Zeal Education Society's

ZEAL COLLEGE OF ENGINEERING & RESEARCH, PUNE – 41

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

NBA Accredited, NAAC Accredited with A+ Grade, ISO 21001:2018

DEPARTMENT OF INFORMATION TECHNOLOGY

Curriculum Structure and Syllabus of

T.Y. B. Tech. – Information Technology

(With effect from - Academic Year 2026 - 27)
(2024 Pattern)

VISION OF THE INSTITUTE

To be a premier institute in technical education by imparting academic excellence, research, social and entrepreneurial attitude.

MISSION OF THE INSTITUTE

- To achieve academic excellence through an innovative teaching and learning process.
- To imbibe the research culture for addressing industry and societal needs.
- To inculcate a social attitude through community engagement initiatives.
- To provide a conducive environment for building entrepreneurial skills.

DEPARTMENT OF INFORMATION TECHNOLOGY

VISION:

To nurture the wisdom of young minds through modern, qualitative and Interdisciplinary research-oriented education to become a successful IT Professional.

MISSION:

M1: Advancing knowledge through fundamental and applied research.
M2: To encourage students for Innovative development and higher studies.
M3: To motivate the crux of learners towards real time solutions.
M4: To prepare skillful engineers for industry to cater best in IT Enabled Services.
M5: To nourish a student's leadership skills by inculcating personal touch and respect in professional relationships.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs):

PEO1: Graduates will develop a strong foundation in mathematics, science, and engineering Principles, enabling them to solve real-world technological problems in the field of Information Technology.
PEO2: Graduates will acquire the knowledge and technical skills necessary to design and Implement innovative IT solutions, fostering adaptability, analytical thinking, and entrepreneurial spirit.
PEO3: Graduates will demonstrate ethical responsibility, effective communication, teamwork and a commitment to continuous learning, enabling them to contribute meaningfully to Society and perform effectively in a global professional environment.

PROGRAM OUTCOMES (POs):

PO1: 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: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development.
PO3: 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.

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.

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.

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.

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws.

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.

PROGRAM SPECIFIC OUTCOMES (PSOs):

PSO1: Apply theoretical and practical knowledge of Information Technology to design, develop, and manage efficient information systems and interdisciplinary applications

PSO2: Analyze real-world problems, identify system requirements, and design appropriate IT infrastructure and solutions, particularly for large-scale and enterprise computing systems.

PSO3: Demonstrate understanding of professional, ethical, legal, and societal responsibilities in IT practices, along with business and security processes.

PSO4: Exhibit effective communication, teamwork, and decision-making skills using modern tools and technologies to perform responsibly in professional environments.

LIST OF ABBREVIATIONS

Abbreviation	Description
BSC	Basic Science Course
ESC	Engineering Science Course
PCC	Program Core Course
PEC	Program Elective Course
MDM	Multidisciplinary Minor
OE	Open Elective - Other than a particular program
VSEC	Vocational and Skill Enhancement Course
AEC	Ability Enhancement Course
ENTR	Entrepreneurship
EC	Economics
MC	Management Courses
IKS	Indian Knowledge System
VEC	Value Education Courses
RM	Research Methodology
CEP	Community Engagement Project
FP	Field Project
PROJ	Project
INT	Internship
OJT	On Job Training
CC	Co-curricular Courses
HSSM	Humanities, Social Science and Management
ELC	Experiential Learning Course
B. Tech	Bachelor of Technology
L	Lecture
P	Practical
T	Tutorial
H	Hours
CR	Credits
CIE	Continuous Internal Evaluation
ETE	End Term Evaluation
TW	Term Work
OR	Oral
PR	Project

