॥ योः क्रियावानू स पण्डितः ॥

SAVITRIBAI PHULE PUNE UNIVERSITY

(सावित्रीबाई फुले पुणे विद्यापीठ)

Formerly University of Pune

Syllabus Booklet

For

T.E.

COMPUTER SCIENCE AND DESIGN (CSD)

2024 PATTERN

With effect from Academic Year 2026-27

Contents

Nomenclature

AICTE All India Council for Technical Education

CCE Comprehensive Continuous Evaluation

CO Course Outcomes

ELC Experiential Learning Course

ESE End-Semester Examination

GAPC Graduate Attributes and Professional Competencies

MDM Multidisciplinary Minor

NEP National Education Policy

OE Open Elective

OJT On Job Training

PCC Programme Core Course

PEC Programme Elective Course

PEO Programme Educational Objectives

PO Programme Outcomes

PSO Program Specific Outcomes

QPD Question Paper Delivery

UGC University Grants Commission

VSE Vocational and Skills Enhancement Course

WK Knowledge and Attitude Profile

Preface by Board of Studies - Computer Engineering

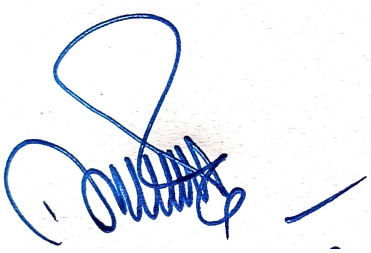
Dear Students and Teachers,

We, the members of Board of Studies Computer Engineering, are very happy to present Third Year Computer Science and Design Engineering syllabus effective from the Academic Year 2026-27.

Computer Engineering have emerged as transformative forces reshaping industries, driving innovation, and impacting our daily lives. Recognizing the growing importance and pervasive nature of these fields, we have designed this comprehensive syllabus to equip students with the foundational knowledge and practical skills. This curriculum is meticulously crafted to provide a holistic learning experience, blending theoretical concepts with hands-on applications. The revised syllabus falls in line with the objectives of NEP-2020, Savitribai Phule Pune University, AICTE, UGC, and various accreditation agencies by keeping an eye on the technological developments, innovations, and industry requirements.

Learners are now getting sufficient time for self learning either through online courses or additional projects for enhancing their knowledge and skill sets. We believe that this well-structured and comprehensive syllabus will serve as a robust foundation for aspiring Computer Engineering and AI professionals, enabling them to contribute significantly to the technological progress and address the challenges of the 21st century.

We would like to place on record our gratefulness to the faculty, students, industry experts and stakeholders for having helped us in the formulation of this syllabus.



Dr. Nilesh Uke
Chairman - Board of Studies (Computer Engineering)
Savitribai Phule Pune University

Members of Board of Studies - Computer Engineering	
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

Program Specific Outcomes (PSO)

- **PSO1:** Demonstrate proficiency in essential concepts of computer science and data science and programming solutions.
- **PSO2:** Formulate robust software design, execution, and testing strategies employing a software paradigms and Artificial Intelligence knowledge to solve real world problems.
- **PSO3:** Apply the techniques of AI and Data Science for forecasting future events in the domain of Healthcare, Education, and Agriculture, Automation , Transport etc

Programme Educational Objectives (PEO)

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	To produce graduates equipped with cutting-edge skills in Computer Engineering, Artificial Intelligence (AI) and Data Science (DS), with expertise in domains such as Machine Learning (ML), Natural Language Processing (NLP), Generative AI, enabling them to collaborate effectively in interdisciplinary teams to solve real-world industrial and societal challenges.
PEO2	Problem solving skills and Ethics	To empower graduates to think critically, apply mathematical, computational, and ethical frameworks, and design scalable, secure, and fair AI-driven systems
PEO3	Professionalism and Lifelong Learning	To inculcate the ability to adapt to changing technology through continuous learning and contribute to research, innovation, and entrepreneurship in AI and Data Science.

Knowledge and Attitude Profile (WK)

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

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

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

Programme 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 and Design Engineering, 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.

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 (ESE).

Comprehensive Continuous Evaluation (CCE) :

1. CCE of 30 marks based on all the Units of course syllabus to be scheduled and conducted at institute level.
2. Case studies included under each unit are intended to support applied learning and are part of Comprehensive Continuous Evaluation
3. 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.
4. 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

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

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

- **Unit Test**
  - **Format :** Questions designed as per Bloom's Taxonomy guidelines to assess various cognitive levels (Remember, Understand, Apply, Analyze, Evaluate, Create).
  - **Implementation**: Schedule the test after completing Units 1 and 2. Ensure the question paper is balanced and covers key concepts and applications.
- **Sample Question Distribution**
  - Remembering (2 Marks): Define key terms related to [Topic from Units 1 and 2].
  - Understanding (2 Marks): Explain the principle of [Concept] in [Context].
  - Applying (2 Marks): Demonstrate how [Concept] can be used in [Scenario].
  - Analyzing (3 Marks): Compare & contrast [Two related concepts] from Units 1 and 2.
  - Evaluating (3 Marks): Evaluate the effectiveness of [Theory/Model] in [Situation].
- **Assignments / Case Study** : Students should submit one assignment or one Case Study Report based on Unit 3 and one assignment or one Case Study Report based on Unit 4.
  - **Format:** Problem-solving tasks, theoretical questions, practical exercises, or case studies that require in-depth analysis and application of concepts.
  - **Implementation:** Distribute the assignments or case study after covering Units 3 and 4. Provide clear guidelines and a rubric for evaluation.
- **Seminar Presentation:**
  - **Format:** Oral presentation on a topic from Unit 5, followed by a Q&A session.
  - **Deliverables:** Presentation slides, a summary report in 2 to 3 pages, and performance during the presentation.
  - **Implementation:** Schedule the seminar presentations towards the end of the course. Provide students with ample time to prepare and offer guidance on presentation skills.
- **Open Book Test:**
  - **Format:** Analytical and application-based questions to assess depth of understanding.
  - **Implementation:** Schedule the open book test towards the end of the course, ensuring it covers critical aspects of Unit 5.
- **Quiz :**
  - **Format:** Quizzes can help your students practice existing knowledge while stimulating interest in learning about new topic in that course. You can set your quizzes to be completed individually or in small groups.
  - **Implementation:** Online tools and software can be used create quiz. Each quiz is made up of a variety of question types including multiple choice, missing words, true or false etc
- **Example Timeline for conducting CCE:**
  - Weeks 1-4 : Cover Units 1 and 2
  - Week 5 : Conduct Unit Test (12 marks)
  - Weeks 6-8 : Cover Units 3 and 4

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

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

End-Semester Examination (ESE)

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

- **Format and Implementation :**
  - **Question Paper Design** : Below structure is to be followed to design an End-Semester Examination (ESE) for a theory subject of 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: Use of information in new situations.
    * Analyzing: Drawing connections among ideas.
    * Evaluating: Justifying a decision or course of action.
    * Creating: Producing new or original work (if applicable).
  - **Detailed Scheme for 70 Marks :** Unit-Wise Allocation (14 Marks per Unit): Each unit will have a combination of questions designed to assess different cognitive levels. By following this scheme, you can ensure a comprehensive and fair assessment of students' understanding and application of the course material, adhering to Bloom's Taxonomy guidelines for cognitive skills evaluation.
  - **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.

***

Curriculum Structure - Semester V

Third Year Engineering (2024 Pattern) – Computer Science and Design Engineering

Course Code	Course Name	Course Type	Teaching Scheme			Examination Scheme						Credits			
			Theory	Tut	Practical	CCE	ESE	Term Work	Practical	Oral	Total	Theory	Tut	Practical	Total
PCC301CSD	Artificial Intelligence	PCC	3	-	-	30	70	-	-	-	100	3	-	-	3
PCC302CSD	Web Technology	PCC	3	-	-	30	70	-	-	-	100	3	-	-	3
PCC303CSD	Computer Network	PCC	3	-	-	30	70	-	-	-	100	3	-	-	3
PCC304CSD	Artificial Intelligence Lab	PCC		-	2	-	-	50	25	-	75	-	-	1	1
PCC305CSD	Web Technology Lab	PCC		-	4	-	-	25	25	-	50	-	-	2	2
PEC321CSD	Elective - I	PEC	3	-	-	30	70	-	-	-	100	3	-	-	3
PEC322CSD	Elective - I Lab	PEC		-	2	-	-	50	-	-	50	-	-	1	1
MDM331CSD	Internet of Things	MDM		1	4	-	-	50		-	50	-	1	2	3
	Open Elective*	OE	2	-	-	15	35		-	-	50	2	-	-	2
ELC342CSD	Technical Seminar	ELC		-	2	-	-	-	-	25	25	-	-	1	1
Total			**14**	1	**14**	**135**	**315**	**175**	50	25	700	**14**	**1**	**7**	**22**

***Note:** Students can opt for Open Electives offered by different discipline/faculty like Arts, Science, Commerce and Management, Humanities or Inter-Disciplinary studies. Example - IPR and Cyber Laws, Sustainability Development, Digital Personal Data Protection, The Constitution of India

Programme Elective - I	
PEC321ACSD	Computer Vision
PEC321BCSD	Multimedia Techniques
PEC321CCSD	Optimization Algorithms
PEC321DCSD	Embedded systems

Curriculum Structure - Semester - VI

Third Year Engineering (2024 Pattern) – Computer Science and Design Engineering

Course Code	Course Name	Course Type	Teaching Scheme			Examination Scheme						Credits			
			Theory	Tut	Practical	CCE	ESE	Term Work	Practical	Oral	Total	Theory	Tut	Practical	Total
PCC351CSD	Augmented and Virtual Reality	PCC	3	-	-	30	70	-	-	-	100	3	-	-	3
PCC352CSD	UI/UX Design	PCC	2	-	-	30	70	-	-	-	100	2	-	-	2
PCC353CSD	AR/VR Lab	PCC	-	-	4	-	-	25	25	-	50	-	-	2	2
PCC354CSD	UI/UX Design Lab	PCC	-	-	2	-	-	25	-	25	50	-	-	1	1
PEC361CSD	Elective II	PEC	3	-	-	30	70	-	-	-	100	3	-	-	3
PEC362CSD	Elective III	PEC	3	-	-	30	70	-	-	-	100	3	-	-	3
PEC363CSD	Elective III Lab	PEC		-	2	-	-	25	25	-	50	-	-	1	1
MDM371CSD	Robotics and Automation	MDM		1	2	-	-	50	-	-	50	-	1	1	2
VSE372CSD	Solar Technology and Maintenance	VSE	-	-	2	-	-	50	-	-	50	-	-	1	1
ELC381CSD	Internship/OJT	ELC	-	-	8	-	-	-	-	50	50	-	-	4	4
Total			**11**	**1**	**20**	**120**	**280**	**175**	50	75	**700**	**11**	**1**	**10**	**22**

Programme Elective - II	
PEC361ACSD	Machine Learning
PEC361BCSD	Cloud Computing
PEC361CCSD	Cyber Security
PEC361DCSD	Industrial IOT

Programme Elective - III	
PEC362ACSD	Natural Language Processing
PEC362BCSD	High Performance Computing
PEC362CCSD	Business Intelligence
PEC362DCSD	Generative AI

Savitribai Phule Pune University, Pune

Maharashtra, India

TE - Computer Science and Design

Semester - V

Savitribai Phule Pune University **Third Year - Computer Science and Design (2024 Pattern)**		
PCC301CSD- Artificial Intelligence		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses : Data Structures

Companion Course: Artificial Intelligence Lab

Course Objectives:

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

Course Contents

Unit I - Introduction to AI and Intelligent Agents (09 hours)

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

Unit–II: Problem Solving : State Space Approach and Search Strategies (09 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

Unit–III: Adversarial Search and Game theory (09 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

Unit-IV : Knowledge representation using Logical Formalisms, Propositional and First order Predicate Calculus (09 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

Unit V : Planning and Industrial Applications of AI (09 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 Applications 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)

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-Intelligence

3. https://www.kdnuggets.com/10-free-artificial-intelligence-books-for-2025

MOOC :

- Artificial Intelligence: Knowledge Representation And Reasoning By Prof. Deepak Khemani, IIT Madras https://onlinecourses.nptel.ac.in/noc26_cs63/preview

- An Introduction to Artificial Intelligence By Prof. Mausam, IIT Delhi https://nptel.ac.in/courses/106102

Savitribai Phule Pune University **Third Year - Computer Science and Design (2024 Pattern)**		
PCC302CSD-Web Technology		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses :

Companion Course: Web Technology Lab

Course Objectives:

1. To understand the fundamentals of web technologies and Internet architecture.
2. To develop interactive web pages using modern client-side technologies.
3. To understand server-side programming and web frameworks.
4. To develop dynamic web applications with database connectivity.
5. To learn modern full-stack development and web deployment concepts.

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

- CO1: Design and implement web pages using HTML5 and CSS3.
- CO2: Develop dynamic web applications using JavaScript and modern client-side technologies.
- CO3: Develop server-side applications using modern backend frameworks.
- CO4: Design database-driven web applications using SQL and NoSQL databases.
- CO5: Integrate frontend and backend technologies to develop full-stack applications.

Course Contents

Unit I - : Web Fundamentals and Front-End Basics (09 Hours)

Introduction to Internet and World Wide Web, Web architecture, Web browsers and Web servers, HTTP/HTTPS protocol, client-server communication model. HTML5: Introduction, structure of HTML document, semantic elements, headings, paragraphs, lists, links, tables, images, forms, multimedia elements. CSS3: Introduction to CSS, CSS syntax, selectors, box model, layout techniques, Flexbox, Grid layout, responsive web design. Introduction to UI frameworks: Bootstrap / Tailwind CSS.

Case Study: Design a responsive web page for a blogging or business website using HTML5 and CSS3.

Unit–II: Advanced Front end Technologies (9 Hours)

JavaScript fundamentals: Variables, data types, operators, control structures, functions, arrays, objects. DOM (Document Object Model): DOM structure, element selection, DOM manipulation, event handling. Client-side validation and form handling. AJAX and Fetch API: asynchronous communication with server. JSON data format. Overview of modern frontend frameworks:

Introduction to React.js / Angular / Vue.js concepts

Case Study: Enhancement of blogging application using JavaScript features such as:

- Dynamic content update
- Form validation
- API data fetching

Unit–III: Server Side Development and REST APIs (9 Hours)

Introduction to server-side programming. Node.js environment and architecture. Express.js framework: Routing . Middleware, Request and response handling. RESTful API development. Authentication basics. Introduction to enterprise backend frameworks: Overview of Java Spring Boot.

Case Study: Develop server-side API for a blogging or e-commerce application using Node.js and Express.

