॥ योः क्रियावानू स पण्डितः ॥

SAVITRIBAI PHULE PUNE UNIVERSITY

(सावित्रीबाई फुले पुणे विद्यापीठ)

Formerly University of Pune

Syllabus Booklet

For

T.E.

(COMPUTER SCIENCE)

— 2024 PATTERN —

With effect from Academic Year 2026-27

Contents

Nomenclature

AEC Ability Enhancement Course

AICTE All India Council for Technical Education

CCE Comprehensive Continuous Evaluation

CEP Community Engagement Project

EEM Entrepreneurship Management

ETE End-semester Examination

MDM Multidisciplinary Minor

MOOC Massive Open Online Course

NEP National Educational Policy

NPTEL National Programme on Technology Enhanced Learning

OE Open Elective

PCC Program Core Course

PEO Programme Educational Objectives

PSO Program Specific Outcomes

SWAYAM Study Webs of Active-Learning for Young Aspiring Minds\\

VEC Value Education Course

VTE Vocational and Skill Enhancement Course

WK Knowledge and Attitude Profile

Preface by Board of Studies

Dear Students and Teachers,

We, the members of Board of Studies Computer Engineering, are very happy to present Third Year Computer Science syllabus effective from the Academic Year 2026-27. The present curriculum will be implemented for Third Year of Engineering from the academic year 2026-27. Subsequently this will be carried forward for BE in AY 2027-28, respectively.

Computer Science is a dynamic discipline that lies at the intersection of computer engineering and computer science. It provides the foundation for the design, development, and application of computer systems and other computing devices. This curriculum is designed to provide students with a comprehensive understanding of the fundamental principles, theories, and practices of Computer Science, while also preparing them for the ever-evolving technological landscape.

The revised syllabus falls in line with the objectives of NEP-2020, Savitribai Phule Pune University, AICTE New Delhi, UGC, and various accreditation agencies by keeping an eye on the technological developments, innovations, and industry requirements. Wherever possible additional resource links of platforms such as NPTEL, Swayam are appropriately provided at the end of each course. Learners can be advised to take up online courses, on successful completion they are required to submit certification for the same. This will definitely help learners to facilitate their enhanced learning based on their interest.

This curriculum is the result of extensive consultation with academic experts, industry professionals, and alumni to ensure relevance and excellence. It is designed not only to meet the current industry standards but also to prepare students for higher studies and research in the field of Computer Science.

We hope that this curriculum will inspire students to become competent professionals, responsible citizens, and contributors to the technological advancement of society.

Dr. Nilesh Uke
Chairman - Board of Studies
Computer Engineering - SPPU

Members of Board of Studies - Computer Science	
Dr. Pramod Patil	Dr. Dipti Patil
Dr. Dhananjay Kshirsagar	Dr. Amol Potgantwar
Dr. Sachin Babar	Dr. Balwant Sonkamble
Dr. Suhasini Itkar	Dr. Sachin Sakhare
Dr. Dipak Patil	Dr. Vandana Dhingra
Dr. Deepali Ujalambkar	Dr. Vaishali Vikhe
Dr. Pradip Jawandhiya	Dr. Sandeep Deshmukh

Third Year Computer Science - 2024 Pattern

Program Specific Outcomes (PSO)

- **PSO1:** Professional Skills-The ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient design of computer-based systems of varying complexities.
- **PSO2:** Problem-Solving Skills- The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.
- **PSO3:** Successful Career and Entrepreneurship- The ability to employ modern computer languages, environments and platforms in creating innovative career paths to be an entrepreneur and to have a zest for higher studies.

Programme Educational Objectives (PEO)

Program Educational Objectives (PEOs): Program Educational Objectives are broad statements that describe the career and professional accomplishments that the program is preparing graduates to achieve.

PEO	PEO Focus	PEO Statements
PEO1	Core competence	Attainment of key principles and practices of computation, mathematics and basic principles of engineering to ensure that graduates are able to apply their software development skills in design and implementation of practical systems consisting of software and/or hardware components.
PEO2	Problem solving skills and Ethics	Analyze real-life problems and impart science-based engineering education to develop professional skills that will prepare the students for immediate employment in the industry.
PEO3	Professionalism and Lifelong Learning	Imbibe lifelong learning, professional and ethical attitude for embracing global challenges and make positive impact on environment and society.

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 (GAPCV4.0) - (August 2024) Page 55.

Third Year Computer Science - 2024 Pattern

Program Outcomes (PO)

Program Outcomes are statements that describe what students are expected to know and be able to do upon graduating from the program. These relate to the skills, knowledge, attitude and behaviour that students acquire through the program. On successful completion of B.E. in Computer Science, graduating students/graduates will 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)

Reference: self-Assessment Report (SAR) Format Undergraduate Engineering Programs Graduate Attributes and Professional Competencies Version 4.0 (GAPC V4.0) - (August 2024) Page 56.

Third Year Computer Science - 2024 Pattern

General Rules and Guidelines

- **Course Outcomes (CO):** Course Outcomes are narrower statements that describe what students are expected to know, and are able to do at the end of each course. These relate to the skills, knowledge and behaviour that students acquire in their progress through the course.
- **Assessment:** Assessment is one or more processes, carried out by the institution, that identify, collect, and prepare data to evaluate the achievement of Program Educational Objectives and Program Outcomes.
- **Evaluation:** Evaluation is one or more processes, done by the Evaluation Team, for interpreting the data and evidence accumulated through assessment practices. Evaluation determines the extent to which Program Educational Objectives or Program Outcomes are being achieved, and results in decisions and actions to improve the program

Guidelines for Examination Scheme

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

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:

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

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

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

Format and Implementation of Comprehensive Continuous Evaluation (CCE)

- **Unit Test**
  - **Format:** Questions designed as per Bloom's Taxonomy guidelines to assess various cognitive levels (Remember, Understand, Apply, Analyze, Evaluate, Create).
  - **Implementation**: Schedule the test after completing Units 1 and 2. Ensure the question paper is balanced and covers key concepts and applications.
- **Sample Question Distribution**
  - Remembering (2 Marks): Define key terms related to [Topic from Units 1 and 2].
  - Understanding (2 Marks): Explain the principle of [Concept] in [Context].
  - Applying (2 Marks): Demonstrate how [Concept] can be used in [Scenario].
  - Analyzing (3 Marks): Compare & contrast [Two related concepts] from Units 1 and 2.
  - Evaluating (3 Marks): Evaluate the effectiveness of [Theory/Model] in [Situation].
- **Assignments / Case Study:** Students should submit one assignment or one Case Study Report based on Unit 3 and one assignment or one Case Study Report based on Unit 4.
  - **Format:** Problem-solving tasks, theoretical questions, practical exercises, or case studies that require in-depth analysis and application of concepts.
  - **Implementation:** Distribute the assignments or case study after covering Units 3 and 4. Provide clear guidelines and a rubric for evaluation.
- **Seminar Presentation:**
  - **Format:** Oral presentation on a topic from Unit 5, followed by a Q&A Session.
  - **Deliverables:** Presentation slides, a summary report in 2 to 3 pages, and performance during the presentation.
  - **Implementation:** Schedule the Seminar presentations towards the end of the course. Provide students with ample time to prepare and offer guidance on presentation skills.
- **Open Book Test:**
  - **Format:** Analytical and application-based questions to assess depth of understanding.
  - **Implementation:** Schedule the open book test towards the end of the course, ensuring it covers critical aspects of Unit 5.
- **Quiz:**
  - **Format:** Quizzes can help your students practice existing knowledge while stimulating interest in learning about new topic in that course. You can set your quizzes to be completed individually or in small groups.
  - **Implementation:** Online tools and software can be used create quiz. Each quiz is made up of a variety of question types including multiple choice, missing words, true or false etc.

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

Evaluation and Feedback:

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

End-Semester Examination (ESE)

End-semester Examination (ETE) 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 ETE at the end of the semester.

- **Format and Implementation:**

  - **Question Paper Design:** Below structure is to be followed to design an End-semester Examination (ETE) 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, Analyze, Evaluate, and Create. The questions should be structured to cover:
    - Remembering: Basic recall of facts and concepts.
    - Understanding: Explanation of ideas or concepts.
    - Applying: UTE of information in new situations.
    - Analyzing: Drawing connections among ideas.
    - Evaluating: Justifying a decision or course of action.
    - Creating: Producing new or original work (if applicable).
  - **Detailed Scheme for 70 Marks:** Unit-Wise Allocation (14 Marks per Uit): 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' understand- ing and application of the course material, adhering to Bloom's Taxonomy guidelines for cognitive skills evaluation.
  - **Detailed Scheme for 35 Marks:** Unit-Wise Allocation (08 Marks for Unit 1 , 09 Marks for Unit 2, Unit 3 and Unit 4) : 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.

***

Third Year Engineering (2024 Pattern) – Computer Science

Level 5.0

Course Code	Course Type	Course Name	Teaching Scheme			Examination Scheme						Credits			
			Theory	Tutorial	Practical	CCE	EndSem	Term Work	Practical	Oral	Total	Theory	Tutorial	Practical	Total
PCC-301-CSC	Program Core Course	Artificial Intelligence	3	-	-	30	70	-	-	-	100	3	-	-	3
PCC-302-CSC	Program Core Course	Computer Network	3	-	-	30	70	-	-	-	100	3	-	-	3
PCC-303-CSC	Program Core Course	Theory of Computation	3	-	-	30	70	-	-	-	100	3	-	-	3
PCC-304-CSC	Program Core Course	Artificial Intelligence Laboratory	-	-	2	-	-	50	25	-	75	-	-	1	1
PCC-305- CSC	Program Core Course	Computer Network Laboratory	-	-	4	-	-	25	-	25	50	-	-	2	2
PEC-321-CSC	Program Elective	Elective - I	3	-	-	30	70	-	-	-	100	3	-	-	3
PEC-322-CSC	Program Elective	Elective-I Laboratory	-	-	2	-	-	50	-	-	50	-	-	1	1
MDM-331- CSC	Multidisciplinary Minor	Robotics and Automation	-	1	4	-	-	50	-	-	50	-	1	2	3
OLE-341-CSC	Open Elective	Open Elective*	2	-	-	15	35	-	-	-	50	2	-	-	2
VEC-232- CSC	Vocational Skills Education	Technical Seminar	-	-	2	-	-	-	-	25	25	-	-	1	1
Total			**14**	**1**	**14**	**135**	**315**	**175**	**25**	**50**	**700**	**14**	**1**	**7**	**22**

* Note: Students can opt for Open Elective offered by different discipline/faculty like arts, science , commerce, management, humanities or interdisciplinary studies. Example - IPR and cyber laws, sustainability development , digital personal data protection or constitution of India

Elective I Course:

- PEC-321A-CSC: Elective I – Information Retrieval
- PEC-321B-CSC: Elective I-Cloud Computing
- PEC-321C-CSC: Elective I- User Interface and User Experience Design (UI/UX)
- PEC-321D-CSC: Elective-I-Embedded System.

Curriculum Structure - Semester VI

Third Year Engineering (2024 Pattern) – Computer Science

Level 5.0

Course Code	Course Type	Course Name	Teaching Scheme			Examination Scheme						Credits			
			Theory	Tutorial	Practical	CCE	EndSem	Term Work	Practical	Oral	Total	Theory	Tutorial	Practical	Total
PCC-351- CSC	Program Core Course	Machine Learning	3	-	-	30	70	-	-	-	100	3	-	-	3
PCC-352 CSC	Program Core Course	Software Engineering and Modeling	2	-	-	30	70	-	-	-	100	2	-	-	2
PCC-353-CSC	Program Core Course	Machine Learning Laboratory	-	-	4	-	-	50	25	-	75	-	-	2	2
PCC-354 CSC	Program Core Course	Software Engineering and Modeling Laboratory	-	-	2	-	-	25	-	25	50	-	-	1	1
PEC-361-CSC	Program Elective Course	Elective - II	3	-	-	30	70	-	-	-	100	3	-	-	3
PEC-362-CS	Program Elective Course	Elective - III	3	-	-	30	70	-	-	-	100	3	-	-	3
PEC-363-CSC	Program Elective Course	Elective-III Laboratory	-	-	2	-	-	25	-	-	25	-	-	1	1
MDM-371- CSC	Multidisciplinary Minor	Green Computing	-	1	2	-	-	25	-	25	50	-	1	1	2
VSE-372- CSC	Value and Skill Enhancement	Full Stack Web Development	-	-	2	-	-	50	-	-	50	-	-	1	1
ELC-381-CSC	Entrepreneurship / Economics / Management	Internship/OJT	-	-	8	-	-	25	-	25	50	-	-	4	4
Total			**11**	**1**	**20**	**120**	**280**	**200**	**25**	**75**	**700**	**11**	**1**	**10**	**22**

Elective-II Courses:

- PEC-361A- CSC: Elective II -Soft Computing and Fuzzy Logic
- PEC-361B- CSC: Elective II - Augmented Reality and Virtual Reality (AR/VR)
- PEC-361C- CSC: Elective II- Natural Language Processing
- PEC-361D- CSC: Elective II-Quantum Computing

Elective-III Courses:

- PEC-362A- CSC: Elective III – Data Analytics and Visualization
- PEC-362B- CSC: Elective III -Cryptography and Network Security
- PEC-362C- CSC: Elective III -Block Chain Technology
- PEC-362D- CSC: Elective III - Image Processing.

Savitribai Phule Pune University, Pune

Maharashtra, India

TE - Computer Science

2024 Pattern

Semester V

With effect from Academic Year 2026-27

Savitribai Phule Pune University **Third Year of Computer Science (2024 Course)**		
PCC-301-CSC: Artificial Intelligence		
Teaching scheme	**Credits**	**Examination Scheme**
Theory : 03Hours/Week	03	**CCE:** 30 Marks **End-Semester:** 70 Marks

Prerequisite Courses, if any :

1. Fundamentals of Programming Languages (ESC-105-COM)
2. Programming and Problem Solving (PCC-151-ITT)
3. Probability and Statistics(PCC-253-CSC)

Companion Course: Artificial Intelligence Laboratory

Course Objectives: The course aims:

1. To introduce fundamental concepts of Artificial Intelligence, intelligent agents, and ethical considerations in AI.
2. To develop the ability to model and solve problems using state-space search and heuristic algorithms.
3. To enable students to design solutions for adversarial and constraint satisfaction problems using appropriate AI techniques.
4. To impart knowledge representation skills using propositional and first-order logic with inference mechanisms.
5. To equip students with AI planning techniques and relate them to real-world industrial applications.

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

- **CO1:** Explain fundamental concepts of Artificial Intelligence, Agents and ethical implications of AI in real-world scenarios.
- **CO2:** To formulate real-world problems as state-space search models and implement appropriate heuristic, local, and online search algorithms.
- **CO3:** To model real-world competitive and constraint-based problems and implement adversarial search algorithms and CSP techniques
- **CO4:** To represent real-world knowledge using propositional and first-order logic and apply Inference techniques and derive logical conclusions in knowledge-based AI systems.
- **CO5:** To apply AI planning techniques for solving classical planning problems and Relate them to real-world industrial AI application.

Course Contents

Unit I – Introduction to AI and Intelligent Agents (07 Hours)

Introduction to Artificial Intelligent: Introduction to Artificial Intelligence, Foundations of Artificial Intelligence, History of Artificial, Intelligence, Limits of AI, Ethics of AI, Future of AI, AI Components, AI Architectures, Intelligent Agents, Agents and Environments, Good Behavior: Concept of Rationality, Types of Agents, Nature of Environments, Structure of Agents.

Case Study: Autonomous Taxi Agent – Waymo One, AI in Healthcare – IBM Watson for Oncology

Mapping of Course Outcomes for Unit I: CO1

Unit II – Problem Solving: State Approach and Search Strategies (07 Hours)

State Space Search: Tower of Hanoi. Informed (Heuristic) Search Strategies: Introduction to Greedy BFS, A* Search, Iterative-deepening, Heuristic Functions. Local Search and Optimization Problems: Hill-climbing search, Simulated annealing, Local beam search. Online Search Agents and Unknown Environments: Online search problems.

Case Study: Warehouse robots (Amazon Kiva) and self-driving cars, Logistics and Routing: Traveling Salesman Problem, Google DeepMind – AI for Energy Efficiency in Data Centers

***Mapping of Course Outcomes for Unit II:** CO2

Unit III – Adversarial Search, and Game Theory (07 Hours)

Optimal Decisions in Games, Heuristic Alpha–Beta Tree Search, Monte Carlo Tree Search, Stochastic Games, Partially Observable Games, Limitations of Game Search Algorithms, Constraint Satisfaction Problems (CSP), Constraint Propagation: Inference in CSPs, Backtracking Search for CSPs

Case study: AlphaGo – AI in Strategic Board Games, Strategic Decision-Making in Imperfect Information Games-- Libratus, Adversarial Search and Constraint Reasoning in Computer Chess - IBM deep blue

***Mapping of Course Outcomes for Unit III:** CO1, CO3

Unit IV – Knowledge representation using Logical Formalisms, Propositional & First Predicate calculus (08 Hours)

Introduction to Logical Formalisms: Role of logic in Artificial Intelligence, Knowledge-based agents Syntax and semantics of logical systems, Propositional Logic – Basics and Inference: Propositional symbols and well-formed formulas, Logical connectives, Inference rules: Modus Ponens, Modus Tollens, Resolution, First Order Predicate Logic – Fundamentals: Motivation for First Order Logic, Quantifiers (∀, ∃), Well-formed formulas, Translating natural language into FOL Inference in First Order Predicate Logic – Fundamentals: Motivation for First Order Logic, Quantifiers (∀, ∃), Well-formed formulas, Translating natural language into FOL

Case study: Medical Expert System – MYCIN, Knowledge-Based Reasoning in Intelligent Search Systems:- AI-Based Rule Engine in Google Search

***Mapping of Course Outcomes for Unit IV:** CO4

Unit V – Planning and Industrial Application of AI (07 Hours)

Planning: Overview, An example Domain The Blocks world, The components of planning system, Goal stack planning, Nonlinear planning using constraint posting, Hierarchical planning, Industrial Application of AI : AI in Healthcare, AI in Finance, AI in Retail, AI in Agriculture, AI in Education, AI in Transportation, AI in Experimentation and Multi-disciplinary research

Case study: AI-Driven Supply Chain & Production Planning (Manufacturing)

***Mapping of Course Outcomes for Unit V:** CO5

Learning Resources

Text Books:

1. Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach", Third edition, Pearson, 2003, ISBN :10: 0136042597
2. Elaine Rich,Kevin Knight and Nair,"Artificial Intelligence",TMH,ISBN-978-0-07-008770-5
3. Saptarsi Goswami, Amit Kumar Das and Amlan Chakrabarti, "AI for Everyone – A Beginner's Handbook for Artificial Intelligence", Pearson, 2024
4. Deepak Khemani, "A First Course in Artificial Intelligence", McGraw Hill Education(India), 2013, ISBN 978-1-25-902998-1,

Reference Books:

1. Patrick Henry Winston, "Artificial Intelligence", Addison-Wesley Publishing Company, ISBN: 0-201-53377-4
2. Dr.Nilakshi Jain,"Artificial Intelligence,As per AICTE: Making a System Intelligent",Wiley publication, ISBN: 9788126579945
3. Nilsson Nils J , "Artificial Intelligence: A new Synthesis", Morgan Kaufmann Publishers Inc. San Francisco, CA, ISBN:978-1-55-860467-4.
4. Dr. Lavika Goel, "Artificial Intelligence: Concepts and Applications", Wiley publication, ISBN:97881265

E-Books: -

1. http://repo.darmajaya.ac.id/5094/1/Lecture-AI.pdf
2. https://www.freebookcentre.net/ComputerScience-Books-Download/Digital-notes-on-Artificial-Intellig
3. https://www.kdnuggets.com/10-free-artificial-intelligence-books-for-2025

MOOC / NPTEL/YouTube Links: -

1. Artificial Intelligence: Knowledge Representation And Reasoning By Prof. Deepak Khemani, IIT Madras https://onlinecourses.nptel.ac.in/noc26_cs63/preview
2. An Introduction to Artificial Intelligence By Prof. Mausam, IIT Delhi https://nptel.ac.in/courses/1061022https://nptel.ac.in/courTEs/106/106/10610612

Savitribai Phule Pune University **Third Year of Computer Science (2024 Course)**		
PCC-302-CSC: Computer Network		
Teaching scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	**CCE:** 30 Marks **End-Semester :** 70 Marks

Prerequisite Courses:

Digital Electronics and Logic Design (MDM-221- CSC), Discrete Mathematics (PCC-203-CSC), Internet of Things(MDM-271-CSC)

Companion Course: Computer Network Laboratory.

Course Objectives: The course aims to:

1. Introduce fundamental concepts of networking, hardware, software, and reference models.
2. Develop knowledge of physical, data link, network, transport, and application layers with emphasis on design issues, services, and protocols.
3. Equip students with the ability to analyze and compare routing, error control, congestion control, and quality of service mechanisms.
4. Familiarize students with widely used protocols such as TCP, UDP, IP, HTTP, DNS, SMTP, and emerging technologies in multimedia and wireless networks.
5. Strengthen problem-solving skills through case studies and exemplars that connect theoretical concepts to real-world applications.

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

- **CO1: Explain** the fundamental concepts of computer networks, layered architecture, and physical transmission media.
- **CO2: Apply** error detection, correction, and reliable data transfer techniques at the data link layer.
- **CO3: Analyze** routing algorithms, addressing schemes, and Internet protocols at the network layer.
- **CO4: Compare** and **evaluate** transport layer protocols (TCP, UDP, SCTP, RTP) and mechanisms for congestion control and QoS.
- **CO5: Demonstrate** understanding of application layer protocols (HTTP, DNS, Email, FTP, TEL NET, DHCP, SNMP) and emerging network applications.