DEPARTMENT OF INFORMATION TECHNOLOGY

Third Year B. Tech. – Information Technology: Semester V

Course Code	Course Type	Course Name	Teaching Scheme (Hrs./Week)							Evaluation Scheme					
			L	P	T	H	CR			CIE	ETE	TW	PR	OR	Total
							TH	PR/ Tut	Total						
ITPC508	PCC	Automata Theory	3	-	-	3	3	-	3	40	60	-	-	-	100
ITPC509	PCC	Artificial Intelligence & Machine Learning	3	-	-	3	3	-	3	40	60	-	-	-	100
ITPC509P	PCC	Artificial Intelligence & Machine Learning Laboratory	-	2	-	2	-	1	1	-	-	-	50	-	50
ITPC510	PCC	Design and Analysis of Algorithms	3	-	-	3	3	-	3	40	60	-	-	-	100
ITPC510P	PCC	Design and Analysis of Algorithms Laboratory	-	2	-	2	-	1	1	-	-	-	-	25	25
ITPE501	PEC	Program Elective Course - I *	3	-	-	3	3	-	3	40	60	-	-	-	100
ITPE501P	PEC	Program Elective Course - I * Laboratory	-	2	-	2	-	1	1	-	-	-	-	25	25
ALOE503	OE	Open Elective - III #	1	2	-	3	1	1	2	-	-	50	-	50	100
ITMD503	MDM	Web Technology	1	4	-	5	1	2	3	-	-	25	50	-	75
ITIN504	ELC - INT	Internship - IV	4 Weeks				-	2	2	-	-	25	-	-	25
Total			**14**	**12**	**-**	**26**	**14**	**8**	**22**	**160**	**240**	**100**	**100**	**100**	**700**

*** - Select any one course from the given Program Elective Courses**

Course Code	Course Type	Program Elective-I
ITPE501A	PEC	Computer Networking and Technology
ITPE501B		Data Visualization with Python
ITPE501C		Human-Computer Interaction

 # - Select any one course from the given Open Elective Courses

Course Code	Course Type	Open Elective - III
ALOE503A	OEC	Data Analytics
ALOE503B		Green Building
ALOE503C		Electric Mobility
ALOE503D		Introduction to Stock Market
ALOE503E		Patent Law for Engineers and Scientist

BoS Chairman

ZES's Zeal College of Engineering & Research PU/PN/Engg./285/2007 Narhe, Pune-41

Director
ZES's Zeal College of Engineering & Research
Narhe, Pune - 411041.

DEPARTMENT OF INFORMATION TECHNOLOGY

Third Year B. Tech. – Information Technology: Semester – VI

Course Code	Course Type	Course Name	Teaching Scheme (Hrs./Week)							Evaluation Scheme					
			L	P	T	H	CR			CIE	ETE	TW	PR	OR	Total
							TH	PR/ Tut	Total						
ITPC611	PCC	Deep Learning	3	-	-	3	3	-	3	40	60	-	-	-	100
ITPC611P	PCC	Deep Learning Lab	-	2	-	2	-	1	1	-	-	-	25	-	25
ITPC612	PCC	Software Quality and Project Management	3	-	-	3	3	-	3	40	60	-	-	-	100
ITPE602	PEC	Program Elective Course - II *	2	-	-	2	2	-	2	40	60	-	-	-	100
ITPE602P	PEC	Program Elective Course - II * Lab.	-	2	-	2	-	1	1	-	-	-	-	25	25
ITPE603	PEC	Program Elective Course - III *	3	-	-	3	3	-	3	40	60	-	-	-	100
ITPE603P	PEC	Program Elective Course - III * Lab.	-	2	-	2	-	1	1	-	-	-	25	-	25
ITMD604	MDM	Internet of Things (IoT)	3	-	-	3	3	-	3	40	60	-	-	-	100
ITMD604P	MDM	Internet of Things (IoT) Lab.	-	2	-	2	-	1	1	-	-	-	-	25	25
ITVS605	VSEC	Programming Lab-III	-	4	-	4	-	2	2	-	-	25	50	-	75
ITIN605	ELC - INT	Internship - V	4 Weeks					2	2			25	-	-	25
Total			**14**	**12**	**0**	**26**	**14**	**8**	**22**	**200**	**300**	**50**	**100**	**50**	**700**

* - Select any one course from the given Program Elective Courses		
Course Code	**Course Type**	**Program Elective-II**
ITPE602A	PEC	Advanced Business Intelligence and Analytics
ITPE602B		High Performance Computing
ITPE602C		System Programming and Compiler Design

* - Select any one course from the given Program Elective Courses		
Course Code	**Course Type**	**Program Elective-III**
ITPE603A	PEC	Big Data Analytics
ITPE603B		Cryptography and Network Security
ITPE603C		Wireless Communication

BoS Chairman

PU/PN/Engg./285/2007 ZES's Zeal College of Engineering & Research Narhe, Pune-41

Director
ZES's Zeal College of Engineering & Research Narhe, Pune - 411041.