Unit-IV: Database Integration and Web Services (9 Hours)

Introduction to database connectivity in web applications. SQL Databases: MySQL / PostgreSQL,CRUD operations, NoSQL Databases: MongoDB concepts ,Document-based data model. Integration of server-side applications with databases. Web Services: REST architecture, API communication, JSON based services. API testing tools (Postman).

Case Study: Develop a database driven web application for managing user data or product catalog.

Unit V: Full Stack Development and Deployment (9 Hours)

Full stack architecture. Integration of frontend and backend technologies. MVC architecture. Session management and authentication. Version control using Git. Web security fundamentals: Cross-site scripting (XSS) SQL Injection , Authentication security. Introduction to web hosting and cloud deployment. Overview of containerization and DevOps basics

Case Study: Develop and deploy a full stack web application such as:

- Blogging platform
- Online store
- Student portal system

Learning Resources

Text Books:

1. 1. Jeffrey C.Jackson, "Web Technologies: A Computer Science Perspective", Second Edition, Pearson Education, 2007, ISBN 978-0131856035
2. 2. Robert W. Sebesta,“ Programming the World Wide Web”, 4th Edition, Pearson education, 2008

Reference Books

1. 1. Marty Hall, Larry Brown, “Core Web Programming", Second Edition, Pearson Education, 2001, ISBN 978-0130897930.
2. 2. H.M. Deitel, P.J. Deitel and A.B. Goldberg, "Internet & World Wide Web How To Program", Third Edition, Pearson Education, 2006, ISBN 978-0131752429.
3. 3. Chris Bates, “Web Programming Building Internet Applications”, 3rd Edition, Wiley India, 2006.
4. 4. Xue Bai et al, “The web Warrior Guide to Web Programming”, Thomson, 2003.

e-Books:

1. https://www.w3.org/html/
2. HTML, The Complete Reference http://www.htmlref.com/

3. http://w3schools.org/
4. http://php.net/
5. https://jquery.com/
6. https://developer.mozilla.org/en-US/docs/AJAX
7. http://www.tutorialspoint.com/css/

MOOC :

1. http://www.nptelvideos.in/2012/11/internet-technologies.html
2. https://freevideolectures.com/course/2308/internet-technology/25video lecture by Prof. Indranil Sengupta, IIT, Kharagpur
3. https://www.digimat.in/nptel/courses/video/106105191/L01.html
4. http://www.nptelvideos.com/php/php_video_tutorials.php

Savitribai Phule Pune University **Third Year - Computer Science and Design Engineering (2024 Pattern)**		
PCC303CSD- Computer Networks		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses : Digital Electronics , Discrete Mathematics, Computer Organization

Companion Course:

Course Objectives:

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, TELNET, DHCP, SNMP) and emerging network applications.

Course Contents

Unit I - : Introduction to Computer Networks and Physical Layer (09 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.

Unit–II: Data Link Layer (9 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.

Unit–III: Network Layer (9 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: Case study on IPv6 adoption in ISPs, routing comparison between RIP and OSPF in enterprise networks.

Unit-IV: Transport Layer Protocols (9 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: Case study on TCP congestion control in 4G/5G networks, RTP in video conferencing applications.

Unit V: Application Layer (9 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: Case study on DNS security attacks (DNS spoofing), HTTP caching in CDNs, SMTP in enterprise email systems.

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.

e-Books:

1. https://people.cs.clemson.edu/~jmarty/courses/kurose/KuroseCh1-2.pdf
2. http://eti2506.elimu.net/Introduction/Books/Data Communications and Networking By Behrouz A.Forouzan.pdf
3. http://intronetworks.cs.luc.edu/current/ComputerNetworks.pdf
4. https://www.tutorialspoint.com/data_communication_computer_network/data_communicati on_comp

MOOC :

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

Savitribai Phule Pune University **Third Year - Computer Science and Design (2024 Pattern)**		
PCC304CSD- Artificial Intelligence Laboratory		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	01	Term Work: 50 Marks Practical : 25 Marks

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: Upon successful completion of this course, students 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**: To apply AI planning techniques for solving solve classical planning problems and relate them to real-world industrial AI applications.

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

List of Assignments (Part A) - Any FOUR

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.

List of Assignments (Part B) - Any FOUR

1. Use BFS to solve a planning problem. Examples can be 8 puzzle problem, Robot path planning, Blocks world etc.
2. Use DFS for problem solving. Examples can be water jug problem , Missionaries and Cannibals ,maze solving, blocks world etc.
3. To implement Backtracking search to solve the 8-Queens constraint satisfaction problem.
4. To develop a simple chatbot using predefined rules and pattern matching.
5. To design an intelligent agent that guesses a number using feedback (higher/lower).
6. To implement Goal Stack Planning for solving the Blocks World problem and achieve a specified goal configuration.
7. To develop a game-playing agent using the Minimax algorithm for optimal decision-making.

Part C - Mini-Project (In Team of 3-4 Students) - Any ONE but not limited to

1. Medical Diagnosis System (Rule-Based Expert System) using forward and backward chaining
2. To develop a simple rule-based recommendation system (e.g., movie or book suggestion system).

Savitribai Phule Pune University **Third Year - Computer Science and Design (2024 Pattern)**		
PCC305CSD- Web Technology Laboratory		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 04 Hours/Week	02	Term Work: 25 Marks Practical: 25 Marks

Prerequisite Courses : Digital Electronics and Logic Design, Discrete Mathematics, Computer Organization & Microprocessor, Internet of Things

Companion Course: Computer Network

Course Objectives:

1. To learn the web based development environment
2. To use client side and server side web technologies
3. To design and develop web applications using front end technologies and backend databases

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

- CO1**:** Understand the importance of website planning and website design issues
- CO2: Apply the client side and server side technologies for web application development
- CO3: Analyze the web technology languages, frameworks and services
- CO4:Create three tier web based applications

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 and Practical Examination

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy need

to address the average students and inclusive of an element to attract and promote the intelligent students. Use of open source software is encouraged. Based on the concepts learned. Mini project should be implemented by the students in a group of 2-3 students. 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.

List of Assignments :Any 09 Assignments and Mini project Compulsory	
1	Case study:Before coding of the website, planning is important, students should visit different websites (Min. 5) for the different client projects and note down the evaluation results for these websites, either good website or bad website in following format: Sr. No.,Website URL,Purpose of Website,Things liked in the website,Things disliked in the website,Overall evaluation of the website (Good/Bad) From the evaluation, students should learn and conclude different website design issues, which should be considered while developing a website.
2	Implement a web page index.htm for any client website (e.g., a restaurant website project) using following, a HTML syntax: heading tags, basic tags and attributes, frames, tables, images, lists, links for text and images, forms etc. b Use of Internal CSS, Inline CSS, External CSS
3	Create a responsive webpage using CSS3 (Flexbox/Grid) and Bootstrap/Tailwind for a business/blog layout.
4	Implement an application in Java Script using following: a) Design UI of application using HTML, CSS etc.b) Include Java script validation c) Use of prompt and alert window using Java Script
5	Develop a web page with form validation using JavaScript and DOM manipulation (e.g., registration form).
6	Create a simple Node.js server to handle HTTP requests and send responses (basic routing)..
7	Develop RESTful APIs using Express.js for a simple application (e.g., user data CRUD operations).
8	Design a MySQL/MongoDB database and perform CRUD operations for a sample application (student/product data).
9	Integrate backend (Node.js) with database and perform data insertion, retrieval, update, and delete operations.
10	Develop a full-stack application integrating frontend (HTML/CSS/JS) and backend (Node.js) with API communication.
11	Implement basic security features (input validation, prevention of XSS/SQL Injection) and version control using Git.
12	Design and develop a Full Stack Blogging or Student Portal System with user authentication, CRUD operations, REST API, database integration, and deployment (basic hosting).
13	Mini Project

Savitribai Phule Pune University **Third Year - Computer Science and Design (2024 Pattern)**		
PEC321ACSD - Computer Vision		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses:

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

Companion Course : Elective-I Lab

Course Objectives: The course aims to:

1. To understand Human and computer vision.
2. To understand Low level image processing
3. To study image segmentation and feature representation
4. To understand object recognition & Dynamic Scene Analysis
5. To study image understanding strategies and 3D vision

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

- CO1. Understand and apply the principles of image formation, representation, and processing in computer vision systems
- CO2. Understand Low Level Image Processing Tasks
- CO3. Apply image segmentation techniques and feature representation methods
- CO4. Understand various object recognition techniques and motion analysis.
- CO5. Understand and apply various control strategies for image understanding, techniques for 3D vision and reconstruction, and methods

Course Contents

Unit I - Introduction to Computer Vision (09 Hours)

Introduction: Human Vision Vs Computer Vision, Limitations of Human Vision System, Types of Computer Vision, Computer Vision Pipeline, History of Computer Vision. Computer Vision Applications: object detection, Recognition, Surveillance etc.,

Image formation and Representation: Cameras: Pinhole cameras, cameras with lens, Human Eye, Geometric Camera Models: Elements of Analytical Euclidian Geometry, Camera Parameters and Perspective Projections, Radiometry: measuring light Sources, shadows, and shading,

Digital Image Processing: Components, Elements of visual perception, Image Sensing and Acquisition, Image Sampling and Quantization, Relationships between pixels, different Color Models, Image Types

Case Study: Smart Traffic Monitoring System: From Human Eye to Computer Vision Pipeline

Unit II Low-Level Image Processing and Edge Detection (09 Hours)

Low Level Image Processing: Image filtering, 2D Convolution, smoothing, sharpening Spatial and frequency domain filtering, Histogram processing (equalization, matching) Edge detection (Sobel, Canny), edge in multi spectral image, other local pre-processing operators, line, and corner detection. Introduction to Open CV, Tools to Open and Display Images using Python or Eclipse C/C++

Case Study: Enhancing Chest X-Ray Images for Early Pneumonia Detection using Low-Level Image Processing

Unit III - Segmentation and Feature representation (09 Hours)

Image Segmentation: Thresholding techniques Region-based segmentation (region growing, splitting and merging) Clustering-based segmentation (k-mean), Edge-based segmentation, Graph based segmentation.

Color and Texture Features, Shape Feature representation: Region identification, contour- based shape representation and description, region-based shape representation and Description, shape classes

Case Study: Automated Segmentation and Feature-Based Characterization of Brain Tumors in MRI Images for Computer-Aided Diagnosis

Unit IV - Object Recognition and Motion Analysis (09 Hours)

Knowledge representation, review of statistical object recognition, Bays classifier, KNN classifier, hierarchical and non-hierarchical approach, clustering syntactic object recognition, recognition as a graph matching Estimating motion vectors using sequential search algorithm, logarithmic search algorithm, and hierarchical search algorithm , Motion analysis and differential motion analysis methods , Trajectory detection, Kalman filters for motion estimation and tracking

Case Study: Intelligent Video Surveillance System for Crowd Anomaly Detection and Tracking in a Smart City

Unit V -Image Understanding Control Strategies, 3D Vision (09 Hours)

Image Understanding Control Strategies: Parallel and serial processing control , Hierarchical control and Bottom-up control • Model-based control, Combined control, and Non-hierarchical control , Active Appearance Models (AAM) ,Semantic image segmentation and understanding.3D Vision: Methods for 3D vision and projection schemes , Shape from shading, photometric stereo, shape from texture, and shape from focus , Active range finding , Surface representations: point-based representation and volumetric representations , 3D object recognition and 3D reconstruction

Case Study: Real-Time 3D Reconstruction and Dynamic Object Tracking for Autonomous Drone Navigation in Disaster Response

Learning Resources

Text Books:

1. Milan Sonka, Vaclav Hlavac, Roger Boyle, “Image Processing, Analysis, and Machine Vision”, 4th Edition, Cengage Learning, USA, 2014
2. Forsyth and Ponce, “Computer Vision: A modern Approach ” –PHI.
3. R. Szeliski, “Computer vision: algorithms and applications”, ISSN 1868-095X, 2nd Edition,
4. Springer Nature Switzerland AG, 2022

Reference Books

1. R. Davies, “Computer & Machine Vision”, Fourth Edition, Academic Press, 2012.
2. Mark Nixon and Alberto S. Aquado, “Feature Extraction & Image Processing for Computer Vision” Third Edition, Academic Press, 2012.

3. Simon J. D. Prince, “Computer Vision: Models, Learning, and Inference”, Cambridge University Press,2012.

4. Rafael C. Gonzalez, Richard E. Woods, “Digital Image Processing”, 3rdEdition, Pearson, ISBN: 978-81-317-2695-2

MOOC / NPTEL/YouTube Links

1. NPTEL Course: Computer Vision and Image Processing - Fundamentals And Applications by Prof. M. K. Bhuyan https://onlinecourses.nptel.ac.in/noc26_ee31/preview

2. NPTEL Course: Computer Vision by Prof. Jayanta Mukhopadhyay https://onlinecourses.nptel.ac.in/noc

E-Books

1. Richard Szeliski ,Computer Vision: Algorithms and Applications (2nd Edition, 2022)https://szeliski.org/

2. Simon J.D. Prince, Computer Vision: Models, Learning, and Inferencehttps://raw.githubusercontent.com

Savitribai Phule Pune University **Third Year - Computer Science and Design (2024 Pattern)**		
PEC321BCSD - Multimedia Techniques		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses: Computer Graphics

Companion Course: Elective-I Lab

Course Objectives: The course aims to:

1. To understand input and output devices, device drivers, control signals and protocols, DSPs
2. To study and use standards (e.g., audio, graphics, video)
3. To implement applications, media editors, authoring systems, and authoring by studying streams/structu capture/represent/transform, spaces/domains, compression/coding
4. To design and develop content-based analysis, indexing, and retrieval of audio, images, animation, and video
5. To Understand IoT architecture's and Multimedia Internet of things

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

- CO1. Describe the media and supporting devices commonly associated with multimedia information and systems.
- CO2. Demonstrate the use of content-based information analysis in a multimedia information system.
- CO3. Critique multimedia presentations in terms of their appropriate use of audio, video, graphics, color, and other information presentation concepts.
- CO4. Implement a multimedia application using an authoring system.
- CO5. Implement Multimedia Internet of Things Architectures.

Course Contents

Unit I - Introduction to multimedia (09 Hours)

What is Multimedia and their Components, History of Multimedia; Hypermedia, WWW, and Internet; Multimedia Tools: Static (text, graphics, and still images), Active (sound, animation, and video, etc.); Multimedia Sharing and Distribution; Multimedia Authoring Tools: Adobe Premiere, Adobe Director, Adobe Flash.

Case Study: To study and install open-source multimedia Tools

Unit II Graphics and Data Representation Techniques (09 Hours)

What are Graphics data types, 1-bit Images, 8 –bit grey level ,16-bit grey level images, Image data type, Image data type:8 bit & 24-bit color images, Higher bit depth images, Color Lookup tables.

