Garje
Dr. Aditya Abhyankar	Dr. Rajendra Sonar
Dr. Lalit Patil	Dr. R. Venkateswaran
Dr. Madhavi Pradhan	Dr. Maya Ingle
Dr. Poonam Lambhate	Dr. Abhijat Vichare
Dr. Abhijit Patankar	Dr. Surendra Mahajan
Dr. Madhuri Jawale	Dr. Abhijit Patankar

Course Coordinator - TE AIML

Dr. Ketaki Malgi
Coordinator, TE AI&ML Curriculum and Syllabus Development

Team Members for Course Content Design

Dr. Aarti Agarkar	Prof. Ashwini Kanade	Prof. Jayashree Jadhav	Dr. Ketaki A. Malgi
Dr. Meghana Lokhande	Dr. Minakshi Atre	Prof. Mugdha Rane	Prof. Pranoti Kale
Dr. Pushkar Joglekar	Dr. Ravindra Apare	Dr. Sandip Thite	Prof. Sandeep Samleti
Dr. Sayali Deshmukh	Dr. Shital Pawar	Dr. Swati Jadhav	Prof. Vinaya Kulkarni
Dr. Pradeep V. Jadhav	Dr. Deepak Vidhate	Dr. Abhijit Patankar	Dr. Sudeep D. Thepade

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