DEPARTMENT OF INFORMATION TECHNOLOGY

INDEX

SYLLABUS

SEMESTER - V

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)									**Semester:** V
Course: Automata Theory									**Code:** ITPC508
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	-	-	03	40	60	-	-	-	100

Prerequisites:

Basic knowledge of discrete mathematics, logic, sets, relations, and fundamental programming concepts is recommended.

Course Objectives:

1. Understand the mathematical foundations of formal languages and automata.
2. Analyze and design finite automata (DFA and NFA) for pattern recognition.
3. Study regular expressions and their equivalence with finite automata.
4. Explore context-free grammar (CFG), pushdown automata (PDA), and their applications.
5. Understand Turing Machines as a model of universal computation.
6. Investigate decidability, undecidability, and computational complexity classes.

Course Outcomes: After completion of this course, students will be able to -

CO1	Construct deterministic and non-deterministic finite automata for given languages.
CO2	Convert between regular expressions, DFAs, and NFAs using standard algorithms.
CO3	Design context-free grammar and pushdown automata for context-free languages.
CO4	Analyze properties of regular and context-free languages using pumping lemmas.
CO5	Design Turing Machines for computational problems and evaluate their limitations.
CO6	Differentiate between decidable, undecidable, and computationally complex problems.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to Formal Languages and Automata:** Overview of Theory of Computation: Automata, Computability, and Complexity, Alphabets, Strings, Languages, and Operations on Languages, Introduction to Finite Automata: Motivation and Applications, Deterministic Finite Automata (DFA): Definition, Transition Function, and Transition Diagrams, Languages Accepted by DFA, Non-Deterministic Finite Automata (NFA): Definition and Transition Function, Equivalence of DFA and NFA: Subset (Powerset) Construction Method, NFA with ε-Transitions (ε-NFA) and ε-Closure Concept, Minimization of DFA, mealy and Moore machine	7
2.	**Regular Languages and Regular Expressions:** Regular Expressions (RE): Syntax and Semantics, Languages Defined by Regular Expressions, Equivalence of Regular Expressions and Finite Automata, Arden's Theorem: RE from DFA, Thompson's Construction: NFA from RE, Pumping Lemma for Regular Languages and its Applications, Closure Properties of Regular Languages: Union, Concatenation, Kleene Star, Complement, Intersection, Decision Algorithms for Regular Languages: Emptiness, Finiteness, Membership	7

3.	**Context-Free Grammars and Languages:** Context-Free Grammars (CFG): Definition, Derivation, Parse Trees, Leftmost and Rightmost Derivations, Ambiguity in CFG, Removal of Ambiguity (where possible), Inherent Ambiguity, Simplification of CFGs: Elimination of Useless Symbols, Unit Productions, Null Productions, Normal Forms: Chomsky Normal Form (CNF) and Greibach Normal Form (GNF), Pumping Lemma for Context-Free Languages, Closure Properties of Context-Free Languages, Chomsky Hierarchy: Type 0, Type 1, Type 2, Type 3 Grammars	7
4.	**Pushdown Automata (PDA):** Pushdown Automata (PDA): Informal Description and Formal Definition, Deterministic and Non-Deterministic PDA (DPDA and NPDA), Languages Accepted by PDA: Acceptance by Final State and Empty Stack, Equivalence of PDA and Context-Free Grammars, Conversion of CFG to PDA and vice versa, Applications of PDA in parsing and compilers	6
5.	**Turing Machines:** Turing Machine (TM): Informal Description, Formal Definition, Instantaneous Descriptions, Language Accepted by a Turing Machine, Recursive and Recursively Enumerable Languages, Variants of Turing Machine: Multi-tape TM, Non-Deterministic TM, Church-Turing hypothesis, Universal Turing Machine (UTM)	6
6.	**Decidability and Computational Complexity:** Decidable and Undecidable Problems, Reduction Technique for Proving Undecidability, Rice's Theorem, Post's Correspondence Problem (PCP), Introduction to Computational Complexity: Time and Space Complexity, Complexity Classes: P, NP, NP-Complete, NP-Hard, Cook-Levin Theorem, Examples of NP-Complete Problems (SAT, 3-SAT), Undecidability: The Halting Problem (Proof by Diagonalization)	6
	TOTAL	**39**