File Formats: GIF, JPEG, PNG, TIFF, PSD, APS, AI, INDD, RAW, Windows BMP, Windows WMF, Netpbm format, EXIF, PTM, Text file format: RTF, TGA Applications/Use of text in Multimedia

Case Study: To study conversion of image file formats from one to Other.

Unit III Multimedia Representations Techniques - (09 Hours)

Principal concepts for the analog video: CRT, NTSC Video (National Television System Committee), PAL Video (Phase Alternating Line), SECAM Video (System Electronic Couleur Avec Memoire), Digital Video: Chroma Subsampling, High-Definition TV, Ultra High Definition TV (UHDTV), Component Video: High-Definition Multimedia Interface (HDMI),3D Video and TV: various cues, Basics of Digital Audio: What is Sound?, Nyquist Theorem, SNR, SQNR, Audio Filtering, Synthetic Sounds, MIDI Overview: Hardware, Structure, Conversion to WAV, Coding of Audio: PCM, DPCM, DM (Delta Modulation)

Case Study: Install and use Handbrake (link is https://handbrake.fr) software to understand the concept of interlaced, deinterlace, noise filters, bitrate, and frame rate for any sample 30 min video, and note down the observations from the output video.

Unit IV - Compression Algorithms - (09 Hours)

Introduction to multimedia – Graphics, Image and Video representations – Fundamental concepts of video, digital audio – Storage requirements of multimedia applications – Need for compression – Types of compression algorithms- lossless compression algorithms RLC, VLC, DBC, AC, lossless image compression, differential coding of Images, lossy compression algorithms-Rate distortion theory, Quantization ,Transform coding, wavelet based coding, embedded Zerotress of wavelet coefficients . Image compression standard -JPEG standard, JPEG 2000 standard, LS standard, Bilevel image compression standard. Introduction to video compression - video compression based on motion compensation, Search for motion vectors, MPEG Video coding I , MPEG 1,2,4,7 onwards. Basic Audio Compression Techniques -ADPCM in speech coding, Vocoders, MPEG audio compression

Case Study: Implementation of compression algorithms

Unit V -Image Understanding Control Strategies, 3D Vision (09 Hours)

IoT and Multimedia IoT Architecture: IoT Architecture; M-IoT Architectures: Multi-Agent Based, AI-Based Software-Defined, Big Data Layered; Applications of M-IoT: Road Management System, Multimedia IoT in Industrial Applications, Health Monitoring.

Case Studies- Traffic Monitoring System

Learning Resources

Text Books:

1. Tay Vaughan, “Multimedia making it work”, Tata McGraw-Hill, 2011, ISBN: 978-0-07-174850-6 MHID: 0-07-174850-4, eBook print version of this title: ISBN: 978-0-07-174846-9, MHID: 0-07-174846-6

2. Ze-Nian Li, Mark S. Drew and Jiang chuan Liu, “Fundamentals of Multimedia”, Second Edition, Springer, 2011, ISSN 1868-0941 ISSN 1868-095X (electronic), ISBN 978-3-319-05289-2 ISBN 978-3-319-05290-8 (eBook), DOI 10.1007/978-3-319-05290-8, Pearson Education, 2009

Reference Books

1. 1. Ali Nauman et al. “Multimedia Internet of Things: A Comprehensive Survey”, Special Section on Mobile Multimedia: Methodology and Applications, IEEE Access, Volume 8, 2020

2. 2. Kelly S. Hale (Editor), Kay M. Stanney (Editor). 2014. Handbook of Virtual Environments: Design, Implementation, and Applications, Second Edition (Human Factors and Ergonomics) ISBN-13: 978-1466511842. Amazon

E-Books

1. https://users.dimi.uniud.it/~antonio.dangelo/MMS/materials/Fundamentals_of_Multimedia.pdf

2. https://mu.ac.in/wp-content/uploads/2021/04/Multimedia.pdf
3. https://www.baschools.org/pages/uploaded_files/chap13.pdf

MOOC Courses:

1. 1. https://nptel.ac.in/courses/117105083

Savitribai Phule Pune University **Third Year - Computer Science and Design (2024 Pattern)**		
PEC321C CSD:Optimization Algorithms		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses: Students should have basic knowledge of Calculus, Linear Algebra, and Probability and Statistics, along with programming fundamentals.

Companion Course: Elective-I Lab

Course Objectives: The course aims to:

1. Understand mathematical foundations of optimization
2. Apply optimization techniques to engineering problems
3. Learn classical and modern optimization algorithms
4. Explore AI-based and heuristic optimization techniques
5. Implement algorithms using programming tools (Python/MATLAB)

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

CO1. Understand and formulate real-world problems as optimization problems with appropriate objective functions and constraints.

CO2. Apply classical optimization techniques such as gradient-based methods and linear programming to solve engineering problems.

CO3. Implement and evaluate metaheuristic algorithms such as Genetic Algorithm, PSO, and Ant-Colony Optimization.

CO4. Apply optimization techniques in Machine Learning, deep learning, and real-world applications like scheduling and resource allocation

CO5. Develop and compare different optimization algorithms based on performance and problem requirements.

Course Contents

Unit I -Introduction to Optimization (09 Hours)

Definition and importance of optimization ,Real-world applications (AI, networking, scheduling, ML) , Types of optimization problems: ,Linear vs Non-linear ,Constrained vs Unconstrained ,Mathematical formulation: ,Objective function, constraints, decision variables ,Convex vs non-convex optimization,Optimization workflow.

Unit II Classical Optimization Techniques (09 Hours)

Unconstrained optimization: Gradient Descent Method ,Newton's Method Constrained optimization: Lagrange Multipliers ,Karush-Kuhn-Tucker (KKT) conditions Linear Programming (LP): Formulation Graphical method Simplex Method (basic idea)

Unit III Advanced Optimization Methods (09 Hours)

Integer Programming ,Dynamic Programming ,Non-linear optimization techniques, Convex optimization basics, Duality theory, Interior Point Methods (overview)

Unit IV Metaheuristic & Evolutionary Algorithms (09 Hours)

Need for heuristic methods Genetic Algorithms (GA): Selection, crossover, mutation Particle Swarm Optimization (PSO) Ant Colony Optimization (ACO) Simulated Annealing Comparison of metaheuristic methods

: .

Unit V - Optimization in AI & Real Applications (09 Hours)

Optimization in Machine Learning: Loss functions ,Backpropagation optimization. Stochastic Gradient Descent (SGD), Optimization in Deep Learning Applications: Job scheduling (Hadoop/Cloud) ,Network routing ,Resource allocation Case studies & mini-project

Case Study: UPI and mobile banking applications – architecture, security mechanisms, and cloud integration for real-time transactions.

Learning Resources

Text Books:

1. . Hillier, Frederick S., and Lieberman, Gerald J., "Introduction to Operations Research", 10th Edition,
2. McGraw-Hill, 2015, ISBN: 978-0073523453 Boyd, Stephen, and Vandenberghe, Lieven, "Convex Optimization", 1st Edition, Cambridge University Press, 2004, ISBN: 978-0521833783.
3. . Rao, Singiresu S., "Engineering Optimization: Theory and Practice", 4th Edition, Wiley, 2009, ISBN: 978-0470183526.

Reference Books

1. . Nocedal, Jorge, and Wright, Stephen J., "Numerical Optimization", 2nd Edition, Springer, 2006, ISBN:978-0387303031.
2. Yang, Xin-She, "Nature-Inspired Optimization Algorithms", 1st Edition, Elsevier, 2014, ISBN: 978-0124167438.
3. . Sra, Suvrit, Nowozin, Sebastian, and Wright, Stephen J., "Optimization for Machine Learning", 1stEdition, MIT Press, 2012, ISBN: 978-0262016469.

MOOC / NPTEL/YouTube Links

1. . NPTEL Course: "Optimization Theory and Algorithms", by Prof. (IIT), Available at: https://onlinecours
2. NPTEL Course: "Convex Optimization", by Prof. Joydeep Dutta, IIT Kanpur, Available at: https://nptel.a
3. . NPTEL Course: "Optimization from Fundamentals", by Prof. Ankur A. Kulkarni, IIT Bombay, Available at: https://onlinecourses.nptel.ac.in/noc21_me10/preview
4. NPTEL Course: "Optimization Methods for Civil Engineering", by Prof. Rajib Kumar Bhattacharjya, IIT Guwahati, Available at: https://onlinecourses.nptel.ac.in/noc24_ce92/preview

Savitribai Phule Pune University **Third Year - Computer Science & Design (2025 Pattern)**		
PEC321DCSD Embedded System		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses: Digital Electronics, Computer Organization & Architecture

Companion Course: Elective 1 Lab

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

Course Contents

Unit I - Introduction to Embedded Systems (09 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 (09 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 (09 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 (09 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 (09 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 - Computer Science and Design Engineering (2024 Pattern)**		
PEC322CSD- 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 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 - PEC322A CSD: Elective Lab Computer Vision (Elective I)

Suggested List of Laboratory Experiments/Assignments (Any 06 Assignments and Mini Project are Compulsory)

1. OpenCV Installation and working with Python
2. Basic Image Processing - loading images, Cropping, Resizing, Thresholding, Contour analysis, Bolb detection
3. Image Annotation – Drawing lines, text circle, rectangle, ellipse on images
4. Image Enhancement - Understanding Color spaces, color space conversion, Histogram equialization, Convolution, Image smoothing, Gradients, Edge Detection

5. Image Features and Image Alignment – Image transforms – Fourier, Hough, Extract ORB Image features, Feature matching, cloning, Feature matching based image alignment

6. Image segmentation using Graphcut / Grabcut

7. Develop a Python + OpenCV program that processes a video (or webcam feed) containing multiple moving objects (e.g., cars, people, or balls). The system should:• Detect moving objects using background subtraction or frame differencing.• Extract features and recognize/classify the detected objects using statistical methods.• Estimate motion vectors using block matching with different search strategies.

8. Implement Image Segmentation: Simple thresholding (global and adaptive), K-means clustering for image segmentation, Display segmented results, Outcome: Students learn basic segmentation methods

9. Implement Edge Detection and analyze each technique : Implement Sobel, Prewitt (using kernels), and Canny edge detection, Compare outputs for different thresholds, Outcome: Students learn boundary detection techniques

10. Mini- Project

Guidelines for Laboratory Conduction - PEC-322B-CSD : Multimedia Techniques (Elective I)

Suggested List of Laboratory Experiments/Assignments (Any 06 Assignments and Mini Project are Compulsory)

1. To study and install open-source multimedia tools and create an application using appropriate tool to design the college webpage
2. To create JPEG Image that demonstrate various features of an Image editing tool.
3. Create or play a sample MIDI format sound file using LMMS / MuseScore / Tuxguitar software tool. Edit the sample file by applying effects like bend, slide, vibrato, and hammer-on/pull-off. Export / Convert final MIDI to WAV file format.
4. Implement transform coding, quantization, and hierarchical coding for the encoder and decoder of three-level Hierarchical JPEG.
5. Create a web page for a clothing company which contains all the details of that company and atleast five links to other web pages.
6. Design and Implementation of a Basic Image Compression System using JPEG Standard and Comparative Study with Lossless Techniques
7. Mini- Project

Guidelines for Laboratory Conduction - PEC321C CSD : Optimization Algorithms (Elective-I)

Suggested List of Laboratory Experiments/Assignments (Any 06 Assignments and Mini Project are Compulsory)

1. Apply Gradient Descent on a representative function or dataset and analyze the influence of learning rate on convergence behavior and solution accuracy.
2. Formulate and solve a real-world optimization problem, such as resource allocation, using Linear Programming techniques and interpret the optimal solution obtained through computational tools.
3. Solve constrained optimization problems using Lagrange multipliers and examine the impact of constraints on the feasibility and optimality of solutions.
4. Implement advanced numerical methods such as Newton's Method and compare their convergence efficiency with first-order optimization techniques.
5. Address discrete optimization problems, such as the knapsack problem, using Integer Programming and evaluate the effectiveness of the obtained solutions.
6. Develop and implement a Genetic Algorithm, and investigate the effect of parameters such as population size, crossover rate, and mutation rate on solution quality.
7. Utilize Particle Swarm Optimization (PSO) to solve benchmark optimization problems and compare its performance with other evolutionary techniques.
8. Apply Simulated Annealing to optimization problems and analyze its capability to escape local optima and achieve near-global solutions
9. Implement Ant Colony Optimization (ACO) for solving combinatorial problems such as shortest path or traveling salesman problem, and evaluate its efficiency.
10. Perform a comparative analysis of various optimization algorithms based on metrics such as convergence speed, computational efficiency, and solution accuracy.
11. Apply optimization techniques to practical applications such as scheduling, resource allocation, or machine learning model tuning, and assess their impact on overall system performance.

Guidelines for Laboratory Conduction - PEC-321 D CSD: Embedded Systems (Elective-I)

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.

Suggested List of Laboratory Experiments/Assignments (Any 06 Assignments and Mini Project are Compulsory)

1. Write an Assembly Language Program (ALP) to
   (a) Multiply two 16-bit numbers.
   (b) 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
6. Interface a Stepper motor and rotate it in clockwise and anti-clockwise direction.
7. Interface a DAC and generate Triangular and Square waveforms.
8. Display the Hex digits 0 to F on a 7-segment LED interface, with a suitable delay in between.
9. Interface a simple Switch and display its status through Relay, Buzzer and LED
10. Write a program for task synchronization in RTOS with the use of binary and counting semaphore, Priority Inversion.

Mini Project

Savitribai Phule Pune University **Third Year - Computer Science and Design Engineering (2024 Pattern)**		
MDM-331-CSD: Internet of Things		
Teaching Scheme	**Credits**	**Examination Scheme**
Tutorial : 01 Hour/Week **Practical**: 4 Hours/Week	01 02	Term Work : 50 Marks

Prerequisite Courses: Digital Electronics

Course Objectives: The course aims to:

1. .To understand fundamentals of IoT and embedded systems including essence, basic design strategy and process modelling.
2. To learn to implement secure infrastructure for IoT applications.
3. To introduce learners to a set of advanced topics in IoT and lead them to understand research in networks.
4. To develop a comprehensive approach towards building small low cost IoT applications.
5. To learn real world application scenarios of IoT along with its societal and economic impact using case studies & real time examples.

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

CO1. Understand basic fundamentals of embedded systems and IoT Networking.

CO2. Apply knowledge of IoT programming to execute basic programs on IoT boards.

CO3. Explain Communication protocols in IoT, its enabling technologies for developing systems with its emergence.

CO4. Analyze different computing models for building networks and cloud for IoT.

CO5. Demonstrate different case studies in the field of IoT.