Course Contents

Unit I – Introduction to Computer Network and Physical Layer (07 Hours)

Introduction to computer networks; uses of computer networks – business applications, home applications, mobile users; network hardware – PAN, LAN, MAN, WAN and internetworks; network software – protocol hierarchies, design issues for layers, connection-oriented and connection-less services, service primitives, relationship between services and protocols; reference models – OSI and TCP/IP models. Physical Layer: guided transmission media; wireless transmission; telephone system; narrowband and broadband communication systems.

Case Study: Comparison of OSI vs TCP/IP in real-world networks, case study on broadband vs narrowband communication in India

***Mapping of Course Outcomes for Unit I:** CO1

Unit II – Data Link Layer (07 Hours)

Data Link Layer – services provided to the network layer, framing and addressing; design issues of the data link layer – error control, flow control and reliable data transfer; error detection and correction techniques – parity, checksum, CRC and basic error correction concepts; data link layer protocols – elementary protocols and Stop-and-Wait protocol; sliding window protocols – pipelining, Go-Back-N and Selective Repeat protocols; example data link layer technologies – packet over SONET

Case Study: Case study on CRC error detection in Ethernet, SONET backbone deployment in telecom networks.

***Mapping of Course Outcomes for Unit II:** CO2, CO3

Unit III – Network Layer (08 Hours)

Network Layer – Services & Design Issues: connectionless service, connection-oriented service, QoS support, error control, flow control, store-and-forward switching, congestion control, reliability, internetworking challenges; Routing Algorithms: shortest path routing, flooding, distance vector routing, link state routing; Internet Architecture & Protocols: IP addressing (IPv4 classes, CIDR, subnetting), IPv4 vs IPv6, IP datagram, ICMP, ARP, RIP, OSPF

Case Study: IPv6 adoption in ISPs, routing comparison between RIP and OSPF in enterprise networks.

***Mapping of Course Outcomes for Unit III:** CO2, CO3

Unit IV- Transport Layer Protocols (07 Hours)

Process to Process Delivery, Services, Socket Programming. Elements of Transport Layer Protocols: Addressing, Connection establishment, Connection release, Flow control and buffering, Multiplexing, Congestion Control. Transport Layer Protocols: TCP and UDP, SCTP, RTP, Congestion control and Quality of Service (QoS), Differentiated services, TCP and UDP for Wireless networks.

Case Study: TCP congestion control in 4G/5G networks, RTP in video conferencing applications.

***Mapping of Course Outcomes for Unit IV:** CO4

Unit V- Application Layer (08 Hours)

Introduction – principles of application layer, client-server and peer-to-peer models; Web and HTTP – request/response, persistent vs. non-persistent connections, cookies, caching, performance; DNS – hierarchy, resource records, name resolution, caching, security issues; Email – SMTP, MIME, POP3, IMAP, webmail, message format, security; FTP – basics, file transfer process, legacy relevance; TELNET – remote login, limitations, legacy use; DHCP – dynamic host configuration, IP allocation, management; SNMP – network management, monitoring, MIBs, security considerations; Emerging Topics – multimedia applications, RTP for streaming, peer-to-peer applications, cloud-based services.

Case Study: DNS security attacks (DNS spoofing), HTTP caching in CDNs, SMTP in enterprise email systems.

***Mapping of Course Outcomes for Unit:** V CO5

Learning Resources

Text Books:

1. Andrew S. Tanenbaum, David J. Wetherall – Computer Networks, 5th Edition, Pearson Education.
2. Behrouz A. Forouzan – Data Communications and Networking, 5th Edition, McGraw Hill.
3. William Stallings – Data and Computer Communications, 10th Edition, Pearson Education.

Reference Books:

1. Kurose, James F., Ross, Keith W. – Computer Networking: A Top-Down Approach, 8th Edition, Pearson.
2. Peterson, Larry L., Davie, Bruce S. – Computer Networks: A Systems Approach, 5th Edition, Morgan Kaufmann.

MOOC / NPTEL/YouTube Links: -

1. nptel.ac.in/courses/106/105/106105183
2. nptel.ac.in/courses/106/105/106105080
3. nptel.ac.in/courses/106/105/106105081
4. nptel.ac.in/courses/106/106/106106091
5. nptel.ac.in/courses/106/105/106105031
6. https://www.mooc-list.com/tags/computer-networking
7. https://www.coursera.org/courses?query=computer%20network

E-Books :

1. https://people.cs.clemson.edu/~jmarty/courses/kurose/KuroseCh1-2.pdf
2. http://eti2506.elimu.net/Introduction/Books/
3. http://intronetworks.cs.luc.edu/current/ComputerNetworks.pdf

Savitribai Phule Pune University **Third Year of Computer Science (2024 Course)**		
PCC-303-CSC: Theory Of Computation		
Teaching scheme	**Credits**	**Examination Scheme**
Theory : 03Hours/Week	03	**CCE:** 30 Marks **End-Semester**: 70 Marks

Prerequisite Courses: Discrete Mathematics (PCC-203-CSC)

Course Objectives:

1. **Understand** the fundamental concepts of formal languages, automata theory, and computability.
2. **Analyze** the properties of regular languages, including closure and decision properties.
3. **Analyze** the properties of context-free languages and grammars, including parsing and ambiguity concepts.
4. **Construct** pushdown automata and context-free grammars for recognizing and generating context-free languages
5. **Evaluate** the computational capabilities and limitations of Turing machines as models of computation and computability

Course Outcomes: on completion of the course, learners should be able to:

- **CO1: Design** finite automata for formal language recognition.
- **CO2: Construct** Deterministic and Non deterministic Finite Automata for given regular expressions.
- **CO3: Evaluate** context-free grammars in terms of derivation, ambiguity, simplification, normal forms.
- **CO4: Design** pushdown automata for context-free languages.
- **CO5: Build** Turing machines for language recognition and computational problem solving

Course Contents

Unit I – Formal Language Theory and Finite Automata (07 Hours)

Introduction to Formal language, introduction to language translation logic, Essentials of translation, Alphabets and languages, Finite representation of language, Finite Automata (FA): An Informal Picture of FA, Finite State Machine (FSM), Language accepted by FA, Definition of Regular Language, Deterministic and Nondeterministic FA(DFA and NFA), epsilon- NFA, FA with output: Moore and Mealy machines -Definition, models, inter-conversion

Case study: FSM for vending machine, spell checker.

***Mapping of Course Outcomes for Unit I**: CO1

Unit II – Regular Expressions (RE) (07 Hours)

Operators of RE, Building RE, Precedence of operators, Algebraic laws for RE, Conversions: NFA to DFA, RE to DFA Conversions, DFA to RE Conversions: State/loop elimination, Arden's theorem, Properties of Regular Languages: Closure properties, Pumping Lemma for Regular languages, Decision properties.

Cast Study: RE in text search and replace

***Mapping of Course Outcomes for Unit II:** CO2

Unit III – Context Free Grammars (CFG) and Languages (07 Hours)

Introduction, Regular Grammar, Context Free Grammar- Definition, Derivation, Language of grammar, sentential form, parse tree, inference, derivation, parse trees, ambiguity in grammar and Language-ambiguous Grammar, Simplification of CFG: Eliminating unit productions, useless production, useless symbols, and Є-productions, Normal Forms- Chomsky normal form, Greibach normal form, Closure properties of CFL, Decision properties of CFL, Chomsky Hierarchy, Application of CFG: Parser, Markup languages, XML and Document Type Definitions.

Case Studies – CFG for Palindromes, Parenthesis Match

***Mapping of Course Outcomes for Unit III:** CO3

Unit IV – Pushdown Automata(PDA) (07 Hours)

Basic Definitions, Equivalence of Acceptance by Finite State & Empty stack, PDA & Context Free Language, Equivalence of PDA and CFG, Parsing & PDA: Top-Down Parsing, Top-down Parsing Using Deterministic PDA, Bottom-up Parsing, Closure properties and Deterministic PDA

Case study: Balanced Parentheses Checking in Compilers

***Mapping of Course Outcomes for Unit IV:** CO4

Unit V – Turing Machines (TM) (07 Hours)

Turing Machine Model, Representation of Turing Machines, Language Acceptability by Turing Machines, Design of TM, Description of TM, Techniques for TM Construction, Variants of Turing Machines, The Model of Linear Bounded Automata, TM & Type 0 grammars, TM's Halting Problem

Case Studies - Binary Addition System, string reversal

***Mapping of Course Outcomes for Unit V:** CO5

Learning Resources

Text Books:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman, —Introduction to Automata Theory Languages and Computation||, Addison-Wesley, ISBN 0-201-44124-1.
2. H.L. Lewis, Christos H. Papadimitriou, —Elements of the Theory of Computation||, Prentice Hall, ISBN-10: 0132624788; ISBN-13: 978-0132624787.

Reference Books:

1. John Martin, —Introduction to Languages of The Theory of Computation||, 2nd Edition, Mc Graw Hill Education, ISBN-13: 978-1-25-900558-9, ISBN-10: 1-25-900558-5
2. Sanjeev Arora and Boaz Barak, —Computational Complexity: A Modern Approachl , Cambridge University Pre ss, ISBN: 0521424067 9780521424264
3. Daniel Cohen, —Introduction to Computer Theory||, Wiley & Sons, ISBN 9788126513345.
4. J. Carroll & D Long, —Theory of Finite Automata||, Prentice Hall, ISBN 0-13-913708-4.
5. Kavi Mahesh, —Theory of Computation : A Problem-Solving Approach||, Wiley India, ISBN10 8126533110.
6. Michael Sipser, —Introduction to the Theory of Computation||, Cengage Learning, ISBN-13: 9781133187813.
7. Vivek Kulkarni —Theory of Computation||, Oxford University Press, ISBN 0-19-808458

E-Books:

1. https://www-2.dc.uba.ar/staff/becher/Hopcroft-Motwani-Ullman-2001.pdf

MOOC/NPTEL/SWAYAM/YouTube Course Links:

1. https://onlinecourses.nptel.ac.in/noc21_cs83/preview
Theory of Computation By Prof. Raghunath Tewari | IIT Kanpur

2. https://onlinecourses.nptel.ac.in/noc26_cs85/preview
Theory of Computation By Prof. Subrahmanyam Kalyanasundaram | IIT Hyderabad

Savitribai Phule Pune University **Third Year of Computer science (2024 Course)**		
PCC-304-CSC: Artificial Intelligence Laboratory		
Teaching scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	**Practical**: 01	**Term Work:** 50 Marks **Practical :** 25 Marks

Companion Course: Artificial Intelligence

Course Objectives:

1. To provide a comprehensive understanding of the fundamental concepts, principles, and methodologies of Artificial Intelligence.
2. To equip students with the ability to design and implement core AI algorithms for problem solving, search, reasoning, and learning.
3. To develop analytical and practical skills for modeling and solving real-world problems using appropriate AI techniques.

Course Outcomes:

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

- **CO1: Explain** and differentiate fundamental concepts, components, and techniques of Artificial Intelligence, including intelligent agents and AI methodologies.
- **CO2: Design** and implement AI algorithms for state-space search, heuristic search, reasoning, and basic learning tasks using programming tools.
- **CO3: Model** real-world problems as AI problems and **apply** appropriate AI techniques to obtain optimal or near-optimal solutions.
- **CO4: Apply** AI planning techniques for solving solve classical planning problems and relate them to real-world industrial AI application.

Course Contents

Guidelines for Instructor's Manual

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

Guidelines for Student's Laboratory Journal

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

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

Guidelines for Laboratory/Term Work Assessment

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

Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technolog- ical aspects, utility and recent trends related to the topic. The assignment framing policy needs to address the average students and inclusive of an element to attract and promote the intelligent stu- dents. 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. Encourage stu- dents for appropriate use of Hungarian notation, proper indentation and comments. UTE of open source software is to be encouraged. In addition to these, instructors may assign one real life ap- plication in the form of a mini-project based on the concepts learned. Instructors may also set one assignment or mini-project that is suitable to respective branch beyond the scope of the syllabus.

Operating System recommended: - 64-bit Open source Linux or its derivative
Programming tools recommended: - Open Source Python, C, C++

Guidelines for Practical Examination

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

Sr. No	List of Laboratory Experiments/Assignments (Any 04 from the given list)
1	To design and implement a simple reflex agent for the Vacuum Cleaner world and analyze its rational behavior in a given environment.
2	To implement a model-based intelligent agent capable of navigating a grid environment with obstacles using internal state representation.
3	Implement A* Search for 8 puzzle problem.
4	Implement Adversarial Search with Alpha-Beta pruning
5	Implement Tower of Hanoi by State Space Search
6	To implement a Sudoku solver using backtracking and constraint propagation techniques
7	Implement Local Beam Search for Traveling Salesman Problem.

Sr. No.	List of Assignment (Part B)- 8,9,10 mandatory and Any Four from 11 to 14

8	Use BSF to solve a planning problem. Examples can be 8 puzzle problem, Robot path planning ,Blocks world etc.
9	Use DFS for problem solving. Examples can be water jug problem , Missionaries and Cannibals ,maze solving, blocks world etc.
10	To implement Backtracking search to solve the 8-Queens constraint satisfaction problem.
11	To develop a simple chatbot using predefined rules and pattern matching.
12	To design an intelligent agent that guesses a number using feedback (higher/lower).
13	To implement Goal Stack Planning for solving the Blocks World problem and achieve a specified goal configuration
14	To develop a game-playing agent using the Minimax algorithm for optimal decision-making.
Sr. No.	**List of Assignments (Part C)- Mini-Project (In Team of 3-4 Students)- Any ONE**
15	Build unbeatable Tic-Tac-Toe AI using basic minimax to depth 9. Play 10 games vs random opponent
16	Medical Diagnosis System (Rule-Based Expert System) using forward and backward chaining

Savitribai Phule Pune University Third Year of Computer science (2024 Course)		
PCC-305-CSC: Computer Network Laboratory		
Teaching scheme	**Credits**	**Examination Scheme**
Practical : 04 Hours/Week	**Practical**: 02	**Term Work:** 25 Marks **Oral :** 25 Marks

Prerequisites Courses: Digital Electronics and Logic Design (MDM-221- CSC), Discrete Mathematics (PCC-203-CSC), Internet of Things(MDM-271-CSC)

Companion Course: Computer Network

Course Objectives:

1. Provide hands-on experience with computer networking concepts.
2. Familiarize students with network models, protocols, and tools.
3. Develop skills in analyzing and simulating network behavior.
4. Encourage practical implementation of routing, addressing, and error detection techniques.
5. Build competence in using tools like Packet Tracer and Wireshark for network configuration and analysis.

Course Outcomes:

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

- **CO1: Compare** and **analyze** OSI and TCP/IP reference models, topologies, transmission media, and LAN/WAN configurations using simulation tools.
- **CO2: Apply** error detection and correction techniques, subnetting, and IP addressing schemes to design efficient and reliable networks.
- **CO3: Simulate** and **implement** transport layer protocols (Go-Back-N, Selective Repeat, TCP/UDP sockets) and client-server applications such as file transfer and DNS lookup.
- **CO4: Configure** and **evaluate** routing protocols and services while capturing and analyzing traffic.
- **CO5:** Work collaboratively on a mini-project to **design**, **implement,** and **analyze** a networking solution demonstrating innovation, teamwork, and professional skills in laboratory practice.

Course Contents

Guidelines for Instructor's Manual

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

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis). Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal must be avoided. Use of DVD containing student's programs maintained by

Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

Continuous assessment of laboratory work should be based on overall performance of Laboratory assignments by a student. Each Laboratory assignment assessment will assign grade/marks based on

Guidelines for Laboratory/Term Work Assessment

parameters, such as timely completion, performance, innovation, efficient codes, and punctuality.

Guidelines for Laboratory Conduction

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

Guidelines for Practical Examination

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

Sr. No.	List of Laboratory Experiments/Assignments (Part A)- 1,2 mandatory and Any Three from 3 to 7
1	Case study on comparison of OSI and TCP/IP reference models with respect to layers, functions, protocols, and real-world implementation. (Compulsory)
2	Demonstrate the different types of topologies and types of transmission media by using a packet tracer tool. (Compulsory)
3	Setup a WAN which contains wired as well as wireless LAN by using a packet tracer tool. Demonstrate transfer of a packet from LAN 1 (wired LAN) to LAN2 (Wireless LAN).
4	Capture and analyze Ethernet frames to study physical addressing using Wireshark Packet Analyzer Tool.
5	Write a program for error detection and correction for 7/8 bits ASCII codes using Hamming Codes or CRC.
6	Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocol in Peer-to-Peer mode.
7	Write a program to implement link state /Distance vector routing protocol to find suitable path for transmission.
	List of Assignments (Part B) - 8,9,10 mandatory and Any THREE from 11 to 15
8	Implement Subnetting and Test IP Addressing Schemes using a packet tracer tool. (Compulsory)
9	Write a program for DNS lookup. Given an IP address as input, it should return URL and vice Versa. (Compulsory)
10	Capture and analyze HTTP, DNS, and SMTP traffic using Wireshark Packet Analyzer Tool. (Compulsory)
11	Configuration of 3 router networks using one of the following protocols RIP/OSPF/BGP Using packet Tracer tool.
12	Write a program using TCP socket for wired network for File transfer
13	Write a program using UDP Sockets to enable file transfer (Script, Text, Audio and Video one file each) between two machines.
14	Configure DHCP server-client setup using a packet tracer tool.
15	Write a program to analyze following packet formats captured through Wireshark for wired Network. 1. Ethernet 2. IP 3.TCP 4. UDP

	List of Assignments (Part C)-Any ONE from 16 to 19
16	Study and Analyze the performance of HTTP, HTTPS and FTP protocol using Packet tracer Tool.
17	To study the SSL protocol by capturing the packets using Wireshark tool while visiting any SSL secured website (banking, e-commerce etc.).
18	Illustrate the steps for implementation of S/MIME email security through Microsoft Office Outlook.
19	To study the IPSec (ESP and AH) protocol by capturing the packets using Wireshark tool

Learning Resources

Virtual Lab Links: -

1. https://naim30.github.io/OS-virtual-lab/
2. https://profile.iiita.ac.in/bibhas.ghoshal/teaching_os_lab.html

Savitribai Phule Pune University **Third Year of Computer Science (2024 Course)**		
PEC-321A-CSC: Information Retrieval (Elective-I)		
Teaching scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	**CCE:** 30 Marks **End-Semester:** 70 Marks

Prerequisite Courses: Data Structures (PCC-201-CSC), Database Management Systems (PCC-251-CSC).

Course Objectives:

1. To understand the fundamental concepts of Information Retrieval and text processing techniques.
2. To learn various indexing structures and query processing methods used in IR systems.
3. To study different retrieval models, ranking techniques, and recent advancements in IR.
4. To evaluate the performance of IR systems and understand user interaction and visualization techniques.
5. To explore distributed, multimedia, and web-based information retrieval systems and their applications.

Course Outcomes:

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

- **CO1: Apply** text processing and indexing techniques for efficient information retrieval.
- **CO2: Implement** and **analyze** different IR models, query processing, and ranking methods.
- **CO3: Evaluate** IR system performance using standard metrics and user-oriented measures.
- **CO4: Analyze** distributed and multimedia information retrieval systems and their challenges.
- **CO5: Design** basic web search, recommender systems, and apply modern IR techniques.

Course Contents

Unit I - Fundamentals of Information Retrieval (07 Hours)

Introduction to Information Retrieval (IR), difference between Data Retrieval and IR, architecture of IR systems, applications. Text Processing: Tokenization, Stop-word removal, stemming, lemmatization, term normalization. Document Representation: Bag of Words model, term weighting (TF, IDF, TF-IDF). Vector Space Model and similarity measures (Cosine similarity).

Case Studies: Spam Email Filtering using Text Preprocessing.

Unit II - Information Indexing and Query Processing (07 Hours)

Inverted Index: structure, construction and working. Index compression (basic concepts). Query Processing: keyword and phrase queries. IR Models: Boolean, Vector Space, Probabilistic model. Ranking techniques: TF-IDF, BM25. Introduction to text classification (Naïve Bayes) and clustering (k-Means).

Case Studies: Vector Space Model in Academic Digital Libraries

Unit III - Evaluation Metrics and Visualization (07 Hours)

Evaluation framework for IR systems. Metrics: Precision, Recall, F1-score, MRR, NDCG. User-oriented metrics: click-through rate, response time. Search interfaces: query suggestion, autocomplete. Introduction to personalization and log analysis.

Case Studies: Ontology-Based Medical Information Retrieval (e.g., PubMed + MeSH Ontology)

Unit IV - Distributed Information Retrieval and Multimedia Search (07 Hours)

Introduction to multimedia IR. Content-Based Image Retrieval (CBIR). Feature extraction: color, texture, shape. Recommender systems: collaborative and content-based. Semantic search and word embeddings (overview). Deep Learning for Multimedia Retrieval, Image and Video Embeddings, Cross-modal Retrieval (Text–Image Search). Large-scale Multimedia Databases, Ethical Issues and Challenges in Multimedia Retrieval.

Case Studies: Multimedia Retrieval System

Unit V - Web Search and Information Retrieval Systems (07 Hours)

Web search basics and challenges. Search engine architecture and components. Web crawling and indexing (URL frontier, duplicate handling). Link analysis: PageRank, HITS algorithms. Web Data Extraction – Basics of Web Scraping using Python, HTTP Requests, HTML Parsing, Introduction to Beautiful Soup. Distributed IR: sharding and parallel processing concepts. Cloud-based search systems overview.