Textbooks:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman – Introduction to Automata Theory, Languages, and Computation, 3rd Edition, Pearson Education.
2. Michael Sipser – Introduction to the Theory of Computation, 3rd Edition, Cengage Learning.
3. Peter Linz – An Introduction to Formal Languages and Automata, 6th Edition, Jones & Bartlett Learning.
4. J.C. Martin – Introduction to Languages and the Theory of Computation, 4th Edition, McGraw-Hill.

Reference Books:

1. Harry R. Lewis, Christos H. Papadimitriou – Elements of the Theory of Computation, Prentice Hall.
2. Dexter C. Kozen – Automata and Computability, Springer.
3. Thomas A. Sudkamp – Languages and Machines: An Introduction to the Theory of Computer Science, Pearson.

E-Resources:

1. NPTEL Course: Theory of Automata, Formal Languages and Computation – IIT Madras (nptel.ac.in)
2. MIT Open Courseware: 18.404J / 6.840J Theory of Computation
3. Coursera: Automata Theory – Stanford University (Prof. Jeffrey Ullman)

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)							**Semester:** V		
Course: Artificial Intelligence and Machine Learning (AIML)							**Code:** ITPC509		
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	-	-	03	40	60	-	-	-	100

Prerequisites:

Python programming, Data Structures, Linear Algebra, Probability & Statistics, Discrete Mathematics.

Course Objectives:

1. Understand AI and ML fundamentals.
2. Develop intelligent solutions using search, reasoning and planning.
3. Apply supervised and unsupervised learning algorithms.
4. Build AI/ML applications using Python libraries.
5. Understand deep learning, NLP, deployment and Explainable AI.
6. Analyze ethical and industrial aspects of AI/ML

Course Outcomes: After completion of this course, students will be able to -

CO1	Explain AI concepts, search and intelligent agents.
CO2	Apply knowledge representation and reasoning.
CO3	Perform data preprocessing and exploratory analysis.
CO4	Develop supervised and unsupervised ML models.
CO5	Build applications using deep learning and NLP.
CO6	Deploy AI/ML solutions with ethical considerations.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to Artificial Intelligence:** Overview of AI: History, Evolution, and Applications; Types of AI: Narrow AI, General AI, and Super AI; AI vs. Human Intelligence; Problem-solving approaches: Search Algorithms (BFS, DFS, A*); State Space Representation; Heuristic Search Techniques; Introduction to Intelligent Agents and Environments	6
2.	**Knowledge Representation and Reasoning:** Knowledge Representation: Logic, Semantic Networks, Frames, and Ontologies; Propositional and Predicate Logic; Inference Rules and Resolution; Expert Systems: Architecture, Forward and Backward Chaining; Uncertainty in AI: Bayesian Networks, Fuzzy Logic; Probabilistic Reasoning; Introduction to Natural Language Processing (NLP).its Applications, Closure Properties of Regular Languages: Union, Concatenation, Kleene Star, Complement, Intersection, Decision Algorithms for Regular Languages: Emptiness, Finiteness, Membership	7