Course Contents

Unit I - Introduction to IoT (03 Hours)

Embedded System, Definition, Characteristics, Modern IoT Applications, Sensors and Actuators. IoT Architecture and block diagram Networking for IoT: Connectivity Terminologies. IoT Network Configuration

Unit II - Programming in IoT (03 Hours)

Introduction to Arduino Programming: Features of Arduino, Board details, Setup and IDE.
Introduction to Python programming: Python IDE, Basic programs on Raspberry Pi, Setup and Installation of OS, Pin Configuration, Implementation of IoT Applications with Raspberry Pi.

Unit III - IoT Protocols (03 Hours)

Protocol Standardization for IoT, M2M and WSN Protocols, RFID Protocol, Modbus Protocol, Zigbee Architecture. IP based Protocols: MQTT (Secure), 6LoWPAN, LoRa

Unit IV Cloud for IoT (03 Hours)

Introduction to SDN: Overview, Architecture, attributes, challenges.
SDN for IoT: Benefits, Different Approaches, SDN for Mobile Networking: ODIN, Ubi-Flow, Mobi-Flow, Data Handling and Analytics, Cloud for IoT.

Unit V Case Studies (03 Hours)

Smart Cities and Smart Homes, Connected Vehicles, Smart Grid, Industrial IoT, Agriculture, Healthcare, Activity Monitoring

Learning Resources

Text Books:

1. T1. Arshdeep Bahga, Vijay Madisetti, —Internet of Things – A hands-on approach‖,Universities Press, ISBN: 0: 0996025510, 13: 978-0996025515.

2. T2. Olivier Hersent, David Boswarthick, Omar Elloumi, The Internet of Things: Key Applications and Protocols, 2nd Edition,Wiley Publication,ISBN: 978-1-119-99435-0

Reference Books

Guidelines for Laboratory Conduction

List of Assignment - Group A (Compulsory)

1. Study of Raspberry-Pi/ Beagle board/ Arduino and other microcontroller (History & Elevation)
2. Study of different operating systems for Raspberry-Pi /Beagle board/Arduino. Understanding the process of OS installation.
3. Study of Connectivity and configuration of Raspberry-Pi /Beagle board/Arduino circuit with basic peripherals like LEDs. Understanding GPIO and its use in the program.

Group B - (Any 08)

1. Write a program using Arduino to control LED (One or more ON/OFF, Or Blinking).
2. Create a program that illuminates the green LED if the counter is less than 100, illuminates the yellow LED if the counter is between 101 and 200 and illuminates the red LED if the counter is greater than 200.
3. Create a program so that when the user enters 'b' the green light blinks, 'g' the green light is illuminated 'y' the yellow light is illuminated and 'r' the red light is illuminated.
4. Write a program that asks the user for a number and outputs the number squared that is entered.
5. Write a program to control the color of the LED by turning 3 different potentiometers. One will read for the value of Red, one for the value of Green, and one for the value of Blue.
6. Write a program read the temperature sensor and send the values to the serial monitor on the computer.
7. Write a program so it displays the temperature in Fahrenheit as well as the maximum and minimum temperatures that has been
8. Write a program to show the temperature and shows a graph of the recent measurements.
9. Write a program using piezo element and use it to play a tune after someone knocks.
10. Understanding the connectivity of Raspberry-Pi /Beagle board circuit / Arduino with IR sensor. Write an application to detect obstacle and notify user using LEDs.

Savitribai Phule Pune University **Third Year - Computer Science and Design Engineering (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	Business Law	Commerce & Management
2	IPR and Cyber Laws	Law / Faculty of Humanities
3	Business Marketing	Commerce & Management
4	Agri Business Management: Banking Operation and finance	Commerce & Management
5	Banking ,Finance & Insurance	Commerce & Management
6	Statistics and Computer Applications	Commerce & Management
7	Business Administration	Commerce & Management
8	Cost & Works Accounting	Commerce & Management
9	Sustainability Development	
10	The Constitution of India	
11	Digital Personal Data Protection	
12	Product Costing for Mechanical Engineering	
13	Material and Logistics	

Savitribai Phule Pune University **Third Year - Computer Science & Design Engineering (2024 Pattern)**		
ELC342CSD: Technical Seminar		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	01	Oral/Presentation : 25 Marks

Course Objectives: The course aims to:

1. To develop research orientation and technical communication skills in emerging Computer Engineering 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 and relevant technical topics through literature survey.
- CO2: Analyze and synthesize information from research papers, journals, and credible sources.
- CO3: Demonstrate effective technical communication skills through oral presentation.
- CO4: Prepare a structured technical report following academic writing standards.
- CO5: Use modern tools (presentation software, plagiarism checkers, referencing tools) for seminar preparation.

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

- Topic must be from emerging Computer Engineering and Artificial Intelligence domains (last 3–5 years).
- Must involve a minimum of 5 recent research papers (IEEE, ACM, Elsevier, Springer etc). They should summarizing paper – Reading abstracts and finding ideas, conclusion, Advantages of Their approach, and the drawbacks of the papers. Generalize results from a research paper to related research problems. Comparing the approach - Identify weaknesses and strengths in recent research articles in the subject. Practical sessions on how to read, analyze and summarize research papers.
- Should not be a basic textbook topic.
- Must include: Problem statement, State-of-the-art analysis, Comparative study, Ethical & societal impact, Interdisciplinary themes aligned with NEP encouraged.
- Topic approval by a faculty panel

Seminar Process

- **Stage 1: Orientation & Topic Finalization (Week 1–2)**
  - 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 and Design

Semester VI

Savitribai Phule Pune University **Third Year - Computer Science & Design Engineering (2024 Pattern)**		
PCC351 CSD: Augmented and Virtual Reality		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses: Probability and Statistics, Data Science, Python Programming
Course Objectives: The course aims to:

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. 5.To analyze applications, challenges and ethical considerations of AR/VR systems

Course Outcomes: Upon successful completion of this 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 (9 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 (09 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 (09 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, cybersickness 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 -(09 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 - (09 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 & Design Engineering (2024 Pattern)**		
PCC352CSD: UI/UX Design		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 02 Hours/Week	02	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses: Design Thinking

Companion Description: UI/UX Design Laboratory

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

- CO1: Explain fundamental concepts of UI/UX and user-centered 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 - Introduction to UI/UX Design (9 Hours)

What is UI/UX Design: Importance of User-Centric Design, Goals of User Interface Design , Design Thinking Process, Core Principles, Role of UX in Product Development Lifecycle, mental and conceptual model.

Usability Design Principles: 4 Design Principles, Schneiderman's Golden Rules, 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.

UI Design Tools: Introduction to modern UI/UX design tools such as Figma, Adobe XD, Sketch, and InVision

Case Study: A good and a bad User Interface Design

Unit II -Usability Engineering, Evaluation and Testing (9 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. Usability 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, Good and poor design. paper prototyping, introduction to wireframes and interface layout, applying Nielsen's usability heuristics, refining UI based on user feedback.

Case Study: Redesign of a Food Delivery Mobile App

Unit III WEB DESIGN: STRATEGIES AND INFORMATION ARCHITECTURE (9 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.

Case Study: Redesign of an E-Learning Platform

Unit IV Wireframing Fundamentals (9 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.. Creating interactive prototypes, collaborative design workflows, design handoff to developers. Design Documentation: User flow diagrams, storyboards, design specifications, and design system basics.

Case Study: Designing a Mobile Banking App

Unit V Designing UX for Tomorrow (9 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.

Case Study: Smart Healthcare Ecosystem UX Design

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.

Savitribai Phule Pune University **Third Year - Computer Science & Design Engineering (2024 Pattern)**		
PCC 353 CSD: AR/VR LAB		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 04 Hours/Week	02	Term Work: 25 Marks Practical: 25 Marks

Prerequisite Courses:
Companion Course :

Course Objectives:

- To familiarize students with Augmented Reality (AR) and Virtual Reality (VR) hardware devices and development platforms.
- To enable students to create basic immersive virtual environments.
- To implement navigation and interaction techniques in VR environments.
- To develop augmented reality applications using marker-based and markerless tracking techniques.
- To design a small AR/VR application integrating various concepts.

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

- CO1: Understand AR/VR hardware components and development platforms.
- CO2: Develop basic virtual environments with transformations, lighting and textures.
- CO3: Implement navigation and interaction techniques in VR environments.
- CO4: Develop augmented reality applications using marker-based and markerless tracking methods.
- CO5: Design and implement an AR/VR application integrating multiple concepts.

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.

Software Requirements-

- AR/VR development platform: Unity
- AR SDKs: ARCore / ARKit
- Operating System: Windows / Ubuntu
- Optional 3D asset creation tool: Blender

Suggested List of Laboratory Experiments(Any 8 and Mini Project is Compulsory)

1. Study of AR/VR hardware devices and development platforms.
2. Installation and configuration of AR/VR development environment using Unity and XR plugins.
3. Creation of a basic 3D virtual environment and application of geometric transformations.
4. Implementation of lighting, materials and textures to enhance realism in a virtual scene.
5. Implementation of VR navigation techniques such as teleportation or locomotion.
6. Implementation of controller-based object interaction such as grab, move and release.
7. Development of a marker-based Augmented Reality application using image tracking techniques.
8. Development of a markerless Augmented Reality application using plane detection or environment tracking using platforms such as ARCore or ARKit integrated with Unity.
9. Demonstration of the working of HTC Vive, Google Daydream or Samsung gear VR.
10. .Develop a scene in Unity that includes:
    (a) cube, plane and sphere, apply transformations on the 3 game objects.
    (b) Add a video and audio source.

11. Develop a scene in Unity that includes a cube, plane and sphere. Create a new material and texture separately for three Game objects. Change the color, material and texture of each Game object separately in the scene. Write a C# program in visual studio to change the color and Material/texture of the game objects dynamically on button click.

12. Develop and deploy a simple marker based AR app in which you have to write a C# program to play video on tracking a particular marker.

13. Develop and deploy an AR app, implement the following using Vuforia Engine developer portal:

    (a) Plane detection

    (b) Marker based Tracking(Create a database of objects to be tracked in Vuforia)

    (c) Object Tracking

Mini Project

1. Design and development of an Augmented Reality or Virtual Reality application for domains such as education, healthcare, training, industry or gaming, demonstrating integration of virtual environment design, interaction techniques and AR/VR visualization concepts.

2. Create a multiplayer VR game (battlefield game). The game should keep track of score, no. of chances/lives, levels (created using different scenes), involve interaction, animation and immersive environment.

3. Create a treasure hunt AR application which should have the following features:

    (a) A help button for instruction box to appear.

    (b) A series of markers which would give hints on being scanned.

    (c) Involve interaction, sound, and good UI.

Savitribai Phule Pune University **Third Year - Computer Science & Design Engineering (2024 Pattern)**		
PCC354CSD: UI /UX Design Laboratory		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	01	Term Work: 25 Marks Oral: 25 Marks

Course Objectives:

1. To study various tools of UI/UX Design
2. To develop skills in creating visually appealing and cohesive user interfaces.
3. To learn to conduct usability testing and evaluation
4. To understand the role of prototyping in the design process
5. To study collaborative features of UI/ UX Tool

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

- CO1 To study various tools of UI/UX Design
- CO2 To develop skills in creating visually appealing and cohesive user interfaces.
- CO3 To learn to conduct usability testing and evaluation
- CO4 To understand the role of prototyping in the design process
- CO5 To study collaborative features of UI/ UX Tool

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

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 Practical Examination

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

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

Guidelines for Laboratory Conduction

- 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.

Suggested List of Laboratory Experiments/Assignments (8 Assignments are Compulsory)

1. Study of various UI/UX design tools : Wireframe , Mockup , Figma Tools.Identify specialized users and related facilities for a selected product /system and make necessary suggestions for its improved accessibility design
2. Create Low-Fidelity and High Fidelity Wireframes: Start by sketching low-fidelity wireframes for each page using pen and paper or any digital tool you prefer. Focus on the layout, placement of key elements, and overall structure. Use basic shapes and placeholders to represent different elements such as navigation menus, search bars, images, buttons, and form fields. Aim for simplicity and clarity in your wireframes.
3. Online Learning Platform: Design a wireframe for an online learning platform that includes course listings, video lectures, quizzes, and progress tracking. (E-learning Website Design in Figma - YouTube)
4. Designing a Social Fitness App: Create wireframes and a prototype for a social fitness app that allows users to track workouts, connect with friends, and share progress. Design the user interface for logging exercises, setting goals, and incorporating social features. (Fitness App Design In Figma || Figma Tutorial || Design & Prototyping - YouTube)
5. Poster or Flyer Mockup: Select a specific event, campaign, or promotional material. Design a poster or flyer using a graphic design tool with mockup capabilities. Create a visually appealing mockup of the poster or flyer in different sizes and formats. Showcase the design within a realistic environment or context, such as a wall or display
6. Poster or Flyer Mockup: Select a specific event, campaign, or promotional material. Design a poster or flyer using a graphic design tool with mockup capabilities. Create a visually appealing mockup of the poster or flyer in different sizes and formats. Showcase the design within a realistic environment or context, such as a wall or display

7. Use Figma tool to Design a user interface for a recipe finder application, allowing users to search for recipes based on ingredients, categories, and dietary restrictions. Include features like recipe details, cooking instructions, and saving favorites (Create a Food & Drink Recipe app with reviews from Figma no code - YouTube)

8. Design user persona for the users of selected product / system.

   (a) How To Create A User Persona (Video Guide) - YouTube

   (b) How to Create A User Persona in 2022 [FULL GUIDE] - YouTube

9. Use Figma tool to Design a user interface for a Redesign of College Website for Better User Experience

10. Use Figma tool for Improving the User Interface of a Fitness Tracking App: Improve the user interface of an existing fitness tracking app by focusing on simplicity, clarity, and motivational elements. Enhance features like tracking workouts, setting goals, and visualizing progress to create a more engaging and intuitive experience (Figma Fitness mobile app Design | design a Fitness app in Figma | UIUX Design 2021 | Techno-fine - YouTube)

11. Collaborative Design Exercise: Form a design team and work on a collaborative design project using Figma. Assign different design tasks to team members, such as wireframing, visual design, or prototyping. Utilize Figma's collaboration features to work together in real-time. Coordinate and provide feedback to each other to refine and improve the design. (Create Teams in Figma & Real-Time Collaboration in Figma for Designers - YouTube)

12. Usability Testing Simulation: Develop a high-fidelity interactive prototype using any UI/UX tool. Prepare a usability testing plan, recruit participants, and simulate usability testing sessions. Analyze the feedback and iterate on the design based on the insights gathered during the testing. (Usability Testing in UX Design Thinking Process - YouTube)

Mini Project:

1. Creating Social media advertisement using online tools and applications

2. Tourism Guide Mobile App Interface Design

3. UX Design for IoT Smart Home Control Application

4. UI/UX Projects Ideas : Online Journal , A Chatbot, An App Layout for Smart Television

Savitribai Phule Pune University **Third Year - Computer Science & Design (2024 Pattern)**		
PEC361 A CSD: Machine Learning		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses: • Probability and Statistics • Python Programming
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: Upon successful completion of this course, students will be able to:

- CO1 Apply fundamental Machine Learning concepts in various learning paradigms and real-world engineering applications.
- CO2 Make use of various types of 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 (09 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 Study: Machine Learning Based Student Performance Prediction and Feature Engineering Analysis

Unit II - Supervised Learning-Regression(09 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 Study: Comparative Study of Regression Techniques for House Price Prediction

Unit III Supervised Learning-Classification(09 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 Study: Comparative Study of Classification Algorithms for Email Spam Detection.