Case Studies: Web Crawling Architecture (Googlebot)

Learning Resources

Text Books:

1. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, *Introduction to Information Retrieval*, Cambridge University Press. 2008.
2. Information Retrieval in Practice, W. Bruce Croft, Donald Metzler , Trevor Strohmann 2015 This book was previously published by: Pearson Education, Inc.
3. Modern Information Retrieval by Ricardo Baeza-Yates, Berthier Ribeiro-Neto, Addison-Wesley, 2011

Reference Books:

1. Information Storage and Retrieval Systems by Gerald J. Kowalski and Mark T. Maybury, 2nd edition 2000, Springer.
2. Information Retrieval: Implementing and Evaluating Search Engines , by Stefan Büttcher, Charles L. A. Clarke, and Gordon V., MIT Press 2010
3. Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, Third Edition, Daniel Jurafsky , James H. Martin

E-Books:

1. https://web.stanford.edu/~jurafsky/slp3/
2. http://www.mmds.org/

MOOC / NPTEL/YouTube Links:

1. https://nptel.ac.in/courses/106106852
2. https://www.classcentral.com/course/swayam-introduction-to-information-retrieval-503018
3. https://www.youtube.com/@stanfordonline

PEC-321B-CSC: Cloud Computing (Elective-I)		
Teaching scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	**CCE:** 30 Marks **End Semester :** 70 Marks

Prerequisites Courses: Database Management systems(PCC-251-CSC)

Companion Course: Programming Elective Course Lab (PEC-321-CSC)

Course Objectives: The course aims to:

1. To **understand** the foundations, paradigms, and architecture of cloud computing, including service and deployment models.
2. To **analyze** enterprise data storage models and cloud storage architectures for managing large-scale data.
3. To **apply** virtualization and containerization technologies for efficient resource abstraction and orchestration.
4. To **evaluate** public cloud ecosystems, serverless computing, and cloud security frameworks for secure and scalable solutions.
5. To explore and **create** strategies for emerging trends such as DevOps, AIOps, FinOps, IoT-cloud convergence, and sustainable cloud paradigms.

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

- **CO1: Explain** the **evolution** of cloud computing paradigms, NIST model, service and deployment models, and layered architecture.
- **CO2:** Compare and **analyze** traditional storage systems (DAS, NAS, SAN) with cloud storage architectures and distributed file systems.
- **CO3: Apply** virtualization techniques and containerization tools (Docker, Kubernetes) to design scalable cloud-native applications.
- **CO4: Evaluate** cloud platforms (AWS, Azure, GCP), serverless models, and security frameworks including IAM and Zero-Trust Architecture
- **CO5: Design and create** cloud-native DevOps pipelines, optimize costs using FinOps, and propose sustainable and distributed cloud solutions for future paradigms

Course Contents

Unit I- Foundations and Architecture of Cloud Computing (07 Hours)

Introduction to Cloud Computing: Evolution of Cloud Computing, Computing Paradigms: Centralized, Distributed, Parallel, Cluster, and Grid Computing, Concept of Utility Computing, NIST Model: Definition, Essential Characteristics, Service Models (IaaS, PaaS, SaaS), Deployment Models (Public, Private, Hybrid, Community). Cloud Architecture: Layered Architecture, Shared Responsibility Model, Service Level Agreements (SLA): Taxonomy, Lifecycle, Availability and Reliability. Cloud Economics: Pay-as-you-go, Total Cost of Ownership (TCO). Cloud Migration: The 7-Step Migration Process.

Case Study: Healthcare Data Management in the Cloud

Unit II- Cloud Data Storage (07 Hours)

Enterprise Data Storage: Introduction to traditional storage models: DAS (Direct Attached Storage), NAS (Network Attached Storage), and SAN (Storage Area Network). Cloud Storage Architecture: Design principles of Distributed File Systems. Data Intensive Technologies: Concept of Big Data in Cloud, Data Management and Provisioning in Cloud Storage, Characterizing Data Storage from LANs to WANs.

Case Study: Google File System (GFS) and Hadoop Distributed File System (HDFS)

Unit III- Virtualization and Container Technology (07 Hours)

Virtualization Foundations: Theory and Taxonomy of Virtual Machines, **Hypervisors:** Type-1 (Native) and Type-2 (Hosted), Hardware-assisted Virtualization (Intel VT-x, AMD-V). **Resource Abstraction:** CPU Virtualization (Binary Translation, Emulation), Memory Virtualization (Shadow vs. Nested Page Tables), I/O Virtualization (Para-virtualization, Direct Passthrough). **Containerization:** OS-level Virtualization (Namespaces, Cgroups), Docker Architecture: Engine, Images, Registry. **Orchestration:** Introduction to Kubernetes (K8s) Architecture: Pods, Nodes, Control Plane, and Data Plane.

Case Study: College Cloud Lab Virtualization

Unit IV- Cloud Platforms and Security Framework (07 Hours)

Public Cloud Ecosystem: Deep dive into AWS (EC2, S3, VPC, IAM), Azure (App Services, Blob Storage), and GCP (Compute Engine). **Serverless:** FaaS vs. BaaS, Event-driven architectures. **Cloud Security:** Infrastructure Security, Multi-tenancy risks, VM-escape attacks. **Data Security:** Encryption-at-rest/transit, Key Management Systems (KMS). **Identity Management:** IAM roles and policies, Zero-Trust Architecture.

Case Study: Start-up Scaling with Serverless

Unit V- Emerging Trends in cloud computing (07 Hours)

Cloud-Native & DevOps: Infrastructure as Code (IaC) using Terraform/CloudFormation, CI/CD pipelines, GitOps. **Advanced Trends**: AIOps: Leveraging Cloud for GenAI and LLM deployment (Amazon Bedrock, Vertex AI). **Cloud Optimization**: FinOps: Cost Management and Resource Tagging. **Edge and IoT**: Edge-Cloud continuum, IoT-Cloud convergence for Smart Cities. **Future Paradigms**: Sovereign Clouds, Sustainable Cloud Computing (Green Cloud), and Distributed Cloud.

Case Study: Smart City IoT-Cloud Convergence

Learning Resources

Text Books:

1. Thomas Erl, Ricardo Puttini, Zaigham Mahmood *Cloud Computing: Concepts, Technology & Architecture* Publisher: Prentice Hall, 2013
2. Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi *Mastering Cloud Computing: Foundations and Applications Programming* Publisher: McGraw Hill Education, 2013.

Reference Books:

1. Rajkumar Buyya, James Broberg, Andrzej Goscinski *Cloud Computing: Principles and Paradigms* Publisher: Wiley, 2011.
2. Dan C. Marinescu *Cloud Computing: Theory and Practice* Publisher: Morgan Kaufmann/Elsevier, 2013.

E-Books:

Cloud Computing: A Hands-On Approach - Arshdeep Bahga, Vijay Madisetti - Google Books

MOOC / NPTEL/YouTube Links:

1. https://onlinecourses.nptel.ac.in/noc26_cs55/preview
2. https://onlinecourses.nptel.ac.in/noc26_cs29/preview?
3. https://onlinecourses.nptel.ac.in/noc21_cs15/preview?

Web Links: -

1. http://dhoto.lecturer.pens.ac.id/lecture_notes/internet_of_things/CLOUD%20COMPUTING%20Princip
2. https://www.lpude.in/SLMs/Master%20of%20Computer%20Applications/Sem_2/DECAP470_CLOUD
3. https://arpitapatel.files.wordpress.com/2014/10/cloud-computing-bible1.pdf
4. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-291r2.pd

Savitribai Phule Pune University **Third Year of Computer Science (2024 Course)**		
PEC-321C-CSC: User Interface and User Experience Design (UI/UX) (Elective-I)		
Teaching scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	**CCE:** 30 Marks **End-Semester:** 70 Marks

Prerequisite Courses, if any :

Course Objectives: The course aims to:

1. To understand principles of user-centered interface design.
2. To apply visual and interaction design techniques for digital interfaces.
3. To analyze user requirements using UX research methods.
4. To develop wireframes and prototypes for applications.
5. To explore modern UI/UX tools and industry practices.

Course Outcomes (COs): On completion of the course, learners should be able to:

- **CO1: Explain** fundamental concepts of UI/UX and user-cantered design principles.
- **CO2: Apply** human-computer interaction and visual design principles in interface design.
- **CO3: Analyze** user needs using UX research techniques and information architecture.
- **CO4: Develop** wireframes and interactive prototypes using modern design tools.
- **CO5: Evaluate** modern UI/UX trends, tools, and industry practices for designing effective digital products.

Course Contents

Unit I – Foundation of UI/UX Design (07 Hours)

Importance of User-Centric Design, Goals of User Interface Design , Design Thinking Process, Role of UX in Product Development Lifecycle, mental and conceptual model. **Usability Design Principles**: 4 Design Principles, Gestalt Principles of Design, Visual Design Principles, Form versus Function, Metaphors, Idioms and Affordances in UI design, User Interface Elements: Input Controls, Navigation Components, Information Components, Containers **User Research:** Qualitative and Quantitative User Research, Behavioral and Attitudinal User Research, Use of Personas, User Stories and Scenarios, Affinity Mapping.

Mapping of Course Outcomes: CO1

Unit II – Usability Engineering, Evaluation and Testing (08 Hours)

Usability Engineering: Concept of usability, usability principles, benefits of usability for users and organizations, internationalization and localization, human errors and their impact on usability. **Usability Evaluation:** Human information processing and memory, **Fitts's Law** and **Hick's Law**, usability inspection methods such as **Heuristic Evaluation** and **Cognitive Walkthrough**, user studies and field studies. U**sability Testing:** Planning and conducting usability testing, **Think-Aloud testing**, **A/B testing**, use of **heatmaps**, and basic **UX metrics** for evaluating user experience. **UX Design Foundations:** Ideation and research in UX design, content and interaction mapping, **paper prototyping**, introduction to **wireframes and interface layout**, applying **Nielsen's usability heuristics**, **Schneiderman's Golden Rules,** refining UI based on user feedback.

Mapping of Course Outcomes: CO1, CO2

Unit III - WEB DESIGN Principals **(08 Hours)**

The User Experience Process - User-centric design - The UX Phases - Waterfall vs. Agile - Web vs. App. Determining Strategy: User Research - Inspiration - Analytics - User Needs and Client Needs - Target Audience - What is in and What is Out: Outlining Scope - Content and Functionality. The Sitemap: Introduction to Sitemaps - Information Architecture - Sitemap Concerns - annotated process - Elements - Treejack Introduction - Treejack Analysis.

Mapping of Course Outcomes: CO1, CO3

Unit IV - User Experience (UX) Design **(09 Hours)**

Introduction to wireframes, purpose and importance in UX design, low-fidelity vs high-fidelity wireframes, sketching interfaces, layout grids, design hierarchy, and responsive layouts. **Prototyping Concepts:** Introduction to prototypes, types of prototypes (paper prototypes, digital prototypes, interactive prototypes), fidelity levels, advantages of prototyping in UX design, iterative design process**. Interaction Design:** Principles of interaction design, micro-interactions, feedback mechanisms, navigation flows, usability considerations in interaction design, designing intuitive interfaces.
UI Design Tools: Introduction to modern UI/UX design tools such as Figma, Adobe XD, Sketch, and InVision. Creating interactive prototypes, collaborative design workflows, design handoff to developers. **Design Documentation:** User flow diagrams, storyboards, design specifications, and design system basics.

Mapping of Course Outcomes: CO3, CO4

Unit V -Emerging Trends **(08 Hours)**

Emerging technologies in UX Design: Voice UI, Touchless gesture control, Intelligent UX, Conversational UX, Immersive Media and Fluid UX Designing for Web and Mobile Interfaces, IoT applications, Industry Specific UX Design: FinTech, Education, Health Care, E-commerce and Industrial Websites, designing for Wearable Devices, designing for Augmented Reality, Virtual Reality and Mixed Reality, Tomorrow's challenges in UX design.

Mapping of Course Outcomes: CO5

Learning Resources

Text Books:

1. Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs, Niklas Elmqvist, Nicholas Diakopoulos – Designing the User Interface: Strategies for Effective Human-Computer Interaction, Pearson.
2. Don Norman – The Design of Everyday Things, Basic Books.
3. Russ Unger and Carolyn Chandler – A Project Guide to UX Design, New Riders.

Reference Books:

1. Steve Krug – Don't Make Me Think: A Common Sense Approach to Web Usability, New Riders.
2. Alan Cooper, Robert Reimann, David Cronin – About Face: The Essentials of Interaction Design, Wiley.
3. Jesse James Garrett – The Elements of User Experience, New Riders.
4. Jeff Gothelf & Josh Seiden – Lean UX, O'Reilly

MOOC / NPTEL/YouTube Links: -

1. Google UX Design Professional Certificate – Coursera
2. N Interaction Design Specialization – Coursera (UC San Diego)
3. User Experience Design Fundamentals – Udemy
4. Introduction to User Experience Design – Georgia Tech (Coursera)
5. Figma UI/UX Design Essentials – Udemy

Savitribai Phule Pune University **Third Year of Computer Science (2024 Course)**		
PEC-321D-CSC: Embedded Systems (Elective-I)		
Teaching scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	**CCE:** 30 Marks **End-Semester:** 70 Marks

Prerequisite Courses: Digital Electronics & Logical Design.

Companion Course: Elective-I Laboratory.

Course Objectives: The course aims to:

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

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

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

Course Contents

Unit I - Introduction to Embedded Systems (07 Hours)

Overview of Embedded Systems, Processor Embedded into a system, Embedded Hardware Units and Devices in system, Embedded Software, Complex System Design, Design Process in Embedded System, Formalization of System Design, Classification of Embedded Systems.

Case Study: Any one Case study based on "Healthcare", Industrial Automation" or any relevant topic which will be useful for society.

Unit II Embedded System Design Concepts (07 Hours)

Characteristics and Quality Attributes of Embedded Systems, Operational and non-operational quality attributes, Embedded Systems-Application and Domain specific, Hardware Software Co-Design and Program Modeling (excluding UML), Embedded firmware design and development (excluding C language).

Case Study: Any one Case study based on "Smart temperature control system" or "Smart Energy Meter" or any relevant topic which will be useful for society.

Unit III- ARM Architecture (07 Hours)

Introduction, RISC design philosophy, ARM design philosophy, Embedded system hardware – AMBA bus protocol, ARM bus technology, Memory, Peripherals, Embedded system software – Initialization (BOOT) code, Operating System, Applications. ARM Processor Fundamentals, ARM core dataflow model, registers, current program status register, Pipeline, Exceptions, Interrupts and Vector Table, Core extensions. Operating modes of ARM (User, FIQ, IRQ, Supervisor, Abort, Undefined and System mode)

Case Study: Any one Case study based on "Smart Home Lighting Controller" or "Automotive Engine Monitoring System" or any relevant topic which will be useful for society.

Unit IV- Introduction to ARM Instruction Set (07 Hours)

Introduction, Data processing instructions, Load – Store instruction, Software interrupt instructions, Program status register instructions, Loading constants, ARMv5E extensions, Conditional Execution

Case Study: Any one Case study based on "Embedded Operating System Service Call" or "Audio Signal Processing" or any relevant topic which will be useful for society

Unit V- RTOS and IDE for Embedded System (07 Hours)

Operating System basics, Types of operating systems, Task, process and threads (Only POSIX Threads with an example program), Thread preemption, Preemptive Task scheduling techniques, Task Communication, communication protocol ,CAN protocol for Automotive Industry, MODBUS for Industrial Automation, Ethernet for IoT application and UART, SPI, I2C for basic hardware communication.Task synchronization issues in RTOS (Racing and Deadlock) Design parameters for RTOS, Integration and testing of Embedded hardware and firmware, Embedded system Development Environment – Block diagram (excluding keil).

Case Study: Any one Case study based on "Automotive lane departure warning system" or "Robotic Arm Controller" or any relevant topic which will be useful for society.

Learning Resources

Text Books:

1. Embedded Systems - Architecture Programming and Design – Raj Kamal, 2nd ed., 2008, TMH.
2. Shibu K V, "Introduction to Embedded Systems", Tata McGraw Hill Education
3. Andrew N Sloss, Dominic System and Chris Wright, "ARM System Developers Guide", Elsevier, Morgan Kaufman publisher, 1st Edition, 2008

Reference Books:

1. Embedded Microcomputer Systems, Real Time Interfacing - Jonathan W. Valvano - Brookes / Cole, 1999,Thomas Learning

E-Books:

1. Introduction to Embedded Systems,A Cyber-Physical Systems Approach by Edward Ashford Lee, Sanjit Arunkumar Seshia

Savitribai Phule Pune University **Third Year of Computer Science (2024 Course)**		
PEC-322-CSC: Elective – I: Laboratory		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	01	**Term Work** : 50 Marks

Guidelines for Instructor's Manual

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

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks an assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis). Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal must be avoided. Use of DVD containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

Guidelines for Laboratory /Term Work Assessment

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

Guidelines for Practical Examination:

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

Guidelines for Laboratory Conduction – Information Retrieval

List of Assignments –Group-A (Any 4 from the given list)

1. Implement a Python-based text preprocessing pipeline that performs tokenization, stop-word removal, stemming, and lemmatization on a given text corpus, comparing outputs at each stage.
2. Compute TF, IDF, and TF-IDF scores manually and using scikit-learn, then rank documents against a user query using cosine similarity to retrieve the top-3 relevant documents.
3. Build a positional inverted index from a collection of text documents and implement Boolean (AND, OR, NOT) and phrase query processing using the constructed index.
4. Evaluate an IR system using Precision, Recall, F1-Score, MRR, and NDCG metrics on a benchmark dataset, and plot Precision-Recall curves with Mean Average Precision reported across all queries.
5. Build a CBIR system that extracts color histogram, texture (GLCM), and shape (Hu Moments) features from an image dataset, retrieving the top-5 visually similar images for a given query image.

List of Assignment - Group B - Mini-Project (In Team of 3-4 Students any 1)

1. Design and Implementation of a Smart Healthcare Document Retrieval System Using Blockchain-Integrated IR Techniques
2. Web Page Ranking using PageRank Algorithm
3. Clustering Documents using K-Means

Guidelines for Laboratory Conduction –Cloud Computing

Sr. No.	**List of Assignments – Group-A - (Any 4 from the given list)**
1	Provisioning and Configuration of IaaS Compute Instance
2	To demonstrate serverless hosting by configuring a Cloud Storage bucket to serve static web content (HTML/Images) to the public.
3	To understand the lifecycle of a container by pulling, building, and running isolated application environments using Docker.
4	Exploring Database-as-a-Service (DBaaS) using No-Code Tools
5	implement a serverless "Hello World" function and understand event-driven execution, triggers, and cloud logging
6	Google App Engine (GAE) Setup: Install Google App Engine. Create hello world app or any other simple web applications using python/java.
7	Case Study on PaaS (Google App Engine). Use GAE launcher to launch the web applications.
8	Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.
9	Creating an application in Salesforce.com using Apex Programming Language

10	To study creating a warehouse application in Salesforce.com
11	To study installation and Configuration of Hadoop.
12	Install Hadoop single node cluster and run simple application like wordcount.
13	To study the Cloud Computing Security Issue

List of Assignment - Group B - Mini-Project (In Team of 3-4 Students any 1)	
1	Authenticated File Encryption System
2	Network Security Analysis using Encrypted Communication and Port Scanning.

Guidelines for Laboratory Conduction –UI/UX

List of Assignments – Group-A - (Any 4 from the given list)	
1	Creating user personas and user journey maps.
2	Designing low-fidelity wireframes for a web application.
3	Creating interactive prototypes using Figma / Adobe XD.
4	Performing heuristic evaluation of an existing website.
5	Conducting basic usability testing with users.
6	Designing mobile app interface layouts.
7	Creating sitemap and information architecture for a website.
List of Assignments - Group B - Mini-Project (In Team of 3-4 Students any 1)	
1	UI/UX Design for Smart Healthcare Mobile Application
2	Redesign of College Website for Better User Experience
3	UX Design for Online Learning Platform
4	Mobile Banking App Interface Design
5	Smart Parking System Mobile Application UI
6	E-Commerce Mobile App UX Optimization
7	Tourism Guide Mobile App Interface Design
8	Food Delivery App UI/UX Prototype
9	Fitness Tracking Application UI/UX Design
10	Smart Campus Navigation System UI Design
11	UX Design for IoT Smart Home Control Application
12	Redesign of Government Service Portal for Better Accessibility

Guidelines for Laboratory-Embedded System

Conduct the following experiments on an ARM CORTEX M3 evaluation board to learn Assembly Language Program and using evaluation version of Embedded 'C' & Keil uVision-4 tool/compiler.

List of Assignments	
List of Assignments – Group-A - (Any 4 from the given list)	
1	Write an Assembly Language Program (ALP) to i) Multiply two 16-bit numbers. ii) Add two 32-bit numbers.
2	Write a program to find the factorial of a number.
3	Write a program to add an array of 16 bit numbers and store the 32 bit result in internal RAM.
4	Write a program to find the largest or smallest number in an array of 32 numbers.
5	Write a program for task synchronization in RTOS with the use of Mutex & Semaphore
List of Assignments - Group B - Mini-Project (In Team of 3-4 Students any 1)	
1	Interface a Stepper motor and rotate it in clockwise and anti-clockwise direction.
2	Interface a DAC and generate Triangular and Square waveforms.
3	Display the Hex digits 0 to F on a 7-segment LED interface, with a suitable delay in between.
4	Interface a simple Switch and display its status through Relay, Buzzer and LED
5	Write a program for task synchronization in RTOS with the use of binary and counting semaphore, Priority Inversion.