3.	**AI Planning and Optimization** Classical AI Planning: STRIPS, PDDL; Goal-based and Utility-based Agents; Genetic Algorithms and Evolutionary Computing; Simulated Annealing; Ant Colony Optimization and Swarm Intelligence; Constraint Satisfaction Problems (CSP); Game Playing: Minimax Algorithm, Alpha-Beta Pruning.	7
4.	**Machine Learning Fundamentals** Introduction to Machine Learning (ML): Types – Supervised, Unsupervised, Reinforcement Learning; Linear Regression and Logistic Regression; Decision Trees and Random Forests; Support Vector Machines (SVM); Clustering: K-Means, Hierarchical Clustering; Dimensionality Reduction: PCA, LDA; Model Evaluation Metrics: Accuracy, Precision, Recall, F1-Score.	7
5.	**Supervised Learning and Unsupervised Learning: Supervised Learning:** Linear Regression, Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Introduction to XGBoost, Model Evaluation Metrics (Accuracy, Precision, Recall, F1-Score, ROC Curve). **Unsupervised Learning:** K-Means Clustering, Hierarchical Clustering, Principal Component Analysis (PCA), Recommendation Systems. **Deep Learning:** Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Introduction to Transformers and Large Language Models (LLMs)	6
6.	**Applications and AIML:** AI in Healthcare, Finance, Education, and Autonomous Systems; Computer Vision: Object Detection, Image Classification; Speech Recognition and Natural Language Processing Applications; Reinforcement Learning: Q-Learning, Policy Gradient Methods; Explainable AI (XAI); Ethical Issues: Bias, Fairness, Privacy, and Accountability; AI Governance and Regulatory Frameworks; Future Trends in AI.	6
	TOTAL	**39**

Textbooks:

1. Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig, Pearson Education, 4th Edition.
2. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, Aurélien Géron, O'Reilly Media, 3rd Edition.
3. Python Machine Learning, Sebastian Raschka and Vahid Mirjalili, Packt Publishing.
4. Deep Learning with Python, François Chollet, Manning Publications.
5. Introduction to Machine Learning with Python, O'Reilly Media.

Reference Books:

1. Nils J. Nilsson, "Principles of Artificial Intelligence," Tioga Publishing.
2. Kevin Murphy, "Machine Learning: A Probabilistic Perspective," MIT Press.
3. Aurélien Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow," O'Reilly Media.
4. François Chollet, "Deep Learning with Python," Manning Publications.
5. Artificial Intelligence, McGraw-Hill Education.

DEPARTMENT OF INFORMATION TECHNOLOGY

E-Resources:

1. NPTEL – Artificial Intelligence (IIT Madras): https://nptel.ac.in
2. fast.ai – Practical Deep Learning for Coders: https://www.fast.ai
3. TensorFlow Official Documentation: https://www.tensorflow.org
4. Scikit-learn Documentation: https://scikit-learn.orgsss
5. Kaggle – AI/ML Datasets and Competitions: https://www.kaggle.com
6. Google AI – Machine Learning Crash Course: https://developers.google.com/machine-learning/crash-course

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)							**Semester:** V		
Course: Artificial Intelligence and Machine Learning (AIML) Lab							**Code:** ITPC509P		
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
-	02	-	01	-	-	-	-	50	50

Prerequisites:

Python programming, Data Structures, Linear Algebra, Probability & Statistics, Discrete Mathematics.

Course Objectives:

1. **Understand** fundamental AI/ML concepts, techniques, and tools.
2. **Apply** AI algorithms such as search, logic, game playing, NLP, and Genetic Algorithms.
3. **Develop** ML solutions using data preprocessing, EDA, modeling, and evaluation techniques.
4. **Analyze and compare** supervised, unsupervised, and deep learning algorithms.
5. **Develop and deploy** real-world AI/ML applications considering performance and ethical aspects.

Course Outcomes: After completion of this course, students will be able to -

CO1	**Explain** fundamental AI/ML concepts and techniques.
CO2	**Apply** AI algorithms to solve computational problems.
CO3	**Implement** data preprocessing, EDA, and visualization techniques.
CO4	**Analyze** regression, classification, clustering, and PCA algorithms.
CO5	**Compare** ML and deep learning models for suitable applications.
CO6	**Develop and deploy** an AI/ML application considering performance and ethics.