Unit IV Unsupervised Learning and Ensemble Learning (09 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 Study: Customer Segmentation and Sales Prediction

Unit V Reinforcement Learning (09 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 Study: Smart Traffic Signal Control using Q-Learning

Learning Resources

Text Books:

1. Alpaydin, Ethem, “Introduction to Machine Learning”, 2nd Edition, MIT Press, 2014, ISBN: 978-0262028189.

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., “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: 978-1107057135.

E-Books

1. 1. Hastie, Trevor, Tibshirani, Robert, & Friedman, Jerome, “The Elements of Statistical Learning: Data Mining, Inference, and Prediction”, 2nd Edition, Springer, 2009, ISBN: 978-0387848570.

2. 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.pdf.

Savitribai Phule Pune University **Third Year - Computer Science & Design (2024 Pattern)**		
PEC361 B CSD - Cloud Computing		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses: Computer Networks

Companion Course: Programming Elective Course Lab

Course Objectives: The course aims to:

1. To understand the basic concepts of cloud computing and virtualization
2. To understand the implementation of virtualization in cloud computing
3. To learn the application and security on cloud
4. To study risk management in cloud computing
5. To comprehend the modern cloud environment and emerging technologies

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

- **CO1**: Comprehend basic concepts of cloud computing environment
- **CO2**: Analyze Virtualization for cloud and install Virtualization software
- **CO3**: Configure, Test and Deploy applications on Cloud
- **CO4**: Understand and Apply security in cloud applications
- **CO5:** Analyze emerging technologies in modern cloud computing

Course Contents

Unit I - Introduction to Cloud Computing (09 Hours)

Cloud Fundamentals: Definition, Importance of cloud computing, Advantages and Disadvantages of Cloud Computing, Characteristics, Categories of Clouds: Private clouds, Public clouds Cloud Service Models: SaaS, PaaS, IaaS, Cloud Architecture, Cloud Storage: Distributed Data Storage, Data management, Cloud Deployment Models

Case Study: Cloud Computing Model of Amazon

Unit II Virtualization in Cloud Computing (09 Hours)

Virtualization: What's virtualization, Benefits of Virtualization, Types of Virtualization: Processor virtualization, Memory virtualization, Full virtualization, Para virtualization, and Device virtualization, Virtual Clustering, Virtualization Architecture, Containerization and orchestration, Understanding importance of Hypervisors, Virtualization Applications, Issues with Virtualization, Virtualization and Cloud Computing: Virtualizations in Cloud, Virtual Infrastructure, CPU Virtualization, Network and Storage Virtualization

Case Study: Case Study of VMware: Full virtualization, Xen: Para Virtualization, Microsoft HyperV

Unit III Cloud Platforms and Applications - (09 Hours)

Industrial Cloud Platforms: Amazon Web Services (AWS)- AWS infrastructure, Components, Amazon Simple DB, Elastic Cloud Computing (EC2), Amazon Storage System, Amazon Database Services.

Microsoft Azure: Azure core concepts, SQL Azure, and Application Services for managed runtimes. Open Source Platforms: Overview of OpenStack, CloudStack, and Eucalyptus for private cloud deployment.

Cloud Applications:

Data-Intensive & Emerging Applications: Smart Cities & IoT: Integrating sensor data from traffic, waste management, and power grids into a centralized cloud dashboard. AI/ML in the Cloud: Case study on Google Photos (image recognition) or Alexa (Natural Language Processing) using cloud-based TPU/GPU instances. Healthcare & Biology: Gene sequencing, protein folding and ECG analysis in the cloud. Geoscience: Satellite image processing and seismic data analysis using cloud clusters.

Case Study: The Google Case Study Data Processing: The evolution from MapReduce to Dremel and BigQuery. Storage Innovation: Understanding the Google File System (GFS) and BigTable as the backbone of global search.

Unit IV - Security in Cloud Computing - (09 Hours)

Risks in Cloud Computing: Risk Management, Enterprise-Wide Risk Management, Types of Risks in Cloud Computing. Data Security in Cloud: Security Issues, Challenges, advantages, Disadvantages, Cloud Digital persona and Data security, Content Level Security. Cloud Security Services: Confidentiality, Integrity and Availability, Security Authorization Challenges in the Cloud, Secure Cloud Software Requirements, Secure Cloud Software Testing

Case Study: Cloud Security Tool: Acunetix

Unit V - Modern Cloud Environment & Emerging Technologies (09 Hours)

Future Trends in cloud Computing, Mobile Cloud, Comet Cloud, Multimedia Cloud: IPTV, Energy Aware Cloud Computing, Distributed Cloud Computing Vs. Edge Computing, Containers, Dockers, Kubernets, Pod Management

Green Cloud & Sustainability: Sustainable Cloud Architecture, Energy-efficient data centre design and carbon footprint tracking.

Case Studies on DevOps: DocuSign, Forter, Gengo

Learning Resources

Text Books:

1. A. Srinivasan, J. Suresh, “Cloud Computing: A Practical Approach for Learning and Implementation”, Pearson, ISBN: 978-81-317-7651-3
2. Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, “Mastering Cloud Computing”, McGraw Hill Education, ISBN-13:978-1-25-902995-0

Reference Books

1. James Bond ,“The Enterprise Cloud”, O’Reilly Media, Inc. ISBN: 9781491907627
2. Dr. Kris Jamsa, “Cloud Computing: SaaS, PaaS, IaaS, Virtualization and more”, Wiley Publications, ISBN: 978-0-470-97389-9
3. Anthony T. Velte Toby J. Velte, Robert Elsenpeter, “Cloud Computing: A Practical Approach”, 2010, The McGraw-Hill.
4. Gautam Shrof, “ENTERPRISE CLOUD COMPUTING Technology Architecture, Applications”, Cambridge University Press, ISBN: 9780511778476
5. Tim Mather, Subra K, Shahid L.,”Cloud Security and Privacy”, Oreilly, ISBN-13 978-81-8404-815-5

6. Ronald L. Krutz, Russell Dean Vines, “Cloud Security: A Comprehensive Guide to Secure Cloud Computing”, Wiley- India,2010

7. Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Cloud Computing: Principles and Paradigms, Editors: Wile, 2011

E-Books

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://studytm.wordpress.com/wp-content/uploads/2014/03/hand-book-of-cloud-computing.pdf
4. https://arpitapatel.files.wordpress.com/2014/10/cloud-computing-bible1.pdf
5. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-291r2.pdf

MOOC Courses:

1. https://onlinecourses.nptel.ac.in/noc26_cs55/preview
2. http://www.ndl.gov.in/he_document/nptel/nptel/N_C_S_A_E_C_C_A_D_S_N_I_T_C_C_536752663
3. https://onlinecourses.nptel.ac.in/noc26_cs29/preview?
4. https://onlinecourses.nptel.ac.in/noc21_cs15/preview?

Savitribai Phule Pune University **Third Year - Computer Science and Design (2024 Pattern)**		
PEC361C CSD - Cyber Security		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Course Objectives: The course aims to

1. Introduce security fundamentals, threats, basic risks, and ethical practices.
2. Apply cryptographic concepts for confidentiality, integrity, and authentication
3. Examine security controls for systems, applications in real-world scenarios.
4. Equip students with the technical knowledge and skills needed to protect and defend againstcyber threats.
5. Develop skills in incident response lifecycle, and compliance with CERT-In and DPDP regulatoryframeworks

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

- CO1 : Identify common threats and select suitable security controls for a given scenario.
- CO2 : Explain encryption, hashing, digital signatures, and PKI for protecting data.
- CO3 : Analyze common web and API vulnerabilities
- CO4 : Apply secure SDLC and vulnerability management.
- CO5 : Describe data protection techniques, and interpret the regulatory framework under the DPDP

Course Contents

Unit I -Introduction to Cyber Security & Data Privacy (09 Hours)

Cybersecurity – definition, significance, objectives; Cybersecurity vs. Information Security; Cybersecurity in AI and Data Science; CIA Triad; AAA Model; Defense-in-Depth; Least Privilege; Threats; Vulnerabilities; Risk Assessment; Control Measures; Data Privacy – definition, significance; Personally Identifiable Information (PII); Cyber Ethics; Responsible Disclosure; Data Lifecycle Management;Privacy-by-Design.

Case Study: Data Exposure in a Student Mobile Application

Unit II -Data Encryption Techniques (09 Hours)

Cryptography – applications and need in cybersecurity; Cryptographic algorithms and classification; Data in transit, at rest, and in use; Symmetric and asymmetric encryption – AES, RSA, DES; Hashing techniques – bcrypt, SHA-256, integrity mechanisms; Secure password hashing – salting, slow hashes (Argon2id); Digital signatures – principles, applications, benefits; PKI, digital certificates,

Case Study: Analyzing Password Breach (like the 2012 LinkedIn)

Unit III - Secure Engineering and Vulnerability Management (09 Hours)

Secure SDLC: requirements, design review, Secure configuration & hardening: baseline builds, CIS benchmark, least functionality Patch & vulnerability management: identification, triage, remediation, verification, CVSS Secrets management: API keys, credentials, environment variables, rotation 67

Software supply chain: dependency risks, SBOM-Software Bill of Materials, signing, update hygiene Security testing: SAST, DAST, dependency scanning, misconfiguration checks

Case Study: Secure SDLC review for a student portal

Unit IV - Security and Threats Handling (09 Hours)

API Security : Authentication Mechanisms and Patterns, Rate Limiting, Logging, Monitoring ,Classification of Cyber Threats and Attack Methodologies, Malware Typologies: Viruses, Worms, Trojans, Ransomware, Spyware Rootkits ,Social Engineering and Phishing Attacks: Phishing, Spear Phishing, Vishing, AI-Powered Social Engineering Attacks: deepfakes (voice/video), Cloud Security Threats: misconfigurations, insecure APIs, identity and access management.

Case Study: Web Portal Defacement through Input Injection

Unit V - Incident Response and regulatory frameworks (DPDP) (09 Hours)

Incident Response Lifecycle: preparation, detection, containment, eradication, recovery, Incident Response Playbooks Development, AI-generated phishing Incident Response for AI Threats: detection, analysis, mitigation, Security Testing Strategies: identification, prioritization, remediation of vulnerabilities CERT-In Reporting & DPDP Breach Reporting: overview, compliance requirements; DPDP Act 2023 ;DPDP Rules 2025: key provisions, incident response considerations ,Incident Response under Indian Regulatory Frameworks

Case Study: Ransomware incident response plan with evidence collection and recovery steps.

Learning Resources

Text Books:

1. Nina Godbole and Sunit Belapure, Cyber Security: Understanding Cyber Crimes, ComputerForensics and Legal Perspectives, Wiley India.
2. William Stallings, Cryptography and Network Secuity, Pearson India.
3. Tim Mather, Subra Kumaraswamy, Shahed Latif, Cloud Security and Privacy, O'Reilly (India editionsvia Shroff).

Reference Books

1. Charles P. Pfleeger and Shari Lawrence Pfleeger, Security in Computing, Pearson India.
2. Michael Howard and David LeBlanc, Writing Secure Code, Microsoft Press.
3. ISO/IEC 27001 and ISO/IEC 27701 (latest editions).
4. Gordon Lyon, Nmap Network Scanning.

MOOC / NPTEL/YouTube Links

1. OWASP Foundation videos.
2. ZAP Getting Started Guide.
3. .Wireshark User Guide.
4. Greenbone Community documentation.

Savitribai Phule Pune University **Third Year - Computer Science & Design (2024 Pattern)**		
PEC361 D CSD: Industrial IOT		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Prerequisite Courses: IoT
Companion Course :
Course Objectives: The course aims to:

1. Explore the fundamentals of IoT and Industry 4.0.
2. Explore the basics of Industrial IoT.
3. Elaborate Industrial IoT analytic tools.
4. Interpret various IoT protocols for design of Industrial IoT systems.
5. Implement Case studies-Building Automation Systems.

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

- CO1: Describe the concepts, architecture, and enabling technologies of Industry 4.0 and Internet of Things used in smart industries.
- CO2: Understand industrial IoT processes, sensing and actuation systems, IIoT layers, and the role of Artificial Intelligence in industrial automation.
- CO3: Apply IoT gateways, edge computing, cloud computing, and data analytics techniques for industrial data monitoring, control, and predictive maintenance applications.
- CO4: Analyze the performance and applications of industrial communication protocols such as MQTT, OPC Unified Architecture, Modbus, Zigbee, BACnet, and LoRaWAN in IIoT environments.
- CO5: Evaluate intelligent building automation systems and design suitable BAS architectures integrating HVAC, security, fire, and lighting systems using industrial communication protocols.

Course Contents

Unit I -Industry 4.0 (09 Hours)

Introduction to IoT, Technical requirements, IoT background-History and definition, Architecture of IoT ,Industry 4.0: Globalization and Emerging Issues, The Fourth Revolution, LEAN Production Systems,

Smart and Connected Business Perspective, Smart Factories, Cyber Physical Systems and Next Generation Sensors.

Unit II Introduction to Industrial IoT (09 Hours)

Industrial Processes, Industrial Sensing & Actuation, Industrial Internet Systems, Business Models and Reference Architecture, Industrial IoT- Layers: IIoT Sensing, IIoT Processing, IIoT Communication, Collaborative Platform and Product Lifecycle Management, Role of AI for IIoT.

Unit III Data Monitoring & Control (09 Hours)

IoT Gate way, IoT Edge Systems and Its Programming, Cloud computing, Big Data Analytics and Software Defined Networks, Real Time Dashboard for Data Monitoring, Data Analytics and Predictive Maintenance with IIoT technology, Data Security.

Unit IV Industrial IoT Protocols (09 Hours)

OPC UA, MQTT, CoAP, LoRA, LoRaWAN, Modbus, Zigbee, LWM2M,HTTP, Wi-Fi, Bluetooth & BLE, Cellular, Ethernet, TLS/SSL, DTLS, Case studies of IIoT.