Savitribai Phule Pune University **Third Year of Computer Science (2024 Course)**		
MDM-331-CSC: Robotics and Automation		
Teaching scheme	**Credits**	**Examination Scheme**
Tutorial: 01 Hours/Week **Practical** : 04 Hours/Week	**Tutorial:** 01 **Practical**: 02	**Term Work:** 50 Marks

Prerequisite Courses: Engineering Physics, Engineering Mathematics, Basics of Electrical Engineering, Basics of Electronics Engineering & Engineering Graphics.

Course Objectives: The course aims to:

1. To introduce various types of Robots and the functional elements of Robotics.
2. To impart knowledge of robot drive systems & educate on various sensors used in Robotic automation.
3. To introduce various types the end effectors.
4. To impart knowledge of basics of Robot Programming and robotic Applications

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

- **CO1:** To **UNDERSTAND** basic concepts of robotics.
- **CO2:** To **SELECT** appropriate drive & sensors for Robotic applications.
- **CO3:** To **COMPARE** and **SELECT** robot and end effectors as per application.
- **CO4:** To **UNDERSTAND** the fundamentals of robot programming and application

Course Contents

Unit I - Fundamentals of Robotics (03 Hours)

Definitions of Industrial Robot, Type and Classification of Robots, Asimov ‘s laws of robotics, Robot Configurations, Robot Components, Robot Degrees of Freedom, Work volume and work envelope, Robot Joints and symbols, Robot Coordinates

Unit II - Sensors (03 Hours)

Transducers and sensors, Sensors in robotics, Principles and applications of the following types of sensors- Proximity Sensors, Photoelectric Sensors, Position sensors – Piezoelectric Sensor, LVDT, Resolvers, Encoders – Absolute and Incremental: - Optical, Magnetic, Capacitive, pneumatic Position Sensors, Range Sensors- Range Finders, Laser Range Meters, Touch Sensors, Force and torque sensors.

Unit III - Industrial Automation and AI in Robotics (03 Hours)

Introduction to industrial automation, PLC basics and SCADA overview, IoT in automation systems
Basics of AI in robotics (computer vision, ML concepts)

Case studies: smart factories, autonomous robots

Unit IV Autonomous Navigation & Path Planning (03 Hours)

Environment Representation: Occupancy grids, topological maps, and configuration spaces. Path Planning Algorithms: Dijkstra's Algorithm, A* (A-star) search, and RRT (Rapidly-exploring Random Trees). Localization: Odometry, sensor fusion (Kalman Filters), and particle filters. Obstacle Avoidance: Potential field methods and reactive control loops.

Unit V Robotic Middleware & Industry 4.0 (03 Hours)

Robot Operating System (ROS): Architecture (Nodes, Topics, Services, Messages) and Workspace management. Simulation Environments: Working with Gazebo and RViz for testing and validation. Industrial Automation: Introduction to PLC (Programmable Logic Controllers) and SCADA systems. Industry 4.0: Impact of IoT, Big Data, and AI in smart manufacturing and automation.

Learning Resources

Text Books:

1. Industrial Robotics - Mikell P. Groover, McGraw Hill
2. Robotics: Control, Sensing, Vision and Intelligence - McGraw Hill
3. Introduction to Robotics - Tata McGraw Hill
4. Robotics Engineering - Prentice Hall
5. Programmable Logic Controllers - McGraw Hill

Reference Books:

1. Steven S. Skiena, "The Algorithm Design Manual", Springer, 2ndedition, ISBN : 978-1-84800-069-8

MOOC / NPTEL/YouTube Links: -

https://nptel.ac.in/courses/106106133

Guidelines for Laboratory Conduction- Robotics and Automation

List of Assignment - Group A (Any SIX)

1. Study of different robotic components, joints, links, and end effectors
2. Interfacing and control of DC motor using Arduino
3. Servo motor position control using PWM signals
4. Obstacle detection robot using ultrasonic sensor
5. Design and implementation of line follower robot
6. Interfacing IR sensors and proximity sensors with microcontroller
7. Stepper motor control for robotic arm movement
8. Basic PLC programming using ladder logic for industrial automation
9. IoT-based automation system using sensors and cloud monitoring

Group B - Mini-project: Design of a simple autonomous robotic or automation application – Suggested List

1. Smart warehouse robot
2. Automated parking system
3. IoT-based industrial monitoring system
4. Pick-and-place robotic arm
5. Smart conveyor automat
6. Home automation using sensors and actuators

Savitribai Phule Pune University **Third Year - Computer Science (2024 Pattern)**		
Open Electives		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 02 Hours/Week	02	**CCE:** 15 Marks **End-Semester:** 35 Marks

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

Sr.	Open Elective Course Name	Offering Discipline
1	IPR and Cyber Laws	Law / Faculty of Humanities
2	Agri Business Management: Banking Operation and finance	Commerce & Management
3	Product Costing for Mechanical Engineering	Commerce & Management
4	Sustainability Development	Commerce & Management
5	Material and Logistics	Management
6	The Constitution of India	
7	Digital Personal Data Protection	
8	Environmental Law	Law / Faculty of Humanities
9	Construction Laws and contracts	
10	Human Resource Management	
11	Statistics and Computer Applications	Commerce & Management
12	Business Administration	Commerce & Management
13	Business Marketing	Commerce & Management
14	Entrepreneurship Development	Commerce & Management

Savitribai Phule Pune University **Third Year of Computer Science (2024 Course)**		
VEC-232-CSC: Technical Seminar		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	01	**Oral**: 25 Marks

Course Objectives: The course aims to:

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

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

- **CO1: Identify** and select emerging Computer Science and Artificial Intelligence domains research problems through a systematic literature survey.
- **CO2:** Critically **analyze** research papers using technical, methodological, and ethical perspectives.
- **CO3: Prepare** structured technical reports and presentations.
- **CO4:** Deliver effective oral presentations demonstrating clarity, confidence, and domain **understanding.**
- **CO5 : Demonstrate** professional ethics and academic integrity

Course Contents

Guidelines for Conduct of Technical Seminar

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

Topic Selection Guidelines

1. Topic must be from emerging AI & DS domains (last 3–5 years).
2. Must involve a minimum of 5 recent research papers (IEEE/SCI/Scopus indexed preferred).
3. Should not be a basic textbook topic.
4. Must include: Problem statement, State-of-the-art analysis, Comparative study, Ethical & societal impact, Future research scope, Interdisciplinary themes aligned with NEP encouraged.
5. Topic approval by a faculty panel.

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

Seminar Process

- Conduct an orientation session explaining: Objectives of the technical seminar, Evaluation criteria and expected outcomes
- Each student must submit: Title of the seminar, Problem statement, Relevance to current technology trends, Approval by guide is mandatory before proceeding

- **Literature Survey & Problem Understanding (Week 2-4)**

- Students must: Refer minimum 5-8 recent research papers (IEEE, Springer, Elsevier, ACM etc.)
- Use scholarly databases like IEEE Xplore, Google Scholar, ScienceDirect
- Prepare a literature survey matrix, including:
* Author/year
* Methodology used
* Key findings
* Limitations
- Identify: Research gaps and Challenges in existing approaches

- **Synopsis Preparation & Presentation (Week 4-5)**

- Submit a 2-3 page synopsis including: Introduction, Literature insights, Objectives, Proposed seminar scope
- Conduct a Synopsis Presentation (5-7 minutes): Evaluate clarity of understanding, Receive feedback for improvement
- Approval required before proceeding to full report

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

- Students should: Deeply analyze concepts, models, architectures, or case studies, Include diagrams, flowcharts, and comparative tables
- Weekly review meetings with guide: Track progress, Ensure conceptual clarity,
- Emphasis on: Critical analysis (not just description), Real-world applications

- **Draft Report Submission & Review (Week 8-10)**

- Submit first draft of the report
- Guide provides feedback on: Technical content quality, Structure and coherence, Referencing and plagiarism,
- Students must revise based on suggestions by the guide

- **Pre-Seminar Presentation (Mock Evaluation) (Week 10-11)**

- Conduct a mock presentation simulating final evaluation
- Focus on: Presentation skills, Time management, Handling questions
- Peer and faculty feedback should be incorporated

- **Final Report Submission (Week 11-12)**

- Submit: Final hard copy (if required), Soft copy (PDF format)
- Ensure:Proper formatting, Plagiarism compliance (<20%), Correct referencing using reference managers like Zotero and Mendeley Desktop

- **Final Seminar Presentation & Viva Voce (Week 12-13)**

- Presentation duration: 10-15 minutes, Followed by Q&A session (5-10 minutes)

- Evaluation based on: Depth of understanding, Analytical ability, Communication skills

Method of Evaluation

- During the seminar session each student is expected to prepare and present a topic on engineering/ technology, for duration of about 12to 15 minutes.
- Each student is expected to present atleast twice during the semester and the student is evaluated based on that.
- At the end of the semester, he / she can submit a report on his / her topic of seminar and marks are given based on the report.
- A Faculty guide is to be allotted and he / she will guide and monitor the progress of the student and maintain attendance.

Savitribai Phule Pune University, Pune

Maharashtra, India

TE - Computer Science

2024 Pattern

Semester VI

With effect from Academic Year 2026-27

Savitribai Phule Pune University **Third Year of Computer Science (2024 Course)**		
PCC-351-CSC: Machine Learning		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03Hours/Week	03	**CCE:** 30 Marks **End-Semester:** 70 Marks

Prerequisite Courses: Probability and Statistics(PCC-253-CSC).

Course Objectives: The course aims to:

1. Provide the fundamental concepts of Machine Learning.
2. Develop an understanding of regression concepts, techniques, and evaluation metrics used for predictive modeling.
3. Imbibe knowledge of classification models and algorithms for solving real-world classification problems.
4. Familiarize students with clustering algorithms and ensemble learning techniques.
5. Give insight into reinforcement learning concepts and their use in sequential decision-making problems.

Course Outcomes:
After successful completion of the course, learner will be able to:

- **CO1: Apply** fundamental Machine Learning concepts in various learning paradigms and real-world engineering applications.
- **CO2: Use** various regression models for predictive modeling and data analysis.
- **CO3: Identify** different types of classification problems, including binary, multiclass, balanced, and imbalanced classification.
- **CO4: Analyze** clustering algorithms for grouping similar data points and ensemble learning techniques for improving model performance.
- **CO5: Distinguish** reinforcement learning from supervised and unsupervised learning approaches.

Course Contents

Unit I - Fundamentals of Machine Learning (08 Hours)

Introduction to machine learning, scope of machine learning, AI vs ML vs Data Science, traditional programming vs ML paradigm, and real-world engineering applications. Types of Learning: Supervised, unsupervised, semi-supervised, and reinforcement learning. Models of Machine Learning: Geometric model, probabilistic models, logical models, grouping and grading models, parametric and non-parametric models. Introduction to Feature Engineering. Feature Transformation: Dimensionality reduction techniques- Principal Component Analysis (PCA); Linear Discriminant Analysis (LDA).

.

Case Studies: Machine Learning Based Student Performance Prediction and Feature Engineering Analysis.

***Mapping of Course Outcomes for Unit I:** CO1

Unit II - Supervised Learning-Regression (08 Hours)

Introduction to regression, need of regression, regression vs correlation. Types of regression: Univariate vs Multivariate, Linear vs Nonlinear, Simple vs Multiple, Bias–Variance Tradeoff, Overfitting and Underfitting. Regression Techniques: Simple and Multiple Linear Regression; Polynomial Regression; Decision Tree Regression, Random Forest Regression, Support Vector Regression. Regularization Techniques: Ridge Regression (L2); Lasso Regression (L1). Evaluation Metrics: Mean Squared Error (MSE); Mean Absolute Error (MAE); Root Mean Squared Error (RMSE); R-squared (R^2).

Case Studies: Comparative Study of Regression Techniques for House Price Prediction

***Mapping of Course Outcomes for Unit II:** CO2

Unit III - Supervised Learning-Classification (08 Hours)

Introduction to classification, need of classification. Types of Classification: Binary and Multiclass, Binary vs. Multiclass Classification, Balanced and Imbalanced Classification Problems. Binary Classification: Linear classification model, decision boundary. Performance Evaluation: Confusion Matrix, Accuracy, Precision, Recall, F1-Score.

Multiclass Classification: One-vs-One and One-vs-All classification techniques, multiclass confusion matrix; Per-Class Precision and Per-Class Recall; Macro, Micro and Weighted Averaging Methods.

Classification Algorithms: K-Nearest Neighbors (KNN), Linear Support Vector Machine (SVM), Soft Margin SVM. Kernel Functions in SVM: Radial Basis Function (RBF/Gaussian) Kernel, Polynomial Kernel, Sigmoid Kernel.

Case Studies: Comparative Study of Classification Algorithms for Email Spam Detection

***Mapping of Course Outcomes for Unit III:** CO3

Unit IV - Unsupervised Learning and Ensemble Learning (08 Hours)

Introduction to clustering, need for clustering, types of clustering, Hierarchical Clustering – Agglomerative and Divisive methods, Partitioning Methods: K-Means clustering algorithm, advantages and limitations, Elbow method, Silhouette method; K-Medoids, Density-Based Clustering: DBSCAN algorithm, working mechanism, advantages and limitations. Distribution-Based Clustering: Gaussian Mixture Model. Applications, introduction to Ensemble Learning, homogeneous and heterogeneous ensemble methods, advantages and limitations. Basic Ensemble Techniques: Voting (Max Voting, Averaging, Weighted Averaging). Advanced Ensemble Techniques: Bagging and Random Forest. Boosting: AdaBoost, Gradient Boosting , Stacking.

Case Studies: Customer Segmentation and Sales Prediction.

***Mapping of Course Outcomes for Unit IV:** CO4

Unit V - Reinforcement Learning (08 Hours)

Introduction, need of reinforcement learning, components of reinforcement learning, comparison with supervised and unsupervised learning, applications and challenges of reinforcement learning. Markov Decision Process: Markov property, elements of MDP, episodic and continuing tasks. **Reinforcement Learning Framework:** Policy, state value function, action value function, Bellman equation, optimal policy. Reinforcement Learning Algorithms: Exploration vs Exploitation, ε-greedy strategy, dynamic programming, Q-Learning algorithm and update rule, simple reinforcement learning for game playing Tic-Tac-Toe.

Case Studies: Smart Traffic Signal Control using Q-Learning.

***Mapping of Course Outcomes for Unit V:** CO5

Learning Resources

Text Books:

1. Alpaydin, Ethem, "Introduction to Machine Learning", 2nd Edition, MIT Press, 2014, ISBN: 9780262028189.
2. Müller, Andreas C. and Guido, Sarah, "Introduction to Machine Learning with Python: A Guide for Data Scientists", 1st Edition, O'Reilly Media, 2016, ISBN: 978-1-449-36941-5.
3. Mitchell, Tom M E"Machine Learning", 1st Edition, McGraw-Hill Education, 1997, ISBN: 978-0070428072

Reference Books:

1. Flach, Peter, "Machine Learning: The Art and Science of Algorithms that Make Sense of Data," 1st Edition, Cambridge University Press, 2012, ISBN: 978-1107422223.
2. Murphy, Kevin P., "Machine Learning: A Probabilistic Perspective", 1st Edition, MIT Press, 2012, ISBN: 978-0262018029.
3. Shalev-Shwartz, S., & Ben-David, S., "Understanding Machine Learning: From Theory to Algorithms , 1st edition , Cambridge University Press, 2014, ISBN: 978-11070573435.

MOOC / NPTEL/YouTube Links: -

1. NPTEL Course: Introduction to Machine Learning, by Prof.Balaraman Ravindran , IIT Madras https://nptel.ac.in/courses/106106139

E-Books

1. Hastie, Trevor, Tibshirani, Robert, & Friedman, Jerome, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction", 2nd Edition, Springer, 2009, ISBN: 978-0387848570. https://hastie.su.domains/ElemStatLearn/
2. Sutton, Richard S., & Barto, Andrew G., "Reinforcement Learning: An Introduction", 2nd Edition,MIT Press, 2018, ISBN: 978-0262039246. https://www.andrew.cmu.edu/course/10-703/textbook/BartoSutton

Savitribai Phule Pune University **Third Year of Computer Science (2024 Course)**		
PCC-352-CSC: Software Engineering and Modelling		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 02 Hours/Week	02	**CCE:** 30 Marks **End-Semester:** 70 Marks

Prerequisite Courses: Data Structure (PCC-201- CSC), Advanced Data Structures (PCC-252-CSC)

Companion Course: Software Engineering and Modelling Laboratory.

Course Objectives:

1. To understand fundamental principles and challenges of software engineering.
2. To study software process models and requirement engineering techniques.
3. To apply software modelling concepts using UML diagrams.
4. To understand project management, testing concepts.
5. To introduce students to modern and advanced software engineering practices.

Course Outcomes:

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

- **CO1: Explain** software engineering principles and software development life cycle models.
- **CO2: Analyze** and document software requirements using standard practices.
- **CO3: Design** software systems using UML-based software modelling techniques.
- **CO4: Apply** project management, testing, and quality assurance techniques.
- **CO5: Understand** and **evaluate** modern and advanced trends in software engineering.

Course Contents

Unit I - Introduction to Software Engineering (07 Hours)

Software Engineering: Definition, Importance, Software Crisis, Software Myths & Reality, The Software Process: Software Process Framework, SDLC, Prescriptive Process Models: Waterfall, VModel, Incremental, Evolutionary Process Models: Prototyping, Spiral, Rapid Application Development (RAD), Agile Development Models: What Is Agility? Agility and the Cost of Change, Extreme Programming, Other Agile Process Models, Overview of Scrum and Kanban.

Case Studies: Development of an Online Examination System to demonstrate various Software Life Cycle Models and Software Process Structures.

***Mapping of Course Outcomes for Unit I:** CO1

Unit II - Requirements Engineering (07 Hours)

Requirements Engineering, Establishing the Groundwork: Identifying Stakeholders, Recognizing Multiple Viewpoints, working toward Collaboration, Asking the First Questions, Non-functional Requirements, Traceability Eliciting Requirements: Collaborative Requirements Gathering, Usage Scenarios, Elicitation Work Products, Developing Use Cases, Building the Analysis Model, Elements of the Analysis Model, Negotiating Requirements: Requirement Validation and Management. Software Requirements Specification (SRS)

Case Studies: Requirements Engineering and SRS development for a College ERP System covering stakeholder identification, requirement elicitation, use case modelling, non-functional requirements, validation and traceability.

***Mapping of Course Outcomes for Unit II:** CO2

Unit III - Software Modelling (07 Hours)

Introduction to Software Architecture, Architectural design principles and patterns, Types of architectures: Layered, Client-Server, Micro services, MVC, Architectural styles and quality attributes, Architecture documentation (4+1 view model), Object Modelling Using UML Basic Object-Orientation Concepts, Unified Modelling Language (UML), UML Diagrams, Use Case Model, Class Diagrams, Interaction Diagrams, State Chart Diagram, Activity Diagrams, Package Diagrams, Component Diagrams, Deployment Diagram

Case Studies: Design and UML modelling of a Library Management System to demonstrate abstraction, modularity, object-oriented design principles, and complete UML modelling including structural and behavioural diagrams

***Mapping of Course Outcomes for Unit III:** CO3

Unit IV - Software Project Management and Testing (07 Hours)

Software Project Management Complexities, Project Planning, Metrics for Project Size Estimation, Project Estimation Techniques, Empirical Estimation Techniques, COCOMO—A Heuristic Estimation Technique, Scheduling, Organisation and Team Structures, Staffing, Risk Management, Software Configuration Management. Software Testing, A Strategic Approach to Software Testing, Test Strategies for Conventional Software, Test Strategies for Object-Oriented Software, Test Strategies for WebApps, Validation Testing, System, Testing, Debugging, Defect Life Cycle, Manual and Automated testing

Case Studies: Software Project Planning, Estimation, Scheduling, Risk and Configuration Management for an E-Governance Citizen Service Portal using COCOMO and empirical estimation techniques.Testing of an Online Shopping Web Application covering manual and automated testing, test planning, strategic testing approaches, object-oriented testing, Web application testing, validation, system testing and debugging techniques.

***Mapping of Course Outcomes for Unit IV:** CO4

Unit V - Advanced Concepts in Software Engineering (07 Hours)

Agile and DevOps practices, CI/CD pipelines, DevSecOps and Secure Software Development, Software Architecture Patterns (Event-Driven, Serverless, Reactive Systems), AI in Software Engineering (AI-assisted development, automated testing), Low-Code / No-Code Platforms, Software Sustainability and Green Computing.

Case Studies: Design and implementation of a Smart City Service Portal using Agile practices, DevOps, CI/CD pipelines, Microservices architecture, Cloud-native deployment, and reusable software components.

***Mapping of Course Outcomes for Unit V:** CO5

Learning Resources

Text Books

1. Roger S. Pressman, Software Engineering: A Practitioner's Approach, Mcgraw-Hill.
2. Ian Sommerville, Software Engineering, Pearson Education.

Reference Books

1. Rajib Mall, Fundamentals Of Software Engineering, Fifth Edition, ISBN-978-93-88028-02-8
2. Pankaj Jalote, An Integrated Approach to Software Engineering, Springer.
3. Booch, Rumbaugh, Jacobson, The UML User Guide, Addison-Wesley.