Course Contents:

List of Experiments:

1. Installs NumPy, Pandas, and Scikit-learn. Create simple arrays, data frames, and train a basic ML model using sample data.
2. Implement BFS and DFS for graph traversal. Solve a simple maze or 8-puzzle problem using A* search.
3. Build a simple rule-based expert system using Forward Chaining and implement basic propositional logic inference.
4. Perform tokenization, stemming, and lemmatization on text. Create a simple sentiment analyzer and chatbot using keyword matching.
5. Implement the Minimax algorithm for Tic-Tac-Toe and improve it using Alpha-Beta Pruning.
6. Implement a Genetic Algorithm to solve an optimization problem such as the N-Queens problem or the Travelling Salesman Problem (TSP).
7. Perform data cleaning, handling missing values, feature scaling, encoding, and Exploratory Data Analysis (EDA) using Pandas, NumPy, and Matplotlib/Seaborn on a real-world dataset.
8. Implement Linear Regression and Logistic Regression using Scikit-learn and evaluate the models using appropriate performance metrics.
9. Implement and compare the performance of K-Nearest Neighbors (KNN), Decision Tree, Random Forest, and Support Vector Machine (SVM) classifiers using a benchmark dataset.
10. Apply K-Means Clustering and Hierarchical Clustering on a real-world dataset and perform Principal Component Analysis (PCA) for dimensionality reduction and visualization.

11. Develop a simple Artificial Neural Network (ANN) or Convolutional Neural Network (CNN) using TensorFlow/Keras for handwritten digit recognition (MNIST) or image classification.
12. Develop an end-to-end AI/ML application using a real-world dataset. The project should include data preprocessing, model development, evaluation, and deployment using Streamlit or Flask. Students should present performance analysis and discuss ethical considerations where applicable.

Textbooks:

1. Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig, Pearson Education, 4th Edition.
2. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, Aurélien Géron, O'Reilly Media, 3rd Edition.
3. Python Machine Learning, Sebastian Raschka and Vahid Mirjalili, Packt Publishing.
4. Deep Learning with Python, François Chollet, Manning Publications.
   Introduction to Machine Learning with Python, O'Reilly Media.

Reference Books:

1. Nils J. Nilsson, "Principles of Artificial Intelligence," Tioga Publishing.
2. Kevin Murphy, "Machine Learning: A Probabilistic Perspective," MIT Press.
3. Aurélien Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow," O'Reilly Media.
4. François Chollet, "Deep Learning with Python," Manning Publications.
5. Artificial Intelligence, McGraw-Hill Education.

E-Resources:

1. NPTEL – Artificial Intelligence (IIT Madras): https://nptel.ac.in
2. fast.ai – Practical Deep Learning for Coders: https://www.fast.ai
3. TensorFlow Official Documentation: https://www.tensorflow.org
4. Scikit-learn Documentation: https://scikit-learn.orgsss
5. Kaggle – AI/ML Datasets and Competitions: https://www.kaggle.com
6. Google AI – Machine Learning Crash Course: https://developers.google.com/machine-learning/crash-course

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)							**Semester:** V		
Course: Design and Analysis of Algorithms							**Code:** ITPC510		
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	-	-	03	40	60	-	-	-	100

Prerequisites:

Students should have basic programming knowledge, an understanding of data structures, recursion, and mathematical concepts.

Course Objectives:

1. To understand the fundamental concepts of algorithm design and analysis.
2. To apply divide and conquer and greedy strategies for problem solving.
3. To develop dynamic programming and backtracking solutions for complex problems.
4. To analyze the time and space complexity of algorithms using asymptotic notations.
5. To understand NP-completeness and computational complexity classes.
6. To design efficient algorithms for real-world computational problems.

Course Outcomes: After completion of this course, students will be able to -

CO1	Analyze the time and space complexity of algorithms using asymptotic notation.
CO2	Design and implement algorithms using Divide & Conquer and Greedy strategies.
CO3	Formulate and solve optimization problems using Dynamic Programming.
CO4	Apply Backtracking and Branch & Bound to constrain satisfaction and optimization problems.
CO5	Implement and evaluate graph algorithms for traversal, shortest path, and spanning trees.
CO6	Classify problems under complexity classes P, NP, and NP-Complete.