.

Unit V - Building Automation Systems(09 Hours)

Intelligent buildings, Its architecture and structure, Evaluation of ntelligence buildings. Facilities management vs Intelligent buildings, Lifecycle of building. BAS System Hierarchy-Field level components, Direct Digital Control(DDC),Supervisory Controller, Server, Operator Workstation(OWS),Different system in BAS which includes HVAC, Security, fire, lighting systems, importance of each system in BAS,BAS communication protocol and addressing concepts-BACnet and LON.

.

Learning Resources

Text Books:

1. Olivier Hersent, David Boswarthick, Omar Elloumi "The Internet of Things: Key Applications andProtocols", ISBN: 978-1-119-99435-0, 2nd Edition, Willy Publications.
2. Alasdair Gilchrist "Industry 4.0: The Industrial Internet of Things", Apress Publications.
3. Sabina Jeschke, Christian Brecher,Houbing Song, Danda B. Rawat "Industrial Internet of Things:Cyber manufacturing Systems" by Springer Publication.

Reference Books

1. 1.Hakima Chaouchi, "The Internet of Things Connecting Objects to the Web" ISBN: 978-1- 84821-140-7, Willy Publications
2. Sam Goundar, J. Avanija, Gurram Sunitha "Innovations in the Industrial Internet of Things (IIoT) andSmart Factory (Advances in Computer and Electrical Engineering)
3. Dr. Ovidiu Vermesan, Dr. Peter Friess, "Internet of Things: Converging Technologies for SmartEnvironments and Integrated Ecosystems", River Publishers.

Savitribai Phule Pune University **Third Year - Computer Science & Design (2024 Pattern)**		
PEC362ACSD - Natural Language Processing		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Course Objectives: The course aims to:

1. Introduce students to the fundamental concepts of natural language processing and linguistic-structures.
2. Enable students to apply text preprocessing techniques, corpus handling methods, and feature-extraction approaches.
3. Facilitate understanding of statistical models and parsing techniques used in NLP tasks.
4. Help students use NLP models to design simple text-processing applications.
5. Enable students to understand modern approaches in neural NLP, including deep learning and-transformer architectures.

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

- CO1: Examine NLP fundamentals, levels of linguistic analysis, and grammatical formalisms.
- CO2: Utilize text preprocessing, corpus handling, and feature extraction techniques.
- CO3: Interpret statistical NLP techniques such as N-grams, smoothing, and probabilistic parsing.
- CO4: Demonstrate NLP tasks such as POS tagging, NER, and sentiment analysis.
- CO5: Analyze modern NLP concepts, including attention mechanisms, encoder–decoder models,and BERT architecture.

Course Contents

Unit I - Linguistic Foundations and Formal Language Models (09 Hours)

Introduction to Natural Language Processing: History of NLP, Generic NLP System, Levels of NLP, Knowledge in Language Processing, Ambiguity in Natural Language, Stages, Challenges, and

Applications of NLP Formal Language Concepts: Alphabets, strings, languages, regular expressions, finite automata (DFA/NFA), and limitations of regular languages for natural language. Context-Free Grammars : CFG components, Derivations, parse trees, Ambiguity Linguistic Levels: Morphology (roots, affixes, morphological parsing), Finite-State Transducers (FSTs) for morphology, Syntax, semantics, pragmatics, Edit Distance and Spelling Correction: Minimum Edit Distance (MED), Levenshtein distance, Noisy channel model Approaches to NLP: Rule-based, Data-based, and Knowledge-based Approaches. Python-based NLP libraries: Natural Language Toolkit (NLTK), spaCy, TextBlob, and use cases.

Case Study: Design a simple rule-based system to validate and analyze basic English sentences such as “The cat eats food” using regular expressions and Context-Free Grammar (CFG). The system should preprocess the text, define grammar rules for sentence structure (Subject–Verb–Object), generate a parse tree for valid sentences, and explain why regular expressions alone cannot fully capture nested or hierarchical sentence structures.

Unit II - Text Representation & Feature Engineering (09 Hours)

Corpus: Types of corpora, Annotation and formats, POS and NER annotation schemes, Annotation tools Text Preprocessing: Tokenization, Sentence segmentation, Normalization, true casing, Stemming

and lemmatization, Text Cleaning, stemming (Porter Stemmer algorithm), Stopword Removal, Handling OOV (Out-of-vocabulary) Words: Unknown tokens, subword techniques, and challenges with

NLP systems. Word Representation: One-hot, distributional semantics, co-occurrence matrices, and information extraction. Sparse vs Dense Representations: Limitations of sparse vectors (high dimensionality, sparsity)

and advantages of dense embeddings, Lexical Resources: WordNet, synsets, semantic relations. Feature Engineering: Concept of feature engineering, types of features, and K-gram models:

Character-level and word-level k-grams and their applications in text representation, spelling correction, and information retrieval. Feature Extraction: Bag of Words, Document-Term Matrix, TF-IDF; Feature selection: Chi-square, Mutual Information Word embeddings: Word2Vec (CBOW, Skip-gram), GloVe, Embedding intuition

Case Study: Sentiment Analysis of Product Reviews, Spam Email Detection System

Unit III - Statistical NLP Models and Language Modeling (09 Hours)

Probability Basics: Conditional Probability concept and applications in NLP; Maximum Likelihood Estimation (MLE)—parameter estimation techniques for languages and statistical models. N-gram Language Models: Unigram, bigram, trigram, Training, perplexity, Backoff and interpolation Classification Models: Naive Bayes for text, Logistic Regression, Evaluation for text classification Tagging Models: Hidden Markov Models (HMMs), Viterbi algorithm, HMM-based POS tagging Parsing with Probabilities: PCFG, Probabilistic parsing, Top-down and bottom-up parsing

Case Study: Develop and analyze a Part-of-Speech (POS) tagging system using a probabilistic sequence model.

Unit IV - NLP Tasks (09 Hours)

Sequence Labeling Tasks: POS Tagging (HMM, Brill tagger), Named Entity Recognition (rulebased + statistical view) Parsing: Dependency parsing (transition-based and graph-based concepts), constituency vs. dependency, Parsing errors and evaluation Evaluation Metrics: Confusion matrix, Precision, Recall, F1, evaluation for tagging and parsing, token-level accuracy, entity-level F1 (NER), Parse accuracy, Attachment score (basic idea), BLEU, ROUGE. Core NLP Applications: Sentiment analysis, Text classification pipeline, document similarity, Recommendation System, Information retrieval: Vector Space Model, Information Extraction using sequence labelling, sentiment analysis,Word Sense Disambiguation (WSD): concept, Lesk algorithm, and applications. Speech Processing: Mel-Frequency Cepstral Coefficients (MFCC), Automatic Speech Recognition (ASR) pipeline Model Deployment: APIbased NLP Services, Introduction to Model Deployment and Inference, REST APIs, Request–Response Architecture, Overview of Hugging Face Inference APIs, Workflow of NLP Deployment, Introduction to Cloud Platforms, Basic Concepts of Containerization, and Performance Monitoring with Ethical Considerations.

Case Study: BERT-based text classification for customer support queries

Unit V - Modern Neural NLP and Transformers (09 Hours)

Neural NLP Overview: Motivation, limits of statistical models, Dense vector representations Attention Mechanism: Intuition and motivation, Self-attention, Scaled dot-product attention, Advantages

over traditional models 19 Encoder–Decoder Models: Sequence-to-sequence architecture, applications, machine translation methods (rule-based machine translation (RBMT), Statistical Machine Translation (SMT), Neural Machine

Translation (NMT)), Text summarization Transformer Architecture: Multi-head attention, Positional embeddings, Feedforward layers, Advantages over RNN Contextual Embeddings: BERT architecture, Token embeddings, segment embeddings. Downstream tasks: QA, text classification Modern Applications: Chatbots, Sentiment analysis, Document classification, Retrieval-Augmented Generation (RAG)

Case Study: AI-Powered Academic Chatbot System. Retrieval Augmented Question Answering System. Legal Document Summarization. Voice enabled regional language assistant.

Learning Resources

Text Books:

1. Jurafsky, D., & Martin, J. H., Speech and Language Processing: An Introduction to Natural LanguageProcessing, Computational Linguistics, and Speech Recognition, 3rd ed., Pearson / Online,2023–Present.

2. Manning, C. D., & Schütze, H., Foundations of Statistical Natural Language Processing, MITPress, 1999.

Reference Books

1. T V Geetha, "Understanding Natural Language Processing – Machine Learning and Deep LearningPerspectives", Pearson, 2024.

2. Steven Bird,Ewan Klein and Edword Loper," NLP with Python: Analyzing text with the Natural-Language Toolkit", O'Reilly Media,Inc

3. Nitin Indurkhya and Fred J. Damerau, "Handbook of Natural Language Processing", 2nd ed. CRCpress.

MOOC / NPTEL/YouTube Links

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

2. https://onlinecourses.nptel.ac.in/noc26_cs45/preview

E-Books

1. Yoav Goldberg. A primer on neural network models for natural language processing, 2015. URL http://u.cs.biu.ac.il/˜yogo/nnlp.pdf

Savitribai Phule Pune University **Third Year - Computer Science & Design (2024 Pattern)**		
PEC 362B CSD - - High Performance Computing		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Course Objectives: The course aims to:

1. To understand the fundamentals of High Performance Computing.
2. .To learn parallel computing concepts and performance evaluation techniques.
3. To apply basic parallel programming models.
4. To understand GPU and accelerator-based computing.
5. To explore cloud-based HPC for AI applications.

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

- CO1 : Explain fundamental HPC concepts and performance metrics.
- CO2 : Apply parallel programming using shared and distributed memory models.
- CO3 : Describe GPU and accelerator-based computing for AI workloads.
- CO4 : Analyze HPC solutions for AI and data-intensive applications.
- CO5 : Evaluate cloud-based HPC systems for scalable computing.

Course Contents

Unit I - Fundamentals of Modern HPC (09 Hours)

Introduction to HPC: Evolution from sequential to parallel computing, need for HPC in modern workloads, limitations of sequential systems. Parallel vs Distributed Computing. Shared vs Distributed Memory. Basics of Parallel Computing: Concurrency vs Parallelism, Data Parallelism, Task Parallelism, Flynn's Classification (SISD, SIMD, MIMD). Performance Metrics: Execution time, Speedup, Efficiency, Throughput, Scalability (Strong and Weak Scaling). Performance Laws: Amdahl's Law and Gustafson's Law. Applications of HPC in AI, ML, Big Data and Scientific Computing.

Case study: Speedup Analysis of Deep Learning Model Training using Multi-core Processors

Unit II - Parallel Programming Models 09 Hours

Introduction to Parallel Programming and "Thinking Parallel". Shared Memory Programming using OpenMP (parallel for, reduction, tasks – basic concepts). Distributed Memory Programming using MPI (send, receive, broadcast – basic communication). Comparison of OpenMP and MPI. Introduction to Hybrid Programming. Synchronization concepts: barriers and basic race conditions.

Case Study: Parallel Matrix Multiplication using OpenMP vs MPI.

Unit III Accelerators and GPU Computing (09 Hours)

Need for hardware accelerators in modern HPC and limitations of CPU-centric systems for AI workloads. Overview of accelerators such as GPUs, FPGAs and TPUs. Fundamentals of GPU architecture: streaming multiprocessors, thread hierarchy (threads, blocks, grids), SIMT execution model, and GPU memory hierarchy. Basic GPU execution workflow. Introduction to CUDA programming model: kernel launch, host-device interaction, and basic memory management concepts. Role of GPUs in AI and deep

learning: matrix operations, training and inference acceleration. Overview of optimized libraries such as cuBLAS and cuDNN. Basic performance considerations: memory bandwidth, latency, and idea of multi-GPU systems with CPU–GPU coordination.

Case Study: Performance measurement, bottleneck identification, optimisation (kernel fusion,Tensor Cores, batch size tuning), Comparison with CPU-only baseline & with FPGA-based inference (optional)

Unit IV -HPC for AI and Data-Intensive Applications (09 Hours)

Introduction to HPC for AI : Why AI needs HPC , computational challenges in deep learning , Key aspect of HPC in AI, Architecture, Overview of AI workloads: training vs.inference, Data Parallelism vs Model Parallelism, Examples of AI framworks and their HPC integration Distributed Deep Learning, Application of HPC for AI. Introduction to Data-Intensive Applications : Big Data Architectures, Distributed File Systems, Data Locality, MapReduce Paradigm, Spark-based Distributed Processing. Performance challenges in large-scale AI systems: communication overhead and scalability issues.

Case Study: Distributed Training of a Deep Learning Model using Data Parallelism.

Unit V - Cloud and Modern HPC Systems (09 Hours)

Introduction to Cloud-based HPC and virtual clusters. On-premise HPC vs Cloud HPC. Infrastructure as a Service (IaaS) for HPC workloads. Containers (Docker – basic concept) for portable HPC applications. Introduction to Kubernetes for workload orchestration. Elastic scaling and auto-scaling in cloud environments. Cost-aware computing and use of spot instances. Overview of Energy efficiency and green computing in HPC. Future trends: Exascale computing, AI-driven HPC optimization.

Case Study: Deployment of AI Model Training on Cloud Cluster – Cost and Scalability Analysis

Learning Resources

Text Books:

1. Grama, Ananth, Gupta, Anshul, Karypis, George, and Kumar, Vipin, Introduction to Parallel Computing,Pearson Education, 2nd Edition, 2003, ISBN: 978-0201648652.
2. Quinn, Michael J., Parallel Programming in C with MPI and OpenMP, McGraw-Hill, 2004, ISBN:978-0072822564.
3. Hwang, Kai, Dongarra, Jack, and Fox, Geoffrey, Distributed and Cloud Computing: From ParallelProcessing to the Internet of Things, Morgan Kaufmann, 1st Edition, 2012, ISBN: 978-0123858801.

Reference Books

1. Kirk, David B., and Hwu, Wen-mei W., Programming Massively Parallel Processors: A Hands-onApproach, Morgan Kaufmann, 3rd Edition, 2016, ISBN: 978-0128119860.
2. Rauber, Thomas and Rünger, Gudula, Parallel Programming for Multicore and Cluster Systems,Springer, 2nd Edition, 2013, ISBN: 978-3642378010.