Savitribai Phule Pune University **Third Year of Computer Science (2024 Course)**		
PCC-353-CSC: Machine Learning Laboratory		
Teaching scheme	**Credits**	**Examination Scheme**
Practical : 04 Hours/Week	**Practical:** 02	**Term Work:** 50 Marks **Practical** : 25 Marks

Prerequisite Courses: Probability and Statistics (PCC-253-CSC), Python Programming

Course Objectives: The course aims to:

1. Enable students to apply feature engineering techniques, implement PCA and LDA for feature reduction, and use a multiple linear regression model with suitable open-access datasets.
2. Provide knowledge of polynomial regression, regularization techniques, and logistic regression using suitable open-access datasets.
3. Introduce Support Vector Machine classifiers, K-Means clustering, and DBSCAN using suitable open-access datasets.
4. Familiarize with ensemble learning techniques and reinforcement learning using suitable datasets and simple decision-making problems.

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

- **CO1: Apply** feature engineering, dimensionality reduction techniques, and multiple linear regression to prepare datasets for predictive modeling.
- **CO2: Demonstrate** the use of polynomial regression, regularization techniques, and logistic regression for regression and classification modeling using suitable datasets.
- **CO3: Make** use of Support Vector Machine classifiers, K-Means clustering, and DBSCAN to perform classification and clustering on suitable datasets.
- **CO4:** E**valuate** predictive modeling and simple decision-making tasks using ensemble learning techniques and the Learning algorithm.

Course Contents

Guidelines for Instructor’ s Manual

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

Guidelines for Student’ s Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor’s sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis). Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal must be avoided. Use of DVD containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

Guidelines for Laboratory /Term Work Assessment

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

Guidelines for Practical Examination

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

Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy needs to address the average students and inclusive of an element to attract and promote the intelligent students. Use of open source software is encouraged. Based on the concepts learned, Instructors may also set one assignment or mini-project that is suitable to AI & DS branch beyond the scope of the syllabus. Operating System recommended: - 64-bit Open Source Linux or its derivatives, or Windows OS. Programming tools recommended: - Open Source Python, Programming tools like Jupyter Notebook, Pycharm, Spyder etc.

Suggested List of Laboratory Experiments/Assignments	
1	Apply basic feature engineering techniques such as handling missing values, encoding categorical variables, and scaling numerical attributes on the suitable open access dataset to prepare a clean and model ready dataset for machine learning analysis.
2	Make use of dimensionality reduction process using PCA and LDA on a suitable open-access high-dimensional dataset and observe the influence of feature reduction on model performance and computational efficiency.
3	Develop a multiple linear regression model using a suitable real-world open-access dataset and assess its predictive capability using regression evaluation metrics such as MSE, RMSE, and R^2.
4	Demonstrate the use of polynomial regression with varying polynomial degrees on a suitable open-access dataset and observe the effect of model complexity on prediction accuracy in terms of the bias–variance trade-off.
5	Classify data using regularization techniques such as Ridge and Lasso regression on a suitable multivariate open-access dataset and observe the influence on model overfitting and multicollinearity.
6	Illustrate the performance of a binary classification model using logistic regression on a suitable open-access dataset through a confusion matrix, precision, recall, and F1-score.
7	Utilize Support Vector Machine classifiers with linear and nonlinear kernels on a suitable open-access dataset and compare classification performance using appropriate evaluation metrics.
8	Apply the K-Means clustering algorithm on a suitable open-access dataset to obtain the optimal number of clusters using the Elbow or Silhouette method.
9	Employ a density-based clustering algorithm such as DBSCAN using a suitable open-access dataset to identify the clustering structure and noise points.
10	Analyze the performance of ensemble learning techniques such as Random Forest or Boosting on a suitable open-access dataset in comparison with a baseline machine learning model.
11	Evaluate the performance of reinforcement learning using the Q-Learning algorithm for a simple game or decision-making problem (e.g., Tic-Tac-Toe or Grid World), focusing on exploration–exploitation strategies and the optimal policy.

Savitribai Phule Pune University **Third Year of Computer science (2024 Course)**		
PCC-354-CSC: Software Engineering and Modelling Laboratory		
Teaching scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	**Practical**: 01	**Term Work:** 25 Marks **Oral :** 25 Marks

Companion Course: Object Oriented Programming, Data Structures & Database Management Systems

Course Objectives:

1. To apply software engineering principles to real-world problem statements.
2. To develop Software Requirement Specification (SRS) documents.
3. To design and model software systems using UML tools.
4. To implement testing strategies and quality assurance techniques.
5. To explore Agile practices, version control, and DevOps fundamentals.

Course Outcomes:

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

- **CO1:** Perform feasibility **analysis** and project estimation for software systems.
- **CO2: Develop** SRS document using standard IEEE format.
- **CO3: Design** UML diagrams using modeling tools.
- **CO4: Prepare** and **Execute** software test cases and quality plans.
- **CO5: Apply** Agile methodology and version control tools in software development.

Course Contents

Guidelines for Instructor's Manual

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

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis). Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal must be avoided. Use of DVD containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

.

Guidelines for Laboratory/Term Work Assessment

Term work evaluation shall be based on:

- Timely Completion – 5 Marks
- Quality of Documentation – 5 Marks
- Tool Usage & Implementation – 5 Marks
- Innovation & Problem Solving – 5 Marks
- Mini Project Performance – 5 Marks

Guidelines for Laboratory Conduction

Problem statement jointly decided by internal and external examiners. Evaluation criteria:

- Implementation – 40%
- Understanding – 30%
- Documentation – 20%
- Viva – 10%

Guidelines for Practical Examination

- Emphasis on experiential learning and skill development. Problem-based and case-based learning approach.
- Collaborative mini-project in teams of 3–4 students.
- Use of industry-relevant tools (StarUML, Visual Paradigm, GitHub). Continuous evaluation based on innovation and implementation

Sr. No.	List of Assignments (Part A) - Any FIVE , 6th and 7th Mandatory
1	Study and compare Waterfall Model, V-Model, Spiral Model, and Incremental Model. Select suitable life cycle model. Identify deliverables and justify model selection.
2	Identify the stakeholders involved in the selected software system, Prepare questionnaire and Develop usage scenarios
3	Study the concepts and practices of Scrum, Kanban, and Extreme Programming used in Agile software development. Analyze how these methodologies help in managing project tasks, improving collaboration, and delivering software incrementally. Based on a selected software system, write sample user stories to capture system requirements. Further, create a sample sprint plan and track tasks using the project management tool Jira to understand Agile workflow management.
4	Prepare IEEE-format SRS document. Include Functional and Non-functional Requirements.

5	Requirements Traceability & Validation. Prepare RTM and perform requirement validation.
6	Risk identification and mitigation plan preparation.
7	Estimate using LOC/FP and apply COCOMO. Prepare Gantt chart.
8	Agile Project Planning and Tracking using Jira Study Scrum, Kanban and Extreme Programming. Prepare sample sprint in jira. To understand and implement Agile project management practices using Jira tool.
Sr. No	**List of Assignments (Part B) Any 3**
09	Study and understand UML modeling and draw Use case diagram. Use the Unified Modeling Language (UML) to model selected system such as a Library Management System or Online Shopping System. Analyze the system requirements and represent them using Use case diagram.Also Draw Data Flow Diagram (DFD), Entity Relationship Diagram (ER Diagram).
10	UML Structural Modelling Draw Class, Package and Component Diagrams.
11	UML Behavioral Modelling Draw Sequence, Activity and State Diagrams.
12	Architectural Design Design Layered, Client-Server and Microservices architecture using 4+1 view model.
13	Software Testing – Test Case Design Write test cases to perform Black Box and White Box testing.
14	Manual and Automated Testing Perform manual testing and automated testing using tools (Selenium) Implement a simple Selenium WebDriver code in Java/Python to open Google Chrome, navigate to Google Search, and enter text in the search box.
15	Study DevOps and CI/CD Implementation Study the architecture and core concepts of DevOps and understand how CI/CD improve software delivery. Implement a simple CI/CD pipeline by creating a project repository using Git. Configure the pipeline to automatically build and test the application whenever code changes are pushed to the repository, demonstrating the basic workflow of automated software integration and deployment.
16	AI in Software Engineering / Low-Code Platform Use AI tool for code/testing OR develop app using low-code platform. Problem Identification, Feasibility Study, SRS Preparation, UML Modeling, Design, Implementation (optional prototype), Testing Plan, Agile Sprint Documentation, Final Report and Demonstration **Examples:** Library Management System, Online Food Ordering System, Hospital Management System, Learning Management System
List of Assignments (Part C) - Mini-Project (Any One-In Team of 3-4 Students)	
17	Problem Identification, Feasibility Study, SRS Preparation, UML Modeling, Design Implementation (optional prototype), Testing Plan, Agile Sprint Documentation, Final Report and Demonstration from any one (but not limited to) (a) Library Management System (b) Online Food Ordering System (c) Hospital Management System (d) Learning Management System

Savitribai Phule Pune University **Third Year of Computer Science (2024 Course)**		
PEC-361A-CSC : Soft Computing and Fuzzy Logic(Elective – II)		
Teaching scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	**CCE:** 30 Marks **End-Semester:** 70 Marks

Course Objectives:

1. To introduce the fundamentals of soft computing and its significance in modern computing systems.
2. To understand neural networks and their role in intelligent systems.
3. To explore fuzzy logic concepts and their applications in decision-making systems.
4. To study advanced fuzzy systems, neuro-fuzzy modeling, and optimization techniques.
5. To apply soft computing techniques to real-world problems in various domains.

Course Outcomes (COs):

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

- **CO1: Explain** the concept of soft computing and differentiate it from hard computing approaches.
- **CO2: Understand** and **apply** neural network principles and machine learning basics.
- **CO3: Analyze** fuzzy sets, fuzzy relations, and fuzzy inference systems.
- **CO4: Design** and implement fuzzy logic controllers and neuro-fuzzy models.
- **CO5: Apply** soft computing techniques such as genetic algorithms, PSO, ACO, and simulated annealing to solve real-world problems.

Course Contents

UNIT I: Introduction to Soft Computing and Neural Networks (7 Hours)

Evolution of computing, soft computing constituents, transition from conventional AI to computational intelligence, Basics of Neural networks, concept of computing systems, “soft” computing versus “hard” computing, characteristics of soft computing, applications of soft computing techniques.

UNIT II: Introduction to Fuzzy Logic (7 Hours)

Fuzzy set theory, fuzzy sets versus crisp sets, crisp relations and fuzzy relations, fuzzy systems: crisp logic and fuzzy logic, introduction and features of membership functions. Operations on fuzzy sets
Fuzzy rule-based systems: fuzzy propositions, formation, decomposition and aggregation of fuzzy rules, fuzzy reasoning, fuzzy inference systems, fuzzy decision making and applications of fuzzy logic.

UNIT III: Fuzzy Set Theory (7 Hours)

Fuzzy relations, rules, propositions and implications, fuzzy tolerance and equivalence relations, defuzzification techniques, λ-cuts for fuzzy relations, fuzzy max-min and max-product composition, fuzzy logic controller design, applications of fuzzy logic.

UNIT IV: Fuzzy Systems (7 Hours)

Fuzzy inference systems – Mamdani fuzzy models and Sugeno fuzzy models, fuzzy modeling and decision making, neuro-fuzzy modelling. Introduction to neuro-fuzzy systems, genetic algorithm-based backpropagation networks, fuzzy backpropagation, fuzzy logic controlled genetic algorithms, simplified fuzzy ARTMAP, case studies.

UNIT V: Applications of Soft Computing Techniques (7 Hours)

Applications of fuzzy logic in pattern recognition and character recognition, applications of evolutionary computing in image processing and computer vision, soft computing in mobile ad-hoc networks, information retrieval and semantic web, software engineering. Optimization techniques: simulated annealing, tabu search, ant colony optimization (ACO), particle swarm optimization (PSO).

Learning Resources

Text Books:

1. S. Rajasekaran and G. A. Vijayalakshmi Pai, *Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis and Applications*, PHI Learning.
2. S. N. Sivanandam and S. N. Deepa, *Principles of Soft Computing*, Wiley India.
3. K. T. Atanassov, *Intuitionistic Fuzzy Sets: Theory and Applications*, Springer.

Reference Books:

1. Timothy J. Ross, *Fuzzy Logic with Engineering Applications*, Wiley.
2. George J. Klir and Bo Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall.
3. Simon Haykin, *Neural Networks and Learning Machines*, Pearson Education.
4. David E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Pearson.
5. Xin-She Yang, *Nature-Inspired Optimization Algorithms*, Elsevier.
6. Marco Dorigo and Thomas Stützle, *Ant Colony Optimization*, MIT Press.

Savitribai Phule Pune University **Third Year of Computer Science (2024 Course)**		
PEC-361B-CSC : Augmented Reality and Virtual Reality (Elective – II)		
Teaching scheme	**Credits**	**Examination Scheme**
Theory : 03Hours/Week	03	**CCE:** 30 Marks **End-Semester:** 70 Marks

Course Objectives

1. To understand the fundamentals and evolution of Augmented Reality (AR) and Virtual Reality (VR) technologies.
2. To learn basic 3D graphics and scene representation concepts required for immersive environments.
3. To study interaction techniques and user interface design principles used in immersive systems.
4. To understand augmented reality tracking, spatial mapping and mobile AR frameworks.
5. To analyze applications, challenges and ethical considerations of AR/VR systems.

Course Outcomes (CO)

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

- **CO1: Explain** the concepts, components and applications of AR/VR systems.
- **CO2: Apply** basic 3D scene representation and transformation concepts in immersive environments.
- **CO3: Describe** interaction techniques and user interface design principles in virtual environments.
- **CO4: Explain** tracking, registration and spatial mapping techniques used in augmented reality systems.
- **CO5: Analyze** real-world applications, performance issues and ethical considerations in AR/VR systems.

Course Contents

UNIT I: Introduction to AR and VR (08 Hours)

Introduction and evolution of immersive technologies, definitions of AR, VR, MR and XR, components of AR/VR systems, hardware devices such as head-mounted displays (HMDs), controllers and sensors, human perception, immersion and presence, field of view and frame rate considerations, motion sickness issues in VR, applications of AR/VR in gaming, healthcare, education and industry, challenges and future trends.

Case Study: Virtual Reality in Medical Training

Study of VR-based surgical simulation systems used in medical education to train surgeons in a safe and controlled virtual environment.

UNIT II: Representing the Virtual World (08 Hours)

Basics of 3D graphics for immersive environments, representation of virtual environments, 3D geometric objects and scene structures, coordinate systems (world, local, camera), geometric transformations (translation, rotation, scaling), lighting and materials in virtual scenes, texture mapping concepts, spatial audio and 3D sound, haptic feedback concepts, avatars and virtual objects.

Case Study: Design of Virtual Environments in Game Engines

Study of how game engines such as Unity create immersive 3D environments using models, lighting, textures and spatial audio**.**

UNIT III: Interaction and Navigation in Virtual Environments (08 Hours)

Interaction techniques in virtual environments, navigation and locomotion methods, gesture-based interaction, controller-based interaction, natural user interfaces, VR user interface design principles, immersive UX design, cyber sickness reduction techniques, performance and latency considerations.

Case Study: Interaction Design in VR Gaming

Analysis of interaction mechanisms in the VR rhythm game Beat Saber focusing on gesture-based interaction and user engagement.

UNIT IV: Augmented Reality Systems and Tracking (08 Hours)

Principles of augmented reality, marker-based tracking techniques, marker-less tracking approaches, concept of simultaneous localization and mapping (SLAM), overview of sensor fusion, spatial mapping and plane detection, AR content registration and anchoring, mobile AR frameworks such as ARCore and ARKit.

Case Study: Augmented Reality in Retail Visualization

Study of AR applications used by furniture retailers that allow customers to visualize products in their homes using mobile devices and marker-less AR.

UNIT V: Advanced Immersive Systems and Applications (8 Hours)

Mixed reality systems, WebXR and browser-based immersive applications, collaborative virtual environments, industrial and healthcare AR/VR applications, ethical, privacy and security issues in immersive systems, future trends in spatial computing.

Case Study: Virtual Reality for Industrial Training

Analysis of VR-based safety and equipment training used in manufacturing industries to simulate real-world operations.

Learning Resources

Text Books:

1. Understanding Virtual Reality: Interface, Application and Design — William R. Sherman and Alan B. Craig, Morgan Kaufmann.
2. Augmented Reality: Principles and Practice — Dieter Schmalstieg and Tobias Hollerer, Addison-Wesley.

Reference Books:

1. Learning Virtual Reality — Tony Parisi, O'Reilly Media.
2. Augmented Reality for Developers — Jonathan Linowes, Packt Publishing.
3. Virtual Reality Technology — Grigore Burdea and Philippe Coiffet, Wiley.
4. 3D User Interfaces: Theory and Practice — Doug Bowman, Addison-Wesley.

Savitribai Phule Pune University **Third Year - Computer Science (2024 Pattern)**		
PEC-361C-CSC : Natural Language Processing (Elective – II)		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03Hours/Week	03	**CCE:** 30 Marks **End-Semester:** 70 Marks

Prerequisite Course: Database Management System, Machine Learning.

Course Objective:

1. To provide a comprehensive understanding of the theoretical foundations of Natural Language Processing.
2. To equip students with practical skills in designing and implementing NLP pipelines for real-world applications.
3. To develop proficiency in applying statistical, machine learning, and deep learning methods to NLP tasks.
4. To expose students to state-of-the-art pre-trained language models and their fine-tuning strategies.
5. To sensitize students to ethical challenges and responsible practices in deploying NLP systems.

Course Outcomes: Students should be able to:

- **CO1: Understand** the fundamental concepts of Natural Language Processing including linguistic structures, text preprocessing, and morphological analysis.
- **CO2: Apply** statistical language models, Hidden Markov Models, and sequence labeling techniques for POS tagging and Named Entity Recognition.
- **CO3: Analyze** syntactic parsing algorithms, semantic role labeling, and coreference resolution methods for NLP systems.
- **CO4: Design** and implement Transformer-based architectures for various NLP tasks.
- **CO5: Evaluate** and build real-world NLP applications.

Course Contents

Unit I - Introduction to NLP and Text Processing (08 Hours)

Introduction to Natural Language Processing: definition, history and evolution, scope and applications, NLP pipeline, Linguistic levels: phonology, morphology, lexicology, syntax, semantics, pragmatics and discourse, Phonetic Transcription, Articulatory Phonetics, Ambiguities in Phonology, Morphological analysis: inflectional and derivational morphology, finite state automata and finite state transducers, Ambiguities in Morphology, Word representation: Bag-of-Words model, TF-IDF, co-occurrence matrix, Text normalization: stopword removal, sentence segmentation, stemming, lemmatization.

Case Study: Resume Parsing System for an HR Recruitment Platform

Unit II - Language Models and Sequence Labeling (08 Hours)

N-gram language models: unigram, bigram and trigram models, maximum likelihood estimation, Smoothing techniques, Hidden Markov Models (HMM): Markov assumption, emission probabilities, transition probabilities, HMM algorithms: forward algorithm, Viterbi decoding, Baum-Welch training, Parts-of-Speech (POS) tagging: rule-based taggers, stochastic taggers, transformation-based taggers, Named Entity Recognition (NER).

Case Study: Named Entity Recognition for Clinical Health Records.

Unit III -Parsing and Semantic Analysis (08 Hours)

Context-Free Grammars (CFG): formal definition, derivations, parse trees, ambiguity, Parsing algorithms: CYK algorithm, Earley parser, chart parsing, Probabilistic Context-Free Grammars (PCFG): rule probabilities, probabilistic CYK, Dependency parsing: arc-standard and arc-eager transition systems, graph-based parsing, Lexical semantics: WordNet, word sense disambiguation (WSD), Semantic role labeling (SRL): FrameNet, PropBank, argument structure, Coreference resolution: mention detection, entity linking, anaphora resolution, pronoun resolution.

Case Study: Semantic Parsing for an Automated Customer Support System

Unit IV Role of Generative AI in NLP (08 Hours)

Word embedding: Word2Vec (CBOW and Skip-gram), GloVe, FastText, Transformer architecture: multi-head self-attention, positional encoding, feed-forward layers, Pre-trained language models: BERT, GPT, RoBERTa; pre-training objectives and fine-tuning strategies, Recent advances: Large Language Models (LLMs), prompt engineering, in-context learning, instruction tuning.

Case Study: Sentiment Analysis of Product Reviews using BERT Fine-Tuning

Unit V NLP Applications and Advanced Topics (08 Hours)

Machine Translation: statistical machine translation (SMT), neural machine translation (NMT), evaluation metrics: BLEU, METEOR, TER, Information Extraction: relation extraction, event detection, Question Answering systems: extractive QA (SQuAD benchmark), generative QA, knowledge graph-based QA, Dialogue systems: rule-based chatbots, retrieval-based models, task-oriented dialogue, generative dialogue systems, Text summarization: extractive summarization, abstractive summarization, evaluation using ROUGE metric, Cross-lingual NLP: multilingual models, low-resource language processing, transfer learning, Ethics in NLP: bias in language models, fairness, hallucination in LLMs, responsible AI.

Case Study: Building a Multilingual Legal Document QA System

Learning Resources

Text Books:

1. Daniel Jurafsky and James H. Martin, Speech and Language Processing, 3rd Edition, Pearson Education.
2. Steven Bird, Ewan Klein and Edward Loper, Natural Language Processing with Python, O'Reilly Media.
3. Christopher D. Manning and Hinrich Schutze, Foundations of Statistical Natural Language Processing, MIT Press.

Reference Books: -

1. Jacob Eisenstein, Introduction to Natural Language Processing, MIT Press, 2019.
2. Yoav Goldberg, Neural Network Methods for Natural Language Processing, Morgan and Claypool Publishers.
3. Eugene Charniak, Statistical Language Learning, MIT Press.

Savitribai Phule Pune University **Third Year of Computer science (2024 Course)**		
PEC-361D-CSC- Quantum Computing (Elective II)		
Teaching scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	**CCE:** 30 Marks **End Semester :** 70 Marks

Prerequisites Courses: Engineering Mathematics, Data Structures and Algorithm, Python Programming

Course Objectives:

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

Course Outcomes:

After successful completion of the course, learner will be able:

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

Course Contents

Unit I - Foundations of Quantum Computing (08 Hours)

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

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

Unit II - Mathematical Foundations for Quantum Computing, Qubits and Quantum Gates. (08 Hours)

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

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

Unit III Multi-Qubit Systems , Quantum Circuits and Building Blocks for Quantum Program (08 Hours)

Architecture of a Quantum Computing platform, Details of q-bit system of information representation, Multi-qubits States, Quantum superposition of qubits (valid and invalid superposition, Multi-qubit representation, Tensor product concept, Controlled gates (CNOT, CZ) ,Entanglement and Bell states ,Quantum circuit model, No-cloning theorem ,Quantum teleportation.

Case Study: Design and Analysis of a Two-Qubit Entangled System for Quantum Teleportation

Unit IV Basic Quantum Algorithms (08 Hours)

Quantum parallelism, Deutsch algorithm, Deutsch–Jozsa algorithm, Grover search algorithm, Shor algorithm, Quantum error correction using repetition codes 3 qubit codes, Shor's 9 qubit error correction Code.

Case Study: Speedup Through Quantum Algorithms -From Deutsch to Grover

Unit V Quantum Programming and Applications (08 Hours)

Introduction to Qiskit / Cirq ,Writing basic quantum programs ,Circuit simulation, Result visualization, Quantum cryptography basics, Quantum Machine Learning and Quantum AI, Quantum Neural Networks, Quantum Natural Language Understanding, Quantum Cryptography, Application Domains for quantum image processing overview, Current challenges and future scope, OSS Toolkits for implementing Quantum program, IBM quantum experience, Microsoft Q Rigetti PyQuil (QPU/QVM).

Case Study: Building and Deploying a Quantum Program Using Open-Source Quantum Toolkits

Learning Resources

Text Books:

1. Michael A. Nielsen, Quantum Computation and Quantum Information||, Cambridge University Press.
2. David McMahon,Quantum Computing Explained, Wiley, 2008.
3. Qiskit textbook https://qiskit.org/textbook-beta/
4. Vladimir Silva, Practical Quantum Computing for Developers, 2018 .
5. Wittek, "Quantum Machine Learning (What Quantum Computing Means to Data Mining)", Peter University of Boras, Sweden - Elsevier Publications
6. Andreas Winchert, "Principles of Quantum Artificial Intelligence",Instituto Superior Técnico Universidade de Lisboa, Portugal - World Scientific Publishing, British Library Cataloguing-in Publication Data

Reference Books:

1. Bernard Zygelman, A First Introduction to Quantum Computing and Information,2018
2. Supriyo Bandopadhyay and Marc Cahy,Introduction to Spintronics||, CRC Press, 2008
3. The Second Quantum Revolution: From Entanglement to Quantum Computing and Other SuperTechnologies, Lars Jaeger
4. La Guardia, Giuliano Gladioli Quantum Error correction codes Springer,2021
5. Press Stephen Kan, "MetricsandModelsinSoftwareQualityEngineering||,Pearson,ISBN-10:0133988082; ISBN-13:978-0133988086 2.
6. Michael A. Nielsen, "Quantum Computation and Quantum Information", Cambridge University PressStephen Kan, Metrics and Models in Software Quality Engineering, Pearson, ISBN-10:0133988082; ISBN-13: 978-0133988086 3.
7. David McMahon, "Quantum Computing Explained", Wiley .

8. Microsoft Quantum Development Kithttps://www.microsoft.com/enus/quantum/developmentkit Forest SDK PyQuil: https://pyquil.readthedocs.io/en/stable/

9. Amazon Bracket Documentation on AWS:https://aws.amazon.com/braket/ 7 D-Wave Systems Documentation:

E Books: -

1. https://docs.dwavesys.com/docs/latest/index.html

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

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

Savitribai Phule Pune University **Third Year of Computer Science (2024 Course)**		
PEC-362A-CSC- Data Analytics and Visualization (Elective III)		
Teaching scheme	**Credits**	**Examination Scheme**
Theory : 03Hours/Week	03	**CCE:** 30 Marks **End-Semester:** 70 Marks

Prerequisites Courses: Database Management Systems (PCC-251-CSC)

Course Objectives: The course aims to:

1. To understand fundamental concepts of data analytics, data types, and the analytics lifecycle.
2. To develop skills in data collection, cleaning, preprocessing, and transformation using DBMS and analytical techniques.
3. To apply statistical and mathematical concepts for effective data analysis.
4. To perform exploratory data analysis (EDA) and visualize data using appropriate tools and techniques.
5. To use analytical models and modern tools (Python, Tableau, Power BI) to extract insights and support data-driven decision-making.

Course Outcomes:

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

- **CO1: Understand** fundamental concepts of data analytics and data types.
- **CO2: Apply** preprocessing techniques to clean and prepare data.
- **CO3: Analyze** and **Interpret** data using exploratory data analysis techniques through statistical measures and visualization methods.
- **CO4: Apply** basic analytical models for pattern discovery.
- **CO5: Design** and **develop** dashboards using modern visualization tools and Interpret and communicate insights effectively using visualization techniques.

Course Contents

Unit I - Introduction to Data Analytics (08 Hours)

Data: Types and Sources (Structured, Semi-structured, Unstructured) , Data Analytics Lifecycle (Discovery, Preparation, Model Planning, Model Building, Evaluation) , Role of Data Analytics in Decision Making , Basic Statistics: Mean, Median, Mode, Variance, Standard Deviation , Introduction to Probability Concepts (from Discrete Mathematics) Overview of Tools: Python, R, SQL

Case study: Global Innovation Social Network and Analysis (GINA).

Unit II - Data Collection and Preprocessing (08 Hours)

Data Acquisition from Databases (SQL Queries, Joins, Aggregation) , Data Cleaning Techniques (Missing Values, Noise, Outliers), Data Integration and Transformation ,Data Reduction Techniques ,Data Normalization and Scaling , Introduction to ETL (Extract, Transform, Load)

Case Study: Designing an ETL Pipeline for Data Integration and Transformation

Unit III Exploratory Data Analysis (EDA) and Basic Visualization (08 Hours)

Concept of Exploratory Data Analysis, Data Summarization Techniques, Visualization Techniques: Bar Chart, Line Chart, Pie Chart Histogram, Box Plot, Scatter Plot Correlation and Covariance, Use of Python Libraries: Matplotlib, Seaborn , Introduction to Dashboards.

Case Study: Statistical Analysis and Visualization Using Python Tools

Unit IV - Advanced Analytics and Visualization (08 Hours)

Regression Analysis (Linear, Multiple), Classification Techniques (Basic Concepts) ,Clustering (K-Means – Conceptual) Time-Series Analysis Basics , Advanced Visualization:- Heatmaps ,Geospatial Visualization, Network Graphs Interactive Visualization Concepts.

Case Study: Time-Series Analysis and Demand Prediction in E-Commerce Platforms

Unit V - Tools, Applications and Case Studies (08 Hours)

Data Visualization Tools: Tableau, Power BI, Dashboard Design Principles, Storytelling with Data, Big Data Analytics Overview , Real-Time Data Visualization ,Case Studies in Business, Healthcare, Finance ,Mini Project / Practical Implementation

Case Study: End-to-End Data Visualization Project Using Tableau and Power BI

Learning Resources

Text Books:

1. Data Analytics Made Accessible – Anil Maheshwari Maximize Personality, WILEY INDIA,
2. Hands-On Exploratory Data Analysis with Python – Suresh Kumar Mukhiya & Usman Ahmed
3. Pattern Recognition and Machine Learning – Christopher M. Bishop Learning Tableau – Joshua N. Milligan

Reference Books:

1. Anil Maheshwari -Data Analytics Made Accessible , McGraw Hill ISBN-13: 978-9355324559
2. Foster Provost and Tom Fawcett - Data Science for Business, O'Reilly Media ISBN: 978-1-4493-6132-7\
3. Venkatesh Ganti & Anish Das Sarma- Data Cleaning: A Practical Perspective, Morgan & Claypool Publishers ISBN- 978-1608456772
4. Suresh Kumar Mukhiya & Usman Ahmed- Hands-On Exploratory Data Analysis with Python, Packt Publishing ISBN- 978-1789537253
5. Andreas C. Müller & Sarah Guido- Introduction to Machine Learning with Python- O'Reilly Media (2016)-ISBN 9352134575

E-Books:

1. **Introduction to Data Science** – Jeffrey M. Stanton Creative Commons (self-published through Syracuse University) , **ISBN:** 9781506377513
2. **https://surface.syr.edu/cgi/viewcontent.cgi?article=1165&context=istpub**

Links to online SWAYAM/NPTEL Courses:

1. **https://onlinecourses.nptel.ac.in**
2. **https://onlinecourses.nptel.ac.in/noc26_cs86/preview**

Savitribai Phule Pune University **Third Year - Computer Science (2024 Pattern)**		
PEC-362B-CSC - Cryptography and Network Security (Elective III)		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	**CCE:** 30 Marks **End-Semester:** 70 Marks

Prerequisite Courses: Computer Networks, Cloud Computing
Companion Course: PEC363-Programme Elective Course Lab 3
Course Objectives: The course aims to:

1. To understand the fundamental principles of **Information Security and the CIA Triad.**
2. To explore the mathematical foundations of **Symmetric and Asymmetric cryptography.**
3. To analyse and implement Integrity and Authentication mechanisms using **Hash functions and Digital Signatures.**
4. To evaluate various **Network Security protocols (IPSec, TLS)** used to secure data in transit.
5. To study System-level defenses, including **Firewalls, IDS, and the legal framework** governing cybersecurity.

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

- **CO1: Analyse** various security attacks, services, and mechanisms to identify vulnerabilities within a network model.
- **CO2: Compare** the mathematical logic and computational complexity of symmetric and asymmetric cryptographic algorithms.
- **CO3: Illustrate** the role of cryptographic hash functions and digital signatures in achieving
- data integrity and non-repudiation.
- **CO4: Explain** the architectural workings of network security protocols such as SSL/TLS and IPSec for securing web and network communications.
- **CO5**: **Examine** the principles of Firewalls and IDS while interpreting the legal and ethical implications of the IT Act 2000.

Course Contents

Unit I - Foundations of Security and Cryptography (08 Hours)

Introduction to Security, The Need for Security, Security Approaches, Principles of Security, Types of Security Attacks, Security Services, Security Mechanisms, A Model for Network Security, Plaintext and Ciphertext, Substitution Techniques, Transposition Techniques, Encryption and Decryption, Symmetric and Asymmetric Key Cryptography, Steganography, Key Range and Key Size, Possible Types of Attacks, Lightweight Cryptography (LWC), Resource-Constrained Environments, NIST LWC Standards.

Case Study: The Caesar Cipher in Roman Military Communication

Unit II Symmetric and Asymmetric Cryptographic Algorithms (08 Hours)

Symmetric Key Ciphers: Block Cipher principles (Feistel and Substitution-Permutation Networks), Data Encryption Standard (DES), Advanced Encryption Standard (AES), Blowfish, RC5, International Data Encryption Algorithm (IDEA), Block cipher modes of operation (ECB, CBC, CFB, OFB, CTR), Stream ciphers (RC4 and Introduction to ChaCha20), Limitations of legacy ciphers.**Asymmetric Key Ciphers:** Principles of public key cryptosystems, RSA algorithm, Elgamal Cryptography, Diffie-Hellman Key Exchange, Knapsack Algorithm, Key management and distribution in asymmetric systems.

Case Study: The WhatsApp End-to-End Encryption Model

Unit III Hash Functions, Digital Signatures, and Key Management (08 Hours)

Integrity and Authentication: One-way Hash Functions, Properties of Hash Functions, Message Authentication Codes (MAC), Hash Algorithms: MD5, SHA-1, SHA-256/512, HMAC, Digital Signatures: Principles, Digital Signature Standard (DSS).**Key Management and Infrastructure**: Symmetric Key Distribution using Symmetric and Asymmetric Encryption, Kerberos, X.509 Certificates, Public Key Infrastructure (PKI), Trust Models, Key Revocation.

Case Study: The "Flame" Malware and MD5 Collisions

Unit IV Network Security and Transport Layer Protection (08 Hours)

IP Security: Architecture, Authentication Header (AH), Encapsulating Security Payload (ESP), Security Associations, Key Management (IKE). **Transport Layer Security:** SSL/TLS Architecture, Handshake Protocol, Change Cipher Spec Protocol, Alert Protocol, Record Protocol, HTTPS, SSH. **Web Security:** Secure Electronic Transaction (SET), Web Security threats and countermeasures. **Email Security:** PGP (Pretty Good Privacy), S/MIME (Secure/Multipurpose Internet Mail Extensions).

Case Study: The Great "Heartbleed" Vulnerability

Unit V System Security and Defensive Countermeasures (08 Hours)

Intrusion Detection: Intruders and the Intrusion Cycle, Intrusion Detection Systems (IDS): Host-based IDS (HIDS), Network-based IDS (NIDS), Signature-based vs. Anomaly-based Detection, Honeypots. **Malicious Software:** Nature of Malware, Viruses: Lifecycle and Classification, Worms, Trojans, Rootkits, Botnets, Ransomware, Anti-virus Techniques and Countermeasures.**Firewalls:** Firewall Design Principles, Types of Firewalls: Packet Filtering, Application-Level Gateway, Circuit-Level Gateway, Firewall Configurations: Bastion Host, Screened Subnet (DMZ).**Legal and Ethical Issues:** Cybercrimes and Cyber Laws, Introduction to IT Act 2000, Ethical Hacking Concepts, Privacy and Intellectual Property in Cybersecurity.

Case Study: Intrusion Detection and Honeypots

Learning Resources

Text Books:

1. William Stallings, Lawrie Brown, “Computer Security Principles and Practice”, 3rd_Edition, Pearson
2. William Stallings, “Cryptography and Network Security Principals and Practice”, Fifth edition, Pearson
3. Nina Godbole, Sunit Belapure, “Cyber Security”, Wiley, ISBN: 978-81-265-2179-1
4. Behrouz A. Forouzan & Debdeep Mukhopadhyay, Cryptography and Network Security, McGraw Hill Education.

Reference Books

1. Atul Kahate,” Cryptography and Network Security”, 3e, McGraw Hill Education
2. Prakash C. Gupta, “Cryptography and Network Security”, PHI
3. V.K. Pachghare, “Cryptography and Information Security”, PHI Learning
4. Bernard Menezes, “Network Security and Cryptography”, Cengage Learning India, 2014, ISBN No.: 8131513491
5. **Christof Paar and Jan Pelzl**, *Understanding Cryptography*, Springer. (Highly recommended for the mathematical theory behind Units 2 and 3).
6. **Bernard Menezes**, *Network Security and Cryptography*, Cengage Learning.
7. **V.K. Pachghare**, *Cryptography and Information Security*, PHI Learning. (Aligned with the Indian Engineering academic structure).

MOOC / NPTEL/YouTube Links

1. Cryptography I – Offered by Stanford University (Coursera)
2. Cybersecurity Fundamentals – Offered by University of Maryland (Coursera)
3. Network Security – Offered by Georgia Tech (Udacity / Online Platforms)
4. Recommended course of NPTEL: 1. Cryptography and Network Security – Prof. Debdeep Mukhopadhyay, IIT Kharagpur 2. Network Security – Prof. Dheeraj Sanghi, IIT Kanpur

E-Books:

1. **NIST Special Publication 800-175B (Rev. 1):** *Guideline for Using Cryptographic Standards in the Federal Government: Cryptographic Mechanisms*
2. **IETF RFC 8446:** *The Transport Layer Security (TLS) Protocol Version 1.3*

MOOC Courses:

1. **NPTEL:** *Introduction to Cryptology by Prof. Sugata Gangopadhyay (IIT Roorkee).*
2. **NPTEL:** *Foundations of Cryptography by Prof. Ashish Choudhury (IIIT Bangalore).*
3. **SWAYAM:** *Cyber Security by Prof. G. Padmavathi.*

Savitribai Phule Pune University **Third Year of Computer Science (2024 Course)**		
PEC-362C-CSC: Blockchain Technology (Elective-III)		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Weeks	03	**CCE:** 30 Marks **End Semester :** 70 Marks

Prerequisite Courses: Computer Network

Companion Course: Elective-III Laboratory

Course Objectives: The course aims to:

1. To understand the fundamental concepts of Blockchain Technology and, and working principles of Blockchain technology.
2. To learn the role of cryptography in ensuring security, privacy, and integrity in Blockchain systems.
3. To study various consensus algorithms used in distributed Blockchain networks.
4. Evaluate the design and implementation of smart contracts and decentralized applications by assessing their efficiency, security, scalability, and suitability for real-world Blockchain-based solutions.
5. To explore Explore real-world applications of Blockchain technology and examine associated challenges by investigating use cases, identifying limitations, analyzing security issues, and understanding emerging trends and future developments in the Blockchain ecosystem

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

- **CO1: Understand** the fundamental concepts and working principles of Blockchain Technology.
- **CO2: Explain** the role of cryptography in ensuring security, privacy, and integrity in Blockchain systems.
- **CO3: Analyze** various consensus algorithms used in distributed Blockchain networks.
- **CO4: Evaluate** the design and implementation of smart contracts and decentralized applications by assessing their efficiency, security, scalability, and suitability for real-world Blockchain-based solutions.
- **CO5: Analyze** real-world applications of Blockchain technology and examine associated challenges.

Course Contents

Unit I - Introduction to Blockchain (06 Hours)

Basic ideas behind Blockchain, Evolution of digital currency, Overview of Bitcoin, Centralized vs decentralized systems , Basic architecture of Block chain(blocks, transactions, Merkle trees), Types of block chains: Public, Private, Consortium, Private vs public block chain.

Case Studies: Introduction to Block chain – Supply Chain Management

***Mapping of Course Outcomes for Unit I:** CO1

Unit II - Cryptography in Blockchain (08 Hours)

Introduction to cryptographic concepts, Symmetric Key Cryptography and Asymmetric Key Cryptography, Hash functions (SHA-256) Public key and private key cryptography Digital Signature Algorithm (DSA), Elliptic Curve Cryptography (ECC), Merkel Trees.

Case Studies: Case Study: Role of Cryptography in Blockchain Technology

***Mapping of Course Outcomes for Unit II:** CO2

Unit III - Consensus Mechanisms and Distributed Systems (08 Hours)

Distributed ledger technology (DLT), Consensus algorithms: Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS), Smart Contracts, Byzantine Fault Tolerance (BFT), Comparison of consensus algorithms, Mining process, rewards, and incentives, Forks: soft fork vs hard fork, Network attacks: 51% attack, Sybil attack, double spending

Case Studies: Distributed Ledger Technology, Consensus Mechanisms, and Smart Contracts in Blockchain

***Mapping of Course Outcomes for Unit III:** CO3

Unit IV - Smart Contracts and Platforms (06 Hours)

Introduction to smart contracts, Ethereum architecture and components (EVM, Gas, Nodes), Solidity programming basics (syntax, variables, functions), Writing, compiling, and deploying smart contracts, Tools: Remix IDE, Truffle, Ganache, Decentralized Applications (DApps): architecture and workflow

Case Studies: Smart Contracts and DApps on Ethereum (Online Voting System)

***Mapping of Course Outcomes for Unit IV:** CO4

Unit V - Applications, Challenges, and Future Trends (06 Hours)

Blockchain use cases: finance, supply chain, healthcare, voting, Decentralized Finance (DeFi), NFTs (Non-Fungible Tokens), Security, privacy, and scalability challenges, Regulatory and legal aspects, Future of blockchain technology.

***Mapping of Course Outcomes for Unit V:** CO5

Learning Resources

Text Books:

1. Blockchain Basics – Daniel Drescher.
2. Mastering Bitcoin – Andreas M. Antonopoulos.
3. Artemis Caro, "Blockchain: The Beginners Guide to Understanding the Technology Behind Bitcoin & Crypto currency".
4. Scott Marks, "Blockchain for Beginners: Guide to Understanding the Foundation and Basics of the Revolutionary Blockchain Technology", Create Space Independent Publishing Platform

Reference Books:

1. Mark Watney, "Blockchain for Beginners".
2. Cryptography and Network Security – William Stallings
3. Mastering Ethereum – Andreas M. Antonopoulos *&* Gavin Wood
4. Alwyn Bishop, "Blockchain Technology Explained".
5. The Basics of Bitcoins and Blockchains – Antony Lewis,
6. Bitcoin and Cryptocurrency Technologies – *by* Arvind Narayanan *et al.*

E- Books:

1. Mastering Bitcoin, 3rd Edition: Programming the Open Blockchain - Free Computer, Programming,

MOOC/NPTEL/YouTube Links:

1.NPTEL Course **"Introduction to Blockchain Technology & Applications"**
https://nptel.ac.in/courses/106/104/106104220/
2.NPTEL Course on **"Blockchain Architecture &UseCases"**
https://nptel.ac.in/courses/106/105/106105184/

YouTube Links:

1. https://youtu.be/yubzJw0uiE4?si=PgUOp289KVacrZwh
2. https://www.youtube.com/watch?v=pyaIppMhuic
3. https://youtu.be/ZE2HxTmxfrI?si=LG0B7gqY9uqZGEto
4. https://youtu.be/M3RYFJsLC_A?si=y4pHUz_SKlmK8-Rc
5. https://youtu.be/SBZ20jnzXdY?si=gwrAzOuDDZN-uHdc

PEC-362D-CSC: Image Processing (Elective-III)		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory: 03 Hours/Weeks	03	**CCE:** 30 Marks **End Semester :** 70 Marks

Prerequisite Courses: Discrete Mathematics, Data Structures, Programming Skills

Companion Course: Image Processing Lab

Course Objectives: The course aims to:

1. To explain the fundamental concepts of digital image representation, sampling, and quantization.
2. To apply spatial and frequency domain techniques for image enhancement and filtering.
3. To analyze image degradation models and implement restoration techniques for noise removal.
4. To design image segmentation and morphological algorithms for object detection and feature extraction.
5. To evaluate image compression and color image processing techniques for real-world engineering applications.