MOOC / NPTEL/YouTube Links

1. High Performance Computing by Prof. S. Gopalakrishnan, IIT Madras https://nptel.ac.in/courses/10610
2. Parallel Computer Architecture and Programming – IIT Kanpur (NPTEL) https://nptel.ac.in
3. Introduction to High Performance Computing https://www.youtube.com/watch?v=KsL_uwa0ekY

E-Books

1. https://dl.acm.org/doi/pdf/10.5555/3455710 NVIDIA CUDA Programming Guide (Official Documentation)

2. https://docs.nvidia.com/cuda/ OpenMP API Specification (Official Documentation)
3. https://www.openmp.org/specifications/ MPI Standard Documentation
4. https://www.mpi-forum.org/docs/

Savitribai Phule Pune University **Third Year - Computer Science & Design (2024 Pattern)**		
PEC362C CSD: Business Intelligence		
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 : Elective III Lab
Course Objectives: The course aims to:

1. Understand the concepts, architecture, and applications of Business Intelligence (BI).
2. Learn techniques for data warehousing, ETL, OLAP, and data integration.
3. Explore data mining and analytical methods used in BI systems.
4. Understand BI tools, dashboards, data visualization, and reporting.
5. . Apply BI concepts for decision making in business and enterprise environments

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

CO1 Understand the concepts, architecture, and significance of Business Intelligence systems in organizational decision-making.

CO2 Design and explain data warehouse models, ETL processes, and multidimensional schemas for business data analysis.

CO3 Apply OLAP operations and business analytics techniques to analyze and interpret organizational data.

CO4 Use data mining and predictive analysis methods to discover patterns and generate business insights.

CO5 Develop and interpret BI reports, dashboards, and visualizations using appropriate BI tools for decision support.

Course Contents

Unit I -Introduction to Business Intelligence (09 Hours)

- Introduction to Business Intelligence
- Evolution of BI and Decision Support Systems
- Need and importance of BI in organizations
- Data, Information, Knowledge and Wisdom
- Types of managerial decisions: strategic, tactical, operational
- BI applications in business domains: finance, marketing, healthcare, education, manufacturing, e-commerce
- BI process and components
- BI architecture and framework
- Role of BI in organizational decision-making
- Challenges in implementing BI systems

Unit II Data Warehousing and ETL(09 Hours)

- Introduction to Data Warehousing
- OLTP vs OLAP systems
- Characteristics of Data Warehouse
- Data Warehouse architecture
- Data marts and enterprise warehouse
- Metadata and its importance

- Fact tables and dimension tables
- Star schema, Snowflake schema, Fact constellation schema
- ETL process: Extraction, Transformation, Loading
- Data cleansing, data integration and data quality
- Warehouse implementation issues
- Overview of Data Warehouse design methodologies

Unit III OLAP and Data Analytics for BI (09 Hours)

- Introduction to Online Analytical Processing (OLAP)
- OLAP operations:
o Roll-up
o Drill-down
o Slice
o Dice
o Pivot (Rotate)
- Multidimensional data model
- OLAP cube and aggregation
- MOLAP, ROLAP, HOLAP
- Querying and analysis using OLAP
- Key Performance Indicators (KPIs)
- Business metrics and performance measurement
- Introduction to Business Analytics
- Descriptive, Diagnostic, Predictive and Prescriptive Analytics

Unit IV Data Mining and Predictive Intelligence (09 Hours)

- Introduction to Data Mining in BI
- Relationship between BI, Data Mining and Machine Learning
- Data preprocessing and feature selection
- Data mining tasks:
o Classification
o Clustering
o Association rule mining
o Regression
o Anomaly detection
- Decision trees, Naïve Bayes, K-means (basic concepts)
- Market basket analysis
- Trend and pattern analysis
- Forecasting and predictive modeling
- Applications of predictive analytics in business

Unit V BI Tools, Dashboards, Visualization and Applications(09 Hours)

- Introduction to BI tools and platforms
- Reporting systems and business reports
- Interactive dashboards and scorecards
- Data visualization principles
- Charts, graphs, heat maps, KPI indicators
- Dashboard design best practices
- Self-service BI
- Introduction to tools such as:
o Microsoft Power BI
o Tableau

- Google Data Studio / Looker Studio
- Excel for BI

- Real-time BI and cloud-based BI
- Case studies and applications of BI in industry
- Ethical issues, privacy and security in BI

Learning Resources

Text Books:

1. Efraim Turban, Ramesh Sharda, Dursun Delen – Business Intelligence, Analytics, and Data Science: A Managerial Perspective
2. Carlo Vercellis – Business Intelligence: Data Mining and Optimization for Decision Making
3. Ralph Kimball & Margy Ross – The Data Warehouse Toolkit
4. Jiawei Han, Micheline Kamber, Jian Pei – Data Mining: Concepts and Techniques

Reference Books

1. Larissa T. Moss, Shaku Atre – Business Intelligence Roadmap
2. Cindi Howson – Successful Business Intelligence
3. David Loshin – Business Intelligence: The Savvy Manager's Guide

Savitribai Phule Pune University **Third Year - Computer Science & Design (2024 Pattern)**		
PEC362D CSD : Generative AI		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 03 Hours/Week	03	CCE : 30 Marks End-Semester: 70 Marks

Course Objectives: The course aims to:

1. Understand the fundamentals of Artificial Intelligence and Machine Learning in the context of generative models.
2. Learn core generative techniques such as Generative Adversarial Networks, Variational Autoencoders, and Transformer Architecture.
3. .Develop skills to build applications using Generative AI tools .
4. Understand ethical issues and real-world implications.
5. Apply generative models to text, image, and multimedia tasks.

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

- CO1: Understand core concepts of Generative AI models
- CO2: Implement models like GANs, VAEs, and transformers
- CO3: Apply Natural Language Processing techniques for text generation
- CO4: Build real-world applications (chatbots, image generators)
- CO5: Analyze ethical concerns and limitations of Generative AI

Course Contents

Unit I - Introduction to Generative AI (09 Hours)

Introduction: Overview of AI and Generative AI, Difference between discriminative and generative models, Applications: text, image, audio, video.

Introduction to Large Language Models- GPT-4, BERT, LLaMA, PaLM, Claude.

Basics of data and training- Types of Data, Data Preprocessing, Training Process, Training Techniques- Supervised Learning, Unsupervised Learning , Self-supervised Learning, Challenges. Reinforcement learning.

Case Study: Companies use ChatGPT for automated content creation.

Unit II Deep Learning Foundations(09 Hours)

Introduction : neural networks, basics of CNNs and RNNs. Activation functions: Sigmoid Function Tanh (Hyperbolic Tangent) , ReLU (Rectified Linear Unit) , Softmax Function , ELU (Exponential Linear Unit) , Swish (Modern Activation).

Loss functions : Regression Loss Functions, Mean Squared Error (MSE), Mean Absolute Error (MAE), Huber Loss. Optimization techniques (Gradient Descent, Adam)

Case Study: Tesla uses CNNs for object detection in self-driving cars.

Unit III Generative Models (09 Hours)

Generative Adversarial Networks (GANs): Fundamentals of GANs ,Architecture, Types of GANs , Training Techniques , Applications of GANs , Training challenges.Variational Autoencoders (VAEs) : Latent space representation, Diffusion Models (basic concept), Comparison of generative techniques

Case Study: Generates synthetic MRI/CT scans for training models.

Unit IV Transformer Models and NLP(09 Hours)

Working of Transformer Architecture ,Attention mechanism ,Applications in Natural Language Processing ,Text generation and Language Modeling : N-gram models,Neural language models ,Transformer-based models (e.g., GPT, BERT), Neural Text Generation Techniques: Recurrent Neural Networks (RNNs),LSTM / GRU models ,Transformer architecture (encoder-decoder) . Controlled Text Generation:,Style transfer (formal/informal, sentiment control) ,Topic-guided generation .Conditional text generation, Prompt engineering basics

Case Study: Smart Email Reply System (Gmail Smart Compose)

Unit V Applications, Tools & Ethics (09 Hours)

Applications : Building simple Generative AI applications .Tools: TensorFlow, PyTorch, APIs: OpenAI API , Google Cloud NLP API, Hugging Face API.

Ethical issues: Types of Bias, Sources of Bias, Impacts of Bias, Bias Detection & Measurement, Bias Mitigation Techniques. deepfakes, copyright .Future trends in Generative AI: Multimodal Generative AI, Agentic AI, Rise of Multi-Agent Systems, Domain-Specific (Specialized) AI Models.

Case Study: chatbots, image generators

Learning Resources

Text Books:

1. Generative AI: Foundations & Concepts: Authors: Usman Qamar, Muhammad Summair Raza, Publisher: Springer (2026) .
2. Introduction to Generative AI: Authors: Numa Dhamani, Maggie Engler.
3. Generative Deep Learning: Author: David Foster

Reference Books

1. Hands-On Machine Learning with Scikit-Learn Keras and TensorFlow , Author: Aurélien Géron
2. Artificial Intelligence: A Modern Approach Authors: Stuart Russell, Peter Norvig

Savitribai Phule Pune University **Third Year - Computer Science & Design (2024 Pattern)**		
PEC363CSD - Elective III Laboratory		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	01	Term Work: 25 Marks Practical : 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 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 - Natural Language Processing Laboratory

Suggested List of Laboratory Experiments/Assignments

Group A (Any 4)

1. NLP Pipeline and Linguistic Analysis: Develop a Python-based NLP system to apply and analyze the basic stages of Natural Language Processing on a given text corpus. The system should:
   - Perform preprocessing:
   - Tokenization
   - Stop-word removal
   - Lemmatization/Stemming
   - POS tagging

- Identify morphological components (root, suffix)
- Interpret ambiguity in sentences and classify type (lexical/syntactic).
- Compare outputs using two NLP libraries (NLTK and spaCy).

2. Regular Expressions and Finite Automata: Design and implement a system using Regular Expressions to apply and evaluate pattern matching techniques for structured text processing. The system should Construct regular expressions to validate: Email IDs, Phone numbers, Dates, Extract patterns, URLs,

   (a) Hashtags Design and analyze a Finite Automaton (DFA/NFA) for Binary strings ending with "01"

   (b) Evaluate limitations of regular languages in representing nested sentence structures.

3. Edit Distance and Spelling Correction: Develop a system to apply and analyze string similarity techniques for spelling correction. The system should:

   - Implement Minimum Edit Distance (MED) / Levenshtein distance.
   - Compute and compare distances between Misspelled and correct words
   - Apply the Noisy Channel Model to suggest the most probable correction.
   - Evaluate performance of the system on a dataset.

4. Text Preprocessing and OOV Handling: Design and implement a preprocessing pipeline for textual data. The system should perform:

   - Tokenization
   - Sentence segmentation
   - Stopword removal
   - Stemming and Lemmatization
   - Text normalization and true casing
   - Identify and handle Out-of-Vocabulary (OOV) words using:
   - Unknown token replacement
   - Analyze and compare text before and after preprocessing.
   - Dataset: IMDB Movie Reviews Dataset , Twitter dataset (noisy text)

5. Feature Extraction and Text Representation: Develop a system to convert text into numerical representations. The system should

   (a) Implement: One-hot encoding , Bag of Words (BoW) , Document-Term Matrix , TF-IDF

   (b) Compare and analyze: Sparse vs dense representations , Feature dimensionality and sparsity ,

   (c) Interpret and visualize feature vectors. Dataset: Custom dataset (10–20 documents) , 20 Newsgroups dataset (subset)

Group B (Any 4)

1. Text Classification using Naïve Bayes and Logistic Regression: Develop a text classification system to classify documents into predefined categories. The system should:

   (a) Preprocess the text data (tokenization, stopword removal).

   (b) Convert text into numerical features using TF-IDF.

(c) Implement: Naïve Bayes classifier , Logistic Regression model

(d) Evaluate model performance using: Accuracy , Precision, Recall, F1-score ,Confusion Matrix

(e) Compare the performance of both models and justify the results. (Dataset: IMDB Movie Reviews Dataset , SMS Spam Dataset)

2. N-gram Language Model and Perplexity Analysis: Develop a language model to predict word sequences using N-gram techniques. The system should:

   (a) Build : Unigram , Bigram , Trigram models

   (b) Estimate probabilities using Maximum Likelihood Estimation (MLE).

   (c) Apply smoothing techniques (Laplace or Good-Turing).

   (d) Compute perplexity for each model.

   (e) Compare model performance and analyze the effect of smoothing. (Dataset: Brown Corpus , Gutenberg Corpus)

3. POS Tagging using Hidden Markov Model (HMM): Develop a Part-of-Speech (POS) tagging system using Hidden Markov Models. The system should:

   (a) Model POS tagging using HMM (states, observations).

   (b) Estimate transition and emission probabilities.

   (c) Implement the Viterbi algorithm to find the most probable tag sequence.

   (d) Evaluate tagging accuracy on test sentences.

   (e) Analyze errors and discuss limitations of HMM. (Dataset: Penn Treebank Dataset , NLTK tagged corpus)

4. Named Entity Recognition (NER) and Evaluation: Develop a Named Entity Recognition (NER) system to identify entities such as persons, locations, and organizations. The system should:

   (a) Preprocess input text data.

   (b) Apply NER using:

      i. Rule-based approach (pattern matching)

      ii. Pre-trained statistical model (spaCy)

   (c) Extract and classify named entities.

   (d) Evaluate model performance using: Precision , Recall , F1-score

   (e) Compare rule-based and statistical approaches and justify the results. (Dataset:CoNLL-2003 dataset , Custom news articles)

5. Dependency Parsing and Sentence Analysis: Develop a system to analyze grammatical relationships in sentences using dependency parsing. The system should:

   (a) Parse sentences using a dependency parser.

   (b) Identify syntactic relationships (subject, object, modifiers).

   (c) Compare: Dependency parsing ,Constituency parsing (conceptual comparison)

   (d) Visualize parse trees/graphs.

   (e) Analyze parsing errors and discuss limitations. (Dataset:Custom sentence dataset)

6. Sentiment Analysis and Text Classification Pipeline: Develop a sentiment analysis system using a complete NLP pipeline. The system should:

(a) Perform preprocessing and feature extraction (TF-IDF).

(b) Implement a classification model (Logistic Regression / Naïve Bayes).

(c) Classify text into positive/negative sentiments.

(d) Evaluate performance using: Accuracy , Precision, Recall, F1-score , Confusion Matrix

(e) Analyze model performance and suggest improvements.(Dataset:IMDB Movie Reviews Dataset , Twitter Sentiment Dataset)

7. Text Classification using BERT : Develop a text classification system using a pre-trained BERT model. The system should:

   (a) Load a pre-trained BERT model using a transformer library.

   (b) Preprocess input text using tokenization and encoding.

   (c) Fine-tune BERT for classification (e.g., sentiment analysis).

   (d) Evaluate model performance using: Accuracy , Precision, Recall, F1-score,

   (e) Dataset:(IMDB Movie Reviews Dataset , SST-2 Dataset)

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

1. **1**. Text Classification using Naïve Bayes and Logistic Regression: Develop a text classification system to classify documents into predefined categories. The system should:

   (a) Preprocess the text data (tokenization, stopword removal).