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

- **CO1: Illustrate** and interpret fundamental concepts of digital image formation, sampling, and quantization.
- **CO2: Apply** spatial domain enhancement and filtering techniques for image improvement.
- **CO3: Analyze** and implement frequency domain processing techniques for image enhancement and restoration.
- **CO4: Design** and implement image segmentation and morphological algorithms for object detection and feature extraction.
- **CO5: Evaluate** image compression and color image-processing techniques for real-world engineering applications.

Course Contents

Unit I - Digital Image Fundamentals and Mathematical Foundations [08 Hours]

Digital Image Processing – Background and Applications, Fundamental Steps in Image Processing, Elements of Visual Perception, Image Sensing and Acquisition, Image Sampling and Quantization, Spatial Resolution and Intensity Resolution, Basic Relationships between Pixels Mathematical Tools:2D Signals and Systems, 2D Convolution, Linear and Nonlinear Operations.

Case Study: Smart Traffic Density Estimation using Digital Image Processing

***Mapping of Course Outcomes for Unit I:** CO1

Unit II - Image Enhancement in Spatial Domain (08 Hours)

Intensity Transformations and Log Transform, Power-Law Transformations, Histogram Equalization and Histogram Matching, Spatial Filtering: Linear Filtering (Averaging, Gaussian), Order Statistics Filters (Median, Max, Min), Sharpening Filters: Laplacian, Gradient Operators (Sobel, Prewitt), Unsharp Masking and High-Boost Filtering.

Case Study: Low-Light Image Enhancement and Edge Detection for Night Surveillance

***Mapping of Course Outcomes for Unit II:** CO2

Unit III - Frequency Domain Processing (08 Hours)

Fourier Series and Fourier Transform Review, 2D Discrete Fourier Transform (DFT), Properties of DFT, Frequency Domain Filtering, Ideal Filters, Butterworth Filters, Gaussian Filters, Laplacian in Frequency Domain, Homomorphic Filtering

Case Study: Removal of Periodic Noise in Satellite Images using Frequency Domain Processing

***Mapping of Course Outcomes for Unit III:** CO3

Unit IV -Image Restoration, Segmentation and Morphological Processing (08 Hours)

Image Restoration: Image Degradation/Restoration Model, Noise Models (Gaussian, Salt & Pepper, Periodic), Inverse Filtering, Wiener Filtering, **Image Segmentation:** Edge Detection (Roberts, Sobel, Prewitt, Canny), Thresholding (Global & Otsu's Method), Region-Based Segmentation, Introduction to Watershed Algorithm

Morphological Image Processing: Structuring Elements, Dilation and Erosion, Opening and Closing, Morphological Reconstruction

Case Study: Automatic Vehicle Number Plate Recognition

***Mapping of Course Outcomes for Unit IV:** CO4

Unit V - Image Compression and Color Image Processing (08 Hours)

Image Compression: Fundamentals of Compression, Redundancy and Information Theory Basics, Lossless Compression (Huffman Coding), Lossy Compression (JPEG), Discrete Cosine Transform (DCT), Introduction to Wavelet Transform Color Image Processing: Color Models (RGB, CMY, HSV, YCbCr), Color Transformations, Pseudo Color Processing, Applications in Medical Imaging, Surveillance, and Computer Vision.

Case Study: Facial Recognition System for Computer Vision Applications

***Mapping of Course Outcomes for Unit V:** CO5

Learning Resources
Text Books:

1. Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods Publisher: Pearson Education / Prentice Hall ,Edition: 3rd / 4th Edition,ISBN: 978-0131687288.
2. Anil K. Jain, Fundamentals of Digital Image Processing, Pearson.

Reference Books:

1. Fundamentals of Digital Image Processing by Anil K. Jain, Publisher: Prentice Hall of India (PHI) , Edition: 1st Edition
2. Digital Image Processing by S. Jayaraman, S. Esakkirajan and T. Veerakumar Publisher: McGraw-Hill Education (India)
3. Image Processing, Analysis, and Machine Vision by Milan Sonka, Vaclav Hlavac and Roger Boyle, Publisher: Cengage Learning Edition: 4th Edition

E-Books:

https://openlibrary.org/books/OL2036436M/Fundamentals_of_digital_image_processing
https://openlibrary.org/books/OL17236688M/Fundamentals_of_digital_image_processing

MOOC Courses:

NPTEL: *Computer Vision, IIT Kharagpur* *https://nptel.ac.in/courses/106105216*
Coursera (Northwestern University), Fundamentals of Digital Image and Video Processing
https://www.coursera.org/learn/image-processing/?utm_source=chatgpt.com

Savitribai Phule Pune University **Third Year of Computer Science (2024 Course)**		
PEC-363-CSC: Elective –III Laboratory		
Teaching scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	01	**Term Work** : 25 Marks

Guidelines for Instructor's Manual

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

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis). Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal must be avoided. Use of DVD containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

Guidelines for Laboratory /Term Work Assessment

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

Guidelines for Laboratory Conduction – Data Analytics and Visualization

· Operating System recommended: - 64-bit Open source Linux or its derivative
· Programming tools recommended: - JAVA/Python/R/Scala/Tableau

List of Assignment - Group A (Any Four)

Sr	List of Assignments
1	**Data Wrangling I** Perform the following operations using Python on any open-source dataset (e.g., data.csv): • Import all the required Python Libraries. • Locate an open-source dataset from the web (Kaggle/UCI Repository/Data.gov). • Provide a clear description of the dataset and its source. • Load the dataset into Pandas DataFrame. • Perform Data Preprocessing: Check for missing values, generate initial statistics, identify datatypes. • Perform Data Formatting and Data Normalization. • Convert categorical variables into quantitative variables using Label Encoding/One-Hot Encoding. • Document all preprocessing operations with observations and outputs. **Suggested Dataset: Titanic Dataset**
2	**Data Wrangling II** Perform the following operations using Python on any open-source dataset: • Scan all variables for: Missing values, Inconsistencies, Duplicate entries. • Scan numerical variables for outliers using IQR Method / Z-score Method. • Apply data transformations for: Changing scale, Converting non-linear data into linear form. • Perform: Standardization, Normalization, Log Transformation. • Compare original and transformed datasets using visualizations. • Document methodology and observations properly. **Suggested Dataset: House Prices Dataset**
3	**Basic Statistics – Measures of Central Tendencies and Variance** Perform the following operations on any open-source dataset: • Provide summary statistics: Mean, Median, Mode, Min, Max, Variance, Standard Deviation, Percentiles. • Group quantitative variables using categorical variables. • Compute statistical summaries for grouped data. • Create suitable visualizations for distribution analysis. • Write Python programs to display statistical measures. • Interpret statistical results and generate analytical observations. **Suggested Dataset: Students Performance Dataset**
4	**Data Analytics I** Create a Linear Regression Model using Python/R to predict house prices. • Import and preprocess dataset. • Identify dependent and independent variables. • Perform feature selection and correlation analysis. • Split dataset into training and testing datasets. • Train Linear Regression model and predict target values. • Evaluate model using: MAE, MSE, RMSE, R^2 Score. • Visualize regression trends and prediction results. **Suggested Dataset: Boston Housing Dataset**
5	**Text Analytics** Extract sample documents and perform document preprocessing. • Tokenization, POS Tagging, Stop-word Removal, Stemming, Lemmatization. • Create document representation using: TF, IDF, TF-IDF. • Generate: Word Frequency Charts, WordCloud, Sentiment Analysis Summary.

	• Interpret analytical results. **Suggested Dataset: Twitter Sentiment Dataset**
6	**Data Visualization I** Use the Titanic dataset and perform exploratory visualization analysis. • Analyze passenger information using Seaborn. • Plot histogram for ticket fare distribution. • Analyze passenger distribution patterns. • Identify trends and anomalies using visualizations. **Technologies: Matplotlib, Seaborn** **Suggested Dataset: Titanic Dataset**
List of Assignment - Group B [One Mini-Project (In Team of 3-4 Students and all statements should cover)]	
1	**Global Innovation Network Analysis (GINA)** Write a case study on Global Innovation Network Analysis. • Discovery business problem framing. • Data collection and preprocessing. • Model planning and analytical techniques. • Visualization and dashboard creation. • Results and key findings. • Interpretation of business insights. **Suggested Dataset: Global Innovation Index Dataset**
2	**E-Commerce Analytics and Sales Forecasting System** Develop an analytics platform for e-commerce business intelligence. • Modules: Customer segmentation, Product trend analysis, Sales forecasting. • Recommendation insights. • Interactive dashboard development. • KPI-based analytics reporting. **Technologies: Python, SQL, Power BI/Tableau** **Suggested Dataset: E-Commerce Dataset**
3	**Healthcare Analytics and Visualization System** Develop a healthcare analytics system using Python. • Modules: Patient data preprocessing, Disease trend analysis, Statistical analysis. • Predictive analytics. • Dashboard visualization. • Analytical reporting and insights generation. **Technologies: Python, Machine Learning, Power BI/Tableau** **Suggested Dataset: Healthcare Dataset**

Guidelines for Laboratory Conduction – Cryptography and Network Security

List of Assignment - Group A (Any Four)

List of Assignment	
Sr. No	
1.	Implement the Caesar Cipher and Playfair Cipher.
2.	Implement file encryption/decryption using AES/DES
3.	To verify data integrity using Hash functions and observe how a tiny change in input results in a massive change in output.
4.	To capture and compare unencrypted (Telnet) and encrypted (SSH) traffic to understand the necessity of transport layer security.
5.	Perform a TCP/UDP port scan on a local virtual machine. Identify open ports and service versions using nmap
Mini-Project (Any One) (In Team of 3-4 Students)	
1.	Authenticated File Encryption System
2.	Network Security Analysis using Encrypted Communication and Port Scanning.

Guidelines for Laboratory Conduction – Blockchain Technology

List of Laboratory Experiments/Assignments (Any 4 assignments and one Mini project is mandatory

1. Installation of Metamask and create an account and deposit some ethers from the faucets in to your account.
2. To understand the basic structure of a blockchain by simulating blocks with hashes, previous hashes, and Merkle trees using Python.
3. Explore Remix IDE. Write a smart contract to print "HELLO WORLD "in solidity on Remix IDE. Write smart contract to add two umbers taken input from the user.
4. Write a smart contract on a test network, for Bank account of a customer for following operations:
   - Deposit money
   - Withdraw Money
   - Show balance
5. Write a program in solidity to create Student data. Use the following constructs:

a) Structures
b) Arrays
c) Fallback

6. Write a survey report on types of Blockchain and its real time use cases.

Mini-Project (Any One) (In Team of 3-4 Students)

1. Create a dApp (de-centralized app) for e-voting system using Blockchain Technology.
2. Design and develop a blockchain-based system for securely storing health or academic records in an immutable and tamper-proof manner. Ensure data integrity, transparency, and controlled access using cryptographic techniques and decentralized architecture.

Guidelines for Laboratory Conduction – Image Processing

List of Assignment - Group A (Any Four)

List of Assignments

1. Image Representation, Sampling and Quantization Tools: MATLAB / Python (OpenCV + NumPy)
2. Read and display grayscale and color images, Convert RGB image to grayscale, Perform image resizing (down-sampling and up-sampling).Perform quantization
3. Implementation of 2D Convolution and Spatial Filtering
4. Implement 2D convolution manually (without built-in function). Apply averaging filter, Gaussian filter. Compare smoothing effects, Add Gaussian noise and remove using smoothing filters.
5. Intensity Transformations and Histogram Processing

Mini-Project (Any One) (In Team of 3-4 Students)

Sr. No	**List of Assignments Group-B**
1.	Develop a system that detects human faces from images or live webcam video and identifies the person using stored datasets.
2.	Design an image processing application that detects and extracts vehicle number plates from images or videos.
3.	Create a secure image encryption system that protects image data from unauthorized access.

Savitribai Phule Pune University **Third Year of Computer Science (2024 Course)**		
MDM-371-CSC - Green Computing		
Teaching scheme	**Credits**	**Examination Scheme**
Tutorial: 01 Hours/Week **Practical :** 02 Hours/Week	02	**Term Work:** 25Marks **Oral :** 25 Marks

Course Objectives: The course aims to:

1. Understand environmental impacts of computing technologies.
2. Explore sustainable hardware, software, and networking solutions.
3. Analyze energy efficiency in modern IT infrastructure.
4. Apply Green IT policies and intelligent techniques to build sustainable computing systems.

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

- **CO1: Explain** environmental impacts of computing and the need for green IT solutions.
- **CO2: Analyze** green business processes, enterprise architecture, and sustainable information systems.
- **CO3: Evaluate** energy-efficient techniques used in data centers and communication networks.
- **CO4: Apply** sustainability metrics, regulatory standards, and AI-based approaches to optimize energy consumption in computing systems.

Course Contents

Unit I - Fundamentals of Green Computing (03 Hours)

Introduction to Green Computing, Definition of Green Computing, Need for Green Computing, Objectives of Green Computing, Relationship between Business, IT and Environment, Environmental Impact of Information Technology, Energy Consumption in IT Infrastructure, Electronic Waste (E-Waste), Environmental Effects of E-Waste, Holistic Approach to Green IT, Concept of Greening IT Infrastruc-ture, Green IT Strategies for Organizations, Drivers of Green IT Adoption, Energy Efficient Computing Practices, Sustainable IT Management, Role of IT in Environmental Sustainability, Green Data Usage, Digitalization to Reduce Paper Consumption, Eco-friendly Computing Technologies, Green Procure-ment in IT, Green Supply Chain in IT, Benefits of Green Computing, Challenges in Implementing Green Computing.

Case Study: A university campus replaces hundreds of old computers every year, resulting in large amounts of electronic waste. Improper disposal may cause environmental pollution due to toxic materials. Propose a sustainable e-waste management system for the campus

Unit II - Green Assets and Modelling (03 Hours)

Green Assets: Buildings, Data Centers, Networks, and Devices – Green Business Process Management: Modeling, Optimization, and Collaboration – Green Enterprise Architecture – Environmental Intelligence – Green Supply Chains – Green Information Systems: Design and Development Models.

Case Study: Green Information Systems: Design and Development Model

Unit III Data Centers and Networking (03 Hours)

Data centre sustainability: Power Usage Effectiveness (PUE) metrics, Virtualization: Consolidating servers and storage, Energy-efficient networking and communication technologies, A detailed syllabus on Green Computing focusing on Data Centres and Networking, Energy-efficient hardware, virtualization, cooling techniques, and sustainable networking protocols., Green Facility Design: Site selection, modular data centre design, building orientation, and energy-efficient building materials. Power Efficiency: Efficient power distribution (AC vs. DC), Uninterruptible Power Supplies (UPS) in high-efficiency modes, and smart PDUs., Cooling Systems Optimization: Hot/cold aisle containment, blanking panels, liquid cooling (direct-to-chip/immersion), and free cooling (outside air economization), Renewable Energy Integration: On-site solar/wind generation, Power Purchase Agreements (PPAs), and Battery Energy Storage Systems (BESS)

Case Study: AI Infrastructure (Cisco): specialized AI-ready data center to manage intensive workloads, using Nexus Dashboard for centralized control.

Unit IV Compliance, Policies, Case Studies (03 Hours)

Regulations: WEEE Directive, RoHS, and environmental laws Metrics: Measuring, monitoring, and reporting energy consumption AI for Sustainable Computing: AI-based energy consumption monitoring, Machine learning models for predicting power usage in data centers, AI-driven cooling systems for green data centers AI-based Resource Optimization: Intelligent workload scheduling, Smart power management in cloud computing, AI-based demand prediction to reduce idle computing resources

Case Study: Green IT in telecom, banking, manufacturing AI-based green data centers, Green computing in homes and offices

Unit V - AI in Green IT (03 Hours)

AI in Environmental Monitoring: AI models to measure carbon footprint, Smart IoT sensors with AI for energy efficiency, AI-enabled smart buildings and smart campuses, **Green AI :** Energy-efficient machine learning models, Reducing computational cost of deep learning, Model compression and efficient AI algorithms

Learning Resources

Text Books:

1. Hwaiyu Geng, Data Center Handbook, 2nd Edition, John Wiley & Sons, 2014. ISBN: 978-1118436639
2. San Murugesan, G. R. Gangadharan (Editors), Harnessing Green IT: Principles and Practices, 1st Edition, John Wiley & Sons / IEEE Press, 2012. ISBN: 978-1119970057
3. Bhuvan Unhelkar, Green IT Strategies and Applications: Using Environmental Intelligence, 1st Edition, CRC Press / Taylor & Francis, 2011. ISBN: 978-1439815977

Reference Books

1. Carl Spathis, Kevin Curran, Green Computing: Tools and Techniques for Saving Energy, Money and Resources, 1st Edition, CRC Press, 2012. ISBN: 978-1439879283
2. Jason Harris, Green Computing and Green IT: Best Practices on Regulations and Industry, 1st Edition, CreateSpace Independent Publishing, 2008. ISBN: 978-1440439612
3. Bud E. Smith, Green Data Centers: Steps for the Journey, 1st Edition, Prentice Hall / IBM Press,

2009. ISBN: 978-0137043750

4. Jorge Marx Gómez, Bernhard Niemann, Richard Thomas, Green IT: Technologies and Applications, 1st Edition, Springer, 2012. ISBN: 978-3642221958

MOOC / NPTEL/YouTube Links

1. Green Computing – IIT Kharagpur https://onlinecourses.nptel.ac.in/noc20_cs79/preview
2. Cloud Computing – IIT Kharagpur https://onlinecourses.nptel.ac.in/noc23_cs18/preview
3. Internet of Things – IIT Kharagpur https://onlinecourses.nptel.ac.in/noc23_cs06/preview
4. Sustainable Engineering – IIT Madras https://onlinecourses.nptel.ac.in/noc22_ce95/preview

E-Books

1. IEEE Xplore Digital Library – Green Computing Research Papers
2. Association for Computing Machinery Digital Library – Sustainable Computing Research
3. Sustainable Computing
4. Sustainability, Green IT and Education Strategies in the Twenty-first Century

Guidelines for Instructor's Manual

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

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set (if applicable), mathematical model (if applicable), conclusion/analysis). Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal must be avoided. Use of DVD containing student's programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

Guidelines for Laboratory /Term Work Assessment

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

Guidelines for Practical Examination

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

Guidelines for Laboratory Conduction

List of Assignment- Group-A (any 5, Part B any 2 and 1 mini project)	
1	**Study of Energy Consumption of Computing Devices** Different computing devices consume different amounts of electrical energy depending on their hardware configuration and workload. Analyze the energy consumption of devices such as desktop computers and laptops under different conditions including idle state, document editing, internet browsing, and high CPU utilization. Record and compare the power usage and estimate the annual electricity consumption. Based on the analysis, recommend suitable green computing practices for reducing energy consumption in educational institutions or offices.
2	**Survey and Classification of Electronic Waste (E-Waste)** Rapid technological advancements lead to frequent replacement of electronic devices resulting in significant electronic waste. Conduct a survey of electronic devices available in a computer laboratory or department. Identify obsolete or unused equipment and classify them into reusable, recyclable, and hazardous categories. Analyze environmental risks associated with improper disposal of e-waste and propose an eco-friendly e-waste management strategy for the institution.
3	**Analysis and Optimization of Green Business Processes** Many organizational processes involve excessive paper usage, redundant steps, and inefficient resource utilization. Select a business process such as student admission, library management, or billing system. Study the existing workflow and identify areas where resources are wasted. Redesign the workflow using green business process management principles to minimize paper usage, reduce energy consumption, and improve efficiency. Represent the optimized workflow using a process modeling diagram.
4	**Design of Green Enterprise Architecture for IT Infrastructure** Organizations rely heavily on IT infrastructure consisting of servers, storage systems, and networking equipment that consume significant energy. Study a typical enterprise IT infrastructure and identify components that contribute to high energy consumption. Propose a green enterprise architecture by incorporating technologies such as virtualization, cloud computing, and energy-efficient networking devices. Illustrate the improved architecture using a suitable system diagram and explain how it reduces environmental impact.
5	**Data Center Energy Efficiency Analysis using PUE** Data centers require large amounts of electricity for servers, cooling systems, and networking devices. Study the concept of Power Usage Effectiveness (PUE) as a metric for measuring energy efficiency. Analyze a hypothetical or sample data center configuration by considering power consumed by IT equipment and supporting infrastructure. Calculate the PUE value and evaluate whether the data center is energy efficient. Suggest possible improvements such as advanced cooling techniques or renewable energy integration.
6	**Implementation of Virtualization for Energy-Efficient Computing** Running multiple servers on separate physical machines results in inefficient resource utilization and increased energy consumption. Implement virtualization by installing multiple virtual machines on a single host system. Monitor CPU, memory, and resource usage before and after virtualization. Analyze how server consolidation reduces hardware requirements and contributes to energy-efficient computing.
7	**Machine Learning-Based Prediction of Energy Consumption** Energy consumption in computing systems often varies depending on workload and system activity. Using a dataset containing historical energy consumption values, implement a machine learning model to predict future energy usage. Train and test the model using suitable regression techniques and evaluate prediction accuracy. Analyze how predictive models can help organizations optimize energy usage and reduce operational costs.
8	**Development of Energy Monitoring and Visualization Dashboard** Monitoring energy consumption helps identify inefficient systems and reduce unnecessary power usage. Develop a dashboard that reads energy consumption data from a dataset and visualizes patterns using charts and graphs. Identify peak consumption periods and propose recommendations to impro9v3e energy efficiency in computing infrastructure. https://archive.ics.uci.edu/dataset/235/individual+household+electric+power+consumpt

Tools Used for Group-B (Any Two Assignments)

- Internet Research Databases
- IEEE / ACM Research Papers
- Microsoft Excel or Google Sheets
- Word Processing and Presentation Tools
- Python (optional for analysis)

1.	Computing devices contribute significantly to energy consumption and carbon emissions. Prepare a report analyzing the carbon footprint generated by computing devices used in educational institutions or offices. Study existing sustainable practices such as power management, device recycling, and reduced paper usage. Suggest practical steps for promoting green computing awareness within the organization.
2.	Technology companies are increasingly adopting sustainable supply chain practices to reduce environmental impact. Select a well-known IT company and study its supply chain management strategies. Analyze how eco-friendly packaging, efficient logistics, and responsible sourcing contribute to sustainability. Present your findings in the form of a structured analysis.
3.	Traditional data centers often suffer from low resource utilization and high operational costs. Study the concept of cloud computing and analyze how cloud infrastructure improves resource utilization and energy efficiency. Compare traditional server infrastructure with cloud-based solutions and prepare a comparative analysis highlighting environmental benefits.
4.	Cooling systems consume a large portion of energy in data centers. Research AI-based cooling technologies used in modern data centers. Analyze how predictive algorithms monitor temperature patterns and automatically adjust cooling levels to maintain optimal conditions while minimizing energy consumption.

Mini-Project Group-C (Any one -In Team of 3-4 Students)

Tools Used for Mini Projects

- Python Programming Language
- Jupyter Notebook
- Streamlit / Power BI for dashboards
- VirtualBox / VMware
- Excel / MATLAB
- IoT Simulation Tools (optional)

1.	Educational institutions operate numerous computing devices such as desktops, servers, printers, and networking equipment. Conduct an energy audit of computer laboratories by collecting data on device usage patterns and electricity consumption. Analyze inefficiencies and recommend strategies such as power management policies, device upgrades, and reduced paper usage to promote green computing.
2.	Design a complete green IT infrastructure for a medium-sized organization. The proposed architecture should incorporate virtualization, cloud computing, and energy-efficient networking equipment. The design should aim to reduce power consumption, improve resource utilization, and support sustainable IT operations.
3.	Develop a simulation model to study how workload distribution and server consolidation affect energy consumption in a data center. The simulation should demonstrate how virtualization and efficient resource allocation reduce overall energy usage.
4.	Develop a software system that collects energy usage data from computing devices and predicts future consumption patterns using machine learning algorithms. The system should visualize energy usage trends and provide recommendations for optimizing energy efficiency.

Savitribai Phule Pune University **Third Year of Computer Science (2024 Course)** **VSE-372-CSC: Full Stack Web Development**		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical: 02 Hours/Week	01	**Term Work:** 50 Marks

Prerequisite Courses: if any: Object Oriented Programming in Java(VSE-281-CSC)

Course Objectives: The course aims to focus learning with following objectives -

1. Understand the concepts of modern Full Stack Development using front-end, back-end, database, and AI-assisted development tools.
2. Develop dynamic and responsive web applications using JavaScript, React, Express, and PostgreSQL.
3. Design and integrate server-side applications, REST APIs, and databases for real-world applications.
4. Apply AI tools and modern software development practices towards the deployment of Full Stack applications.

Course Outcomes:

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

- **CO1: Apply** JavaScript for interactive web applications.
- **CO2: Develop** responsive applications with AI integration using React components advanced React concepts.
- **CO3: Design** server-side applications using Express framework and integrate PostgreSQL databases for data storage, retrieval, and RESTful API development.
- **CO4: Explore** and **Evaluate** AI-assisted software development tools for code generation, debugging, testing, optimization, and deployment of Full Stack applications.

Course Contents

Unit I JavaScript (08 Hours)

Core JavaScript: Variables, Data types, Functions, Arrays & methods (map, filter, reduce), Objects De-structuring, Spread operator, Optional chaining, ES6 modules **Advanced JavaScript:** Asynchronous JS, Callbacks, Promises, async/await, Event loop, Closures, Hoisting, this keyword, Debouncing & throttli

Hands On: Build Todo App (Vanilla JS), Array manipulation exercises, Form validation logic Fetch API Project, Build search with denounce, Call REST API

Unit II React (08 Hours)

React Fundamentals: Why React? SPA concept, Components, JSXP, Props, useState, Conditional rendering, Lists & keys, **React Advanced:** useEffect, useRef, useContext, Custom hooks, React Router, API integration, Error boundaries, Performance optimization basics, **AI Integration in React:** Calling OpenAI API, Creating AI Chatbot, AI content generation feature, Prompt engineering basics

Hands On: Build Todo App in React, Counter App, Form handling, Build Dashboard UI, Fetch backend data, Protected routes

Unit III PostgreSQL and Express (08 Hours)

What is RDBMS? SQL basics, CRUD queries, JOIN, Indexing basics, Normalization, Transactions, Basic query optimization Backend with Express Node.js basics, REST API design, Express routing, Middleware, JWT authentication, Password hashing (bcrypt), Error handling, MVC structure, Environment variables

Hands On: Create student table, Write CRUD queries, Connect PostgreSQL to backend
Build REST API, Connect PostgreSQL, Implement authentication, Secure routes

Unit IV	AI for Software Development	(06 Hours)

AI Coding Assistants: GitHub Copilot, Cursor, ChatGPT, Claude **Prompt Engineering for Developers:** Writing structured prompts, Debugging with AI, Refactoring with AI, Generating test cases.

AI System Design Basics: Where AI fits in architecture, AI API integration pattern, RAG basics (high-level)

Hands On: Generate API using Copilot, refactor legacy code using AI, Add AI feature to project
Generate interview answers using AI

LAB WORK

1) Mini Project: Smart Notes App

Features:

- User Authentication
- Create / Edit Notes
- PostgreSQL storage
- AI summary generator
- AI title suggestion

Tech Stack:
React + Express + PostgreSQL + OpenAI API

2) Capstone Project: AI-Powered Student Career Assistant

Features:

- User registration
- Resume upload
- AI resume analysis
- AI career suggestion
- Chatbot guidance
- Admin dashboard
- Deployed live

Architecture:
Frontend → Backend → PostgreSQL → AI API
Students must:

- Draw architecture diagram
- Explain authentication
- Explain DB relationships
- Explain AI integration
- Explain scaling approach

Learning Resources

Text Books:

1. "Eloquent JavaScript" – Marijn Haverbeke
2. "Node.js Design Patterns" – Mario Casciaro
3. "PostgreSQL: Up and Running" – Regina Obe & Leo Hsu
4. "Designing Data-Intensive Applications" – Martin Kleppmann

Reference Books:

1. "You Don't Know JS (Series)" – Kyle Simpson
2. "Full-Stack React Projects" – Shama Hoque
3. Artificial Intelligence Basics" – Tom Taulli
4. "Building AI Applications with OpenAI APIs" (Latest Editions / Guides)

E-Books:

1. MDN Web Docs https://developer.mozilla.org (Best free resource for HTML, CSS, JS)
2. React Official Docs https://react.dev (Modern hooks-based learning)
3. Node.js Documentation https://nodejs.org/en/docs
4. Express.js Docs https://expressjs.com
5. PostgreSQL Official Docs https://www.postgresql.org/docs
6. OpenAI API Documentation https://platform.openai.com/docs
7. Google Gemini API Docs https://ai.google.dev

Savitribai Phule Pune University **Third Year of Computer Science (2024 Course)**		
ELC-381-CSC: Internship/ On Job Training		
Teaching scheme	**Credits**	**Examination Scheme**
Practical : 08 Hours/Week	04	**Term Work:** 25 Marks **Oral**: 25 Marks

Course Objectives: The course aims to:

1. To expose students to real-world industry practices.
2. To bridge the gap between academic learning and practical implementation.
3. To develop professional, technical, and communication skills.
4. To encourage self-learning and problem-solving abilities.
5. To provide hands-on experience in emerging technologies

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

- **CO1:** Apply theoretical knowledge to solve real-world problems.
- **CO2:** Demonstrate technical competency in tools/technologies used in industry.
- **CO3:** Exhibit professional ethics and teamwork.
- **CO4:** Communicate effectively through reports and presentations.
- **CO5:** Analyze and document industrial workflows and processes.

Guidelines

1. Students should opt for a internship/JOT that would provide them to gain ample field knowledge in the relevant field of engineering such that theoretical knowledge gained in the class can be applied to solve the practical/ field problem.
2. Students must have to opt for technical internship after VI semester and before VII semester, preferably during summer break.
3. **Undergoing a training programme/ Course at a particular organization for specified dura-tion is NOT considered as summer internship**
4. However student can attend such programs mentioned in above to learn new tools for short duration that would help for solving the problem undertaken in the internship
5. Students should take a challenging task, may be a small portion, and apply the knowledge gained to solve it.
6. Internship can also involve data collection from different sources, including generating experimental data, collection of data from field etc. The data may be analyzed later on.
7. Different central and state government organizations, CSIR labs, premier institutions like IITs and IIMs, DRDO,public sector undertaking organizations, top IT companies may be considered for internships.
8. Student need to submit Synopsis, Permission letter and offer letter to Internship coordinator before proceeding to internship.

9. Internship completion will be considered only after submission of valid documents at the end of internship like Completion certificate, Report and presentation of work done, feedback from industry etc.
10. Student will appear for term work evaluation where he/she will present the work done before mentor(s) at the end of internship.

Suggested Internship Activities

- Students are expected to perform the following activities during internship:
- Phase I – Orientation and Requirement Study
  - Understanding organization structure
  - Study of workflow and operational processes
  - Requirement analysis and project allocation
  - Understanding tools and technologies used
- Phase II – Technical Learning and Development
  - Coding and implementation
  - Database design and integration
  - Software testing and debugging
  - API integration and deployment
  - Use of version control systems
  - Documentation practices
- Phase III – Project Execution
  - Module development
  - Testing and validation
  - Performance optimization
  - Client interaction (if applicable)
  - Team collaboration
- Phase IV – Documentation and Presentation
  - Preparation of internship report
  - Preparation of project demonstration
  - Final presentation and viva voce

Deliverables

- Internship Joining Report
- Weekly Logbook
- Mid-term Progress Report
- Supervisor Feedback (Initial)

Internship Structure

The internship may be carried out in any one of the following domains:

- Software Development
- Artificial Intelligence and Machine Learning
- Data Science and Analytics
- Cloud Computing and DevOps
- Cyber Security
- Web and Mobile Application Development
- IoT and Embedded Systems
- Networking and System Administration
- Automation and Robotics Software
- Research and Development
- Entrepreneurship and Startup Projects
- Government/NGO Technical Projects

Nature of Internship

Students shall undergo internship/training in one of the following:

- Registered companies / startups
- Government organizations
- Research institutions
- Recognized industry-academic collaborative projects
- Internships may be conducted in offline, online, or hybrid mode, subject to proper approval and verification.

Guidelines for Internship Report Writing

1. Preliminary Pages
    - Cover Page
    - Certificate from Organization
    - Certificate from Department
    - Acknowledgement
    - Abstract
    - Table of Contents
2. Chapter 1 – Organization Profile
    - Company overview

- Vision and mission
- Products/services
- Organizational structure

3. Chapter 2 – Problem Statement and Objectives
    - Project title
    - Need of project
    - Objectives
    - Scope
4. Chapter 3 – Technologies and Methodology
    - Software/hardware tools used
    - Development methodology
    - System architecture
    - Database design
5. Chapter 4 – Work Carried Out
    - Tasks completed
    - Screenshots/results
    - Challenges faced
    - Solutions implemented
6. Chapter 5 – Learning Outcomes
    - Technical learning
    - Professional skills acquired
    - Industry exposure
    - Future scope
7. Chapter 6 – Conclusion
    - Summary of work
    - Achievements
    - Suggestions

References : IEEE format references preferred

Appendices

- Source code snippets
- Certificates
- Additional screenshots

Learning Resources

Text Books:

1. W. J. King and James G. Skakoon , The Unwritten Laws of Engineering , ASME Press
2. Stuart Walesh, Engineering Your Future: The Professional Practice of Engineering
3. Eliyahu M. Goldratt, The Goal: A Process of Ongoing Improvement
4. AICTE Internship policy : AICTE Internship Policy: Guidelines & Procedures
5. AICTE Internship Portal : https://internship.aicte-india.org

Savitribai Phule Pune University, Pune

Maharashtra, India

Task Force for Curriculum Design and Development

Program Coordinator

Dr. Deepali M. Ujalambkar - Member, Board of Studies - Computer Engineering

Compiled by

Dr. Sandip A. Kahate, Dr. D. Y. Patil Technical Campus, Varale-Talegaon, Pune

Team Members for Course TE Design

Artificial Intelligence	
Dr. Neeta Deshpande	RH Sapat College of Engg, Management Studies and Research
Dr. Vaishali Tidke	MVPS's KBTCOE Nashik
Dr. Chandrakant D. Kokane	Vishwakarma Institute of Technology, Pune
Dr. Vaibhav Dabhade	MET's Institute of Engineering
Dr. Suvarna Patil	D Y Patil International University, Akurdi, Pune
Mrs. Megha Amey Patil	RH Sapat College of Engg, Management Studies and Research

Computer Network	
Dr. S.R. Lahane	RH Sapat College of Engg, Management Studies and Research
Dr. Ranjana Dahake	MET's Institute of Engineering, Nasik
Dr. Salunke Mangesh	Marathwada Mitra Mandal's Institute of Technology
Dr. Megha Kadam	Marathwada Mitra Mandal's Institute of Technology
Mrs. Pallavi Rahinj	Rajiv Gandhi College of Engineering,Karjule Harya
Mr. Prasad Bhosle	Trinity Academy of Engineering

Theory of Computation	
Dr.Prajwal S.Gaikwad	AISSMS's Institute of Information Technology Pune
Dr.Pradnya Kulkarni	Dr.Vishwanath Karad MIT World Peace Univaersity, Pune
Mr.Shreyas N. Hogade	Software developer at NICE Systems Pune
Mrs Jayaprabha Kanase	PES Modern College of Engineering, Pune
Mrs Nilam Patil	D.Y.Patil College of Engineering Akurdi, Pune

Information Retrieval	
Dr. Balaji Kisanrao Bodkhe	MES Wadia College of Engineering Pune
Dr. Vinod Wadne	JSPM's Bhivarabai Sawant Institute of Technology and Research
Rekha Kotwal	JSPM's Bhivarabai Sawant Institute of Technology and Research
Dr. Archana Kale	MES Wadia College of Engineering Pune
Dr. Suvarna Sanjay Pawar	Trinity Academy of Engineering Pune
Dr. Pranali Dahiwal	MIT World Peace University Kothrud Pune

Cloud Computing	
Dr Megha V. Kadam	Marathwada Mitra Mandal's Institute of Technology
Dr. Mangesh D. Salunke	Marathwada Mitra Mandal's Institute of Technology
Dr. Pramod B. Dhamdhere	Marathwada Mitra Mandal's Institute of Technology
Dr. Anand Khatri	Jaihind College of Engineering
Dr. Ashwini B. Gavli	S. B. Patil College of Engineering

UI/UX	
Prof. Pallavi M. Shimpi	Department of Computer Engg, ADYPSoE, Pune
Dr.Deepali Baviskar	MITWPU, Pune
Mr. Chaitnya Bhosale	MAE Alandi,Pune
Mr. Yogesh Mali	ADYPU,PUNE
Mr.Abhishek Bangar	Jaihind College of Engineering, Tal Junnar Dist Pune

Embedded systems	
Dr. Archana Banait	MET's Institute of Engineering,Nasik
Dr. Prashant Salunkhe	GOGATE Electosystem Pvt.Ltd,Nasik
Dr Sujeet More	Trinity college of engineering & research ,Pune
Dr. Supriya sable	MIT ,Academy of Engineering Alandi,Pune
Mrs Amrita Pundlik	GES's R H Sapat COEMSR , Nashik

Machine Learning	
Dr. Archana Vaidya ,	R.H.Sapat College of Engineering,Nashik
Prof Said S.K.	Jaihind College of Engineering, Tal Junnar Dist Pune
Dr. Sandip A. Kahate	Dr. D. Y. Patil Technical Campus, Varale, Talegaon, Pune

Software Engineering and Modelling	
Dr. Uday Patkar	Bharati Vidyapeeth's College of Engineering Lavale Pune.
Dr. Balasaheb S. Tarle	MVPS's KBT College of Engineering,Nasik
Dr. Rahul Gaikwad	Amrutvahini College of Engineering, Sangamner
Mr.Sagar Ashok Dhanake	Dr. D Y Patil College of Engineering and Innovation, Varale, Talegaon

Soft Computing and Fuzzy Logic	
Ms. Priyanka Waghmare	Artificial Intelligence and Data Science Department, ADYPSOE, Pune
Dr. Archana Kale	MESCOE, Pune
Ms. Sweta Wankhade	ADYPSOE, Pune
Ms. Supriya Survase	ADYPSOE, Pune

AR/VR	
Dr N F Shaikh	MES College of Engineering, Pune
Dr. Shabnam Sayyad	AISSMS College of Engineering, Pune
Mr. Sagar Shinde	MES College of Engineering, Pune
Mr. Yogesh Survase	FirebirdVR Pvt. Ltd
Mr. Akshay Rathod	FirebirdVR Pvt. Ltd

Natural Language Processing	
Prof. Gopika Fattepurkar	Ajeenkya DY Patil School of Engineering, Lohgaon
Prof. Rupali N. Wagh	Ajeenkya D.Y. Patil School of Engineering, Lohgaon
Prof. Pranjali Deshpande	MKSSS's Cummins College of Engineering for woman, Pune
Mr. Vishal Bandgar	Nihilant Pvt. Ltd., Pune
Prof. Jayashree Pansare	MES College of Engineering, Pune

Quantum Computing	
Ms. Vrushali Nikam	R.H.Sapat College of Engineering,Nashik
Mr. Nilesh Pinjarkar	JSPM's Rajarshi Shahu College of Engineering
Mrs. Kumkum Bala	Bharati Vidyapeeth's College of Engineering Lavale Pune
Mrs Parinita Chate	Bharati Vidyapeeth's College of Engineering Pune

Data Analytics and Visualization	
Dr. Khatri Anand A.	Jaihind College Of Engineering, Kuran
Prof. Said Shubhangi K.	Jaihind College Of Engineering, Kuran
Dr. Jachak Sweety	Guru Gobind Singh College of Engineering & Research Centre, Nashik
Prof. Bhosale Sachin B.	Jaihind College Of Engineering, Kuran
Prof.Gholap Pooja	Sharadchandra Pawar College of Engineering, Otur
Prof. Nangare Vaibhav	Sahyadri Valley College of Engineering & Technology, Rajuri

Cryptography and Network Security	
Dr. Mangesh D. Salunke	Marathwada Mitra Mandal's Institute of Technology
Dr Megha V Kadam	Marathwada Mitra Mandal's Institute of Technology
Mrs. Pallavi Rahinj	Rajiv Gandhi College of Engineering,Karjule Harya
Dr. Ashwini B. Gavli	S. B. Patil College of Engineering
Dr. Pramod B. Dhamdhere	Marathwada Mitra Mandal's Institute of Technology

Bockchain Technology	
Dr. Vandana Navale	Ajeenkya DY Patil School of Engineering,Lohegaon,Pune
Dr.Divya Sharma	Ajeenkya DY Patil School of Engineering,Lohegaon,Pune
Dr. Priyanka Kedar	Entrata India Pvt. Ltd.,Koregaon Park Pune.
Dr.Varsha Dange	Vishwakarma Institute of Technology, Bibwewadi Pune
Mr.Gajanan Vinayak Malvade	Millennium Management Corporation, Dublin

Image Processing	
Dr. Shabnam Sayyad	AISSMSCOE, Pune
Dr Nidhi Mishra	Electrical Engineering Department, JSPM,Hadapsar Pune
Dr. Shilpa Borde	Computer Engineering Department,JSPM,Hadapsar Pune.
Dr.Rahshri Itkarkar	E&TC Department, AISSMSCOE,Pune
Mr.Shubham Jain	Software Engineer at Microsoft Pune

Green Computing	
Dr. Archana Suhas Vaidya	GES's R H Sapat COEMSR , Nashik
Dr. D. P. Gaikwad	AISSMCOE, Pune
Dr. Brijendra Gupta	Siddhant COE, Pune
Dr. Yogita D. Sinker	VP's KB IoT, Baramati
Dr. Renuka Kajale	PCET's Nutan Maharashtra Institute of Engineering and Technology

Chairman

Dr. Nilesh Uke - Board of Studies Computer Engineering

Savitribai Phule Pune University, Pune

Dean

Dr. Raosaheb Latpate - Dean – Science and Technology

Savitribai Phule Pune University, Pune

***