   (b) Convert text into numerical features using TF-IDF.

   (c) Implement: Naïve Bayes classifier , Logistic Regression model

   (d) Evaluate model performance using: Accuracy , Precision, Recall, F1-score ,Confusion Matrix

   (e) Compare the performance of both models and justify the results. (Dataset: IMDB Movie Reviews Dataset , SMS Spam Dataset)

2. N-gram Language Model and Perplexity Analysis: Develop a language model to predict word sequences using N-gram techniques. The system should:

   (a) Build : Unigram , Bigram , Trigram models

   (b) Estimate probabilities using Maximum Likelihood Estimation (MLE).

   (c) Apply smoothing techniques (Laplace or Good-Turing).

   (d) Compute perplexity for each model.

   (e) Compare model performance and analyze the effect of smoothing. (Dataset: Brown Corpus , Gutenberg Corpus)

3. POS Tagging using Hidden Markov Model (HMM): Develop a Part-of-Speech (POS) tagging system using Hidden Markov Models. The system should:

   (a) Model POS tagging using HMM (states, observations).

   (b) Estimate transition and emission probabilities.

   (c) Implement the Viterbi algorithm to find the most probable tag sequence.

   (d) Evaluate tagging accuracy on test sentences.

   (e) Analyze errors and discuss limitations of HMM. (Dataset: Penn Treebank Dataset , NLTK tagged corpus)

4. Named Entity Recognition (NER) and Evaluation: Develop a Named Entity Recognition (NER) system to identify entities such as persons, locations, and organizations. The system should:
   (a) Preprocess input text data.
   (b) Apply NER using:
      i. Rule-based approach (pattern matching)
      ii. Pre-trained statistical model (spaCy)
   (c) Extract and classify named entities.
   (d) Evaluate model performance using: Precision , Recall , F1-score
   (e) Compare rule-based and statistical approaches and justify the results. (Dataset:CoNLL-2003 dataset , Custom news articles)
5. Dependency Parsing and Sentence Analysis: Develop a system to analyze grammatical relationships in sentences using dependency parsing. The system should:
   (a) Parse sentences using a dependency parser.
   (b) Identify syntactic relationships (subject, object, modifiers).
   (c) Compare: Dependency parsing ,Constituency parsing (conceptual comparison)
   (d) Visualize parse trees/graphs.
   (e) Analyze parsing errors and discuss limitations. (Dataset:Custom sentence dataset)
6. Sentiment Analysis and Text Classification Pipeline: Develop a sentiment analysis system using a complete NLP pipeline. The system should:
   (a) Perform preprocessing and feature extraction (TF-IDF).
   (b) Implement a classification model (Logistic Regression / Naïve Bayes).
   (c) Classify text into positive/negative sentiments.
   (d) Evaluate performance using: Accuracy , Precision, Recall, F1-score , Confusion Matrix
   (e) Analyze model performance and suggest improvements.(Dataset:IMDB Movie Reviews Dataset , Twitter Sentiment Dataset)

Guidelines for Laboratory Conduction - High Performance Computing

Guidelines for Laboratory Conduction - High Performance Computing

1. Laboratory sessions should be conducted in alignment with the approved HPC theory syllabus.
2. Required software tools (OpenMP, MPI, CUDA, Spark, etc.) must be properly installed and tested before conducting practicals.
3. Each experiment should include brief explanation of theory followed by hands-on implementation.
4. Students should compare sequential and parallel versions of programs wherever applicable to analyze performance.
5. Emphasis should be given to performance metrics such as execution time, speedup, and efficiency.
6. Continuous assessment should be carried out based on implementation, understanding, and timely submission of assignments

List of Assignment

1. Install and configure OpenMP/MPI environment. Write a simple C/Python program to measure execution time of a sequential program and analyze performance.
2. Implement a CPU-based parallel program (e.g., array sum or matrix addition using OpenMP). Calculate speedup and efficiency for different number of threads.
3. Implement parallel matrix multiplication or sorting using OpenMP directives. Compare sequential vs parallel execution time.
4. Implement parallel matrix addition or multiplication using MPI (send, receive, broadcast). Analyze communication overhead.
5. Implement a computational problem (e.g., sorting, numerical computation, or array processing) using both OpenMP and MPI. Analyze and compare execution time, speedup, and scalability of the two programming models.
6. Write a simple CUDA program (vector addition or matrix addition). Compare CPU vs GPU execution time.
7. Implement a simple deep learning operation (matrix multiplication or convolution using PyTorch/TensorFlow with GPU). Measure training time with and without GPU.
8. Implement a basic MapReduce task (word count or log analysis) using Apache Spark.Analyze performance for different data sizes.
9. Mini Project: HPC-based AI or Data-Intensive Application : Develop and implement a small project using HPC concepts such as parallel programming, GPU computing, or distributed processing.

Guidelines for Laboratory Conduction - Business Intelligence

(Any 6 and Mini Project Compulsory)

1. Study of BI architecture and applications
2. Design of star schema and snowflake schema
3. ETL process using sample datasets
4. OLAP operations on multidimensional data
5. Data cleaning and preprocessing
6. Classification and clustering using sample business data
7. Association rule mining for market basket analysis
8. Dashboard creation using Power BI / Tableau / Excel
9. KPI and report generation
10. Mini project on BI application using real-world dataset

Mini Project

1. Student performance analytics dashboard
2. Sales and revenue analysis system
3. Hospital patient data intelligence dashboard
4. E-commerce customer behavior analysis
5. Social media sentiment-based BI
6. Employee attrition analytics
7. Retail market basket analysis
8. Financial fraud detection using BI analytics

Guidelines for Laboratory Conduction - Generative AI

(Any 6 and Mini Project Compulsory)

1. Study and implement the basic concepts of Generative AI using tools like ChatGPT for generating text, code, and creative content.
2. Study and implement an AI-based chatbot like ChatGPT in order to understand its ability to generate human-like responses, answer queries, and assist in various tasks such as learning, problem-solving, and content creation.
3. Implement the application which can automatically summarize lengthy text into concise and meaningful content while retaining the key information and main ideas by using Generative AI tool like ChatGPT.
4. Utilize a Generative AI tool like ChatGPT to automatically generate computer programs or code snippets based on user prompts, and to understand its effectiveness in assisting coding and problem-solving tasks.
5. Create images from textual descriptions, and to understand how AI can generate visual content based on user prompts using a Generative AI tool like DALL·E.
6. Study and implement Natural Language Processing (NLP) techniques used in Generative AI for tasks such as text generation, sentiment analysis, and summarization using modern transformer-based models.
7. Mini Project

Savitribai Phule Pune University **Third Year - Computer Science & Design (2024 Pattern)**		
MDM371 CSD - Robotics and Automation		
Teaching Scheme	**Credits**	**Examination Scheme**
Tutorial : 01 Hour/Week **Practical**: 2 Hours/Week	01 01	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 to:

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

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

Case Study:

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

Guidelines for Laboratory Conduction

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 automation
6. Home automation using sensors and actuators

Savitribai Phule Pune University **Third Year - Computer Science & Design (2024 Pattern)**		
VSE372CSD - Solar Technology and Maintenance		
Teaching Scheme	**Credits**	**Examination Scheme**
Practical : 02 Hours/Week	01	Term Work: 50 Marks

Prerequisites: Basic knowledge of Physics (especially topics like electricity, magnetism semiconductors,light/energy concepts, Basic Electrical Engineering or Basic Electronics, Engineering Mechanics. Heat and
energy concepts

Course Objectives: The course aims to:

1. Apply Safely install, wire, and commission basic solar PV systems while measuring key performance parameters.
2. Analyze Break down the impact of environmental and operational factors on solar system efficiency and diagnose common faults.
3. Evaluate Judge the effectiveness of maintenance and troubleshooting procedures for solar PV components and systems.
4. Create Develop simple practical solutions or documentation for improving solar system performance in mini-projects.

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

- CO1: Apply safe installation, wiring, commissioning, and performance measurement of basic solar PV systems.
- CO2: Analyze the impact of environmental/operational factors on solar PV efficiency and diagnose common faults.
- CO3: Evaluate the effectiveness of maintenance and troubleshooting procedures for solar PV components and systems.
- CO4: Create simple practical solutions or documentation for improving solar system performance via mini-projects.

Practical Assignments

1. Experiment no.1, 2 and 10 are compulsory.
2. Perform any 2 Experiments from 3 to 5 and
3. Perform any 3 Experiments from 6 to 9

1. Measurement of solar irradiance using pyrometer/lux meter at different times/angles. Real-World Assignment: Survey irradiance on your college rooftop for one day. Calculate daily energy generation for a 100W panel and suggest best installation time/angle for maximum output.
2. Plot I-V and P-V characteristics of solar PV module under varying light & temperature. Real-World Assignment: Simulate cloudy/rainy day conditions. Calculate module efficiency and estimate annual energy loss in Pune climate.
3. Survey and Comparative Analysis of Solar PV Installation Systems: Grid-Tied, Hybrid, and Off-Grid Configurations. Real-World Assignment: Survey 2–3 real solar installations (e.g., college rooftop, nearby home/business, or online/virtual

4. Series and parallel connection of PV modules, observe mismatch issues. Real-World Assignment: Design a small array for 12V/24V system (e.g., for laptop charging or lab fan). Calculate total power and suggest fuse/ diode protection for mismatch in a multi-panel rooftop installation.
5. Installation and wiring of standalone solar PV system (PV $\rightarrow$ Charge controller $\rightarrow$ Battery $\rightarrow$ Load/Inverter) Real-World Assignment: Prepare a complete wiring diagram and BOM for a 100W system to power a college water cooler or hostel room. Include safety earthing and cable sizing as per real IEC standards.
6. Preventive maintenance: Cleaning, visual inspection, corrosion/loose connection check. Real-World Assignment: Inspect any existing solar panel in college/hostel. Prepare a 6-month maintenance schedule with cost estimation (dust cleaning, tightening)
7. Grid-Related Maintenance Checks for Grid-Tied Solar PV Systems: Inverter Health, Performance Monitoring, and Fault Diagnosis. Real-World Assignment: Survey a real grid-tied installation, Prepare a maintenance schedule:Monthly inverter check, quarterly visual, annual professional inspection.
8. Mounting structure assembly: Rooftop/ground mount, tilt adjustment, stability check Real-World Assignment: Design a simple mounting frame for windy Pune conditions. Calculate wind load and suggest material/cost for a 5kW residential installation.
9. IoT-Based Real-Time Solar PV System Monitoring and Performance Dashboard.
10. Industrial Visit to Solar Energy Facility in Pune Region: Hands-On Learning of Solar PV System Operations and Maintenance

Test Books:

1. S.P. Sukhatme, Solar Energy
2. C.S. Solanki, Solar Photovoltaics
3. D.P. Kothari et al., Renewable Energy Sources
4. G.D. Rai, Non-Conventional Energy Sources
5. H.P. Garg, Solar Energy Utilization

Reference Books:

1. Solar Photovoltaic Technology and Systems: A Manual for Technicians, Trainers and EngineersAuthor: Chetan Singh Solanki.
2. Solar PV System: Design, Installation, Operation and MaintenanceAuthors: L. Ashok Kumar and K. Mohana Sundaram.
3. Solar Engineering of Thermal Processes, Photovoltaics and Wind (5th Edition) Authors: John A. Duffie, William A. Beckman (updated with Nathan Blair).
4. Principles of Solar Engineering (3rd Edition) Authors: D. Yogi Goswami, Frank Kreith, Jan F. Kreider

NPTEL Course:

1. Solar Photovoltaics: Fundamentals, Technology and Applications: https://onlinecourses.nptel.ac.in/noc
2. SkillCat or Other Free Solar Training (Installation Focus). https://www.skillcatapp.com/solar-installation-training

Savitribai Phule Pune University **Third Year - Computer Science & Design (2024 Pattern)**		
ELC381CSD- Internship/On Job Training		
Teaching Scheme	**Credits**	**Examination Scheme**
Theory : 08 Hours/Week	04	Oral : 50 Marks

Internships are educational and career development opportunities, providing practical experience in a field or discipline. Internships are far more important as the employers are looking for employees who are properly skilled and having awareness about industry environment, practices and culture. Internship is structured, short-term, supervised training often focused around particular tasks or projects with defined time scales. Core objective is to expose technical students to the industrial environment, which cannot be simulated/experienced in the classroom and hence creating competent professionals in the industry and to understand the social, economic and administrative considerations that influence the working environment of industrial organizations. Engineering internships are intended to provide students with an opportunity to apply theoretical knowledge from academics to the realities of the field work/training.

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. Develop professional competency, ethics, communication, and teamwork skills.
4. To encourage self-learning and problem-solving abilities.
5. Encourage innovation, entrepreneurship, and research aptitude.

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

- **CO1:** Apply theoretical knowledge to solve real-world engineering problems.
- **CO2:** Demonstrate technical competency in tools/technologies used in industry.
- **CO3:** Exhibit professional ethics and teamwork.
- **CO4:** Prepare technical reports and deliver effective presentations on industrial training experience.
- **CO5:** Analyze industrial processes and suggest feasible improvements or innovations.

Guidelines

1. Students should opt for a internship/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 duration is NOT considered as summer internship**
4. However student can attend such programs mentioned in above to learn new tools for short duration that would help for solving the problem undertaken in the internship
5. Students should take a challenging task, may be a small portion, and apply the knowledge gained to solve it.

6. Internship can also involve data collection from different sources, including generating experimental data, collection of data from field etc. The data may be analyzed later on.
7. Different central and state government organizations, CSIR labs, premier institutions like IITs and IIMs, DRDO,public sector undertaking organizations, top IT companies may be considered for internships.
8. Student need to submit Synopsis, Permission letter and offer letter to Internship coordinator before proceeding to internship.
9. Internship completion will be considered only after submission of valid documents at the end of internship like Completion certificate, Report and presentation of work done, feedback from industry etc.
10. Student will appear for term work evaluation where he/she will present the work done before mentor(s) at the end of internship.

Suggested Internship Activities

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

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

Programme Coordinator

Dr. Dhananjay Kshirsagar - Member, Board of Studies - Computer Engineering

Team Members for Course Design

Dr. Ravindra S. Tambe	Dr. Vithalrao Vikhe Patil COE,Ahilyanagar
Dr. Jayashri S.Hase	KK Wagh COE , Nasik
Dr. Narayan B. Vikhe	Dr. Vithalrao Vikhe Patil COE,Ahilyanagar
Dr. Salunke Mangesh	Marathwada Mitra Mandal's Institute of Technology,Pune
Prof. Vidya V. Jagtap	Dr. Vithalrao Vikhe Patil COE,Ahilyanagar
Dr. Uttam Patole	Amrutvahini CoE,Sangamner

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

***