Unit	**Description**	**Duration (Hrs.)**
1.	**Foundations of Algorithms:** Introduction to algorithms: definition, properties, and importance, Algorithm design goals: correctness, efficiency, and clarity, Time Complexity and Space Complexity, Asymptotic notations: Big-O (O), Omega (Ω), Theta (Θ), Amortized analysis — introduction, Recurrence relations: substitution, recursion tree, Master Theorem, NP, NP-Hard, NP-Complete: definitions and relationships	6
2.	**Divide and Conquer:** Divide and Conquer strategy: paradigm and applicability, Binary Search: analysis and implementation, Merge Sort: algorithm, recurrence, Quick Sort: Lomuto and Hoare partition, pivot selection strategies, Strassen's Matrix Multiplication, finding maximum and minimum using D&C, solving recurrences using Master Theorem	6

3.	**Greedy Algorithms:** Greedy method: strategy, proof of correctness, when it works, Activity Selection Problem, Fractional Knapsack Problem, Huffman Coding: construction and analysis, Job Sequencing with Deadlines; Minimum Spanning Tree: Kruskal's and Prim's Algorithms, Single Source Shortest Paths: Dijkstra's Algorithm	8
4.	**Dynamic Programming:** Principle of Dynamic Programming: overlapping subproblems, optimal substructure, Memorization (top-down) vs Tabulation (bottom-up), Matrix Chain Multiplication; Longest Common Subsequence (LCS), 0/1 Knapsack Problem; Coin Change Problem, All-Pairs Shortest Paths: Floyd-Warshall Algorithm, Optimal Binary Search Tree; Edit Distance	6
5.	**Backtracking & Branch and Bound:** Backtracking: concept, state space tree, pruning, N-Queens Problem, Graph Coloring, Subset Sum using backtracking, Branch and Bound: LIFO, FIFO, and LC search; bounding functions, 0/1 Knapsack using Branch & Bound; Travelling Salesman Problem (TSP)	7
6.	**Graph Algorithms and NP-Completeness:** Graph traversals: BFS and DFS — algorithms, complexity, applications, Topological Sorting; Strongly Connected Components (Kosaraju's / Tarjan's), Bellman-Ford Algorithm; Johnson's Algorithm for All-Pairs SP, P, NP, NP-Hard, NP-Complete: definitions, Cook's theorem, Polynomial Reductions: SAT, 3-SAT, Clique, Vertex Cover, Subset Sum	6
	TOTAL	**39**

Textbooks:

1. Introduction to Algorithms — Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein
2. Algorithm Design — Jon Kleinberg, Éva Tardos
3. Fundamentals of Computer Algorithms — Ellis Horowitz, Sartaj Sahni
4. The Design and Analysis of Computer Algorithms — Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman
5. Data Structures and Algorithm Analysis in C++ — Mark Allen Weiss

Reference Books:

1. Introduction to Algorithms — Cormen, Leiserson, Rivest & Stein (CLRS), 4th Ed., MIT Press
2. Algorithm Design — Kleinberg & Tardos, Pearson Education
3. Data Structures and Algorithm Analysis in C — Weiss, Pearson

E-Resources:

1. https://nptel.ac.in/courses/106101059
2. https://ocw.mit.edu/courses/6-046j-design-and-analysis-of-algorithms-spring-2012/resources/mit6_046js12_notes/

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)							**Semester:** V		
Course: Design and Analysis of Algorithms Laboratory							**Code:** ITPC510P		
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
-	02	-	01	-	-	-	25	-	25

Prerequisites:

Students should have basic programming knowledge, an understanding of data structures, recursion, and mathematical concepts.

Course Objectives:

1. **Understand** fundamental algorithm design techniques and their computational complexity.
2. **Implement** searching, sorting, matrix multiplication, and graph algorithms.
3. **Apply** Greedy, Dynamic Programming, and Backtracking techniques to solve optimization problems.
4. **Analyze and compare** the performance of different algorithms based on time and space complexity.
5. **Develop** efficient algorithmic solutions for real-world computational problems

Course Outcomes: After completion of this course, students will be able to -

CO1	**Implement** searching, sorting, and matrix multiplication algorithms.
CO2	**Apply** Greedy techniques to solve optimization and compression problems.
CO3	**Implement** Dynamic Programming algorithms for optimization and shortest-path problems.
CO4	**Apply** Backtracking techniques to solve constraint-based problems.
CO5	**Implement** graph traversal, spanning tree, and shortest-path algorithms.
CO6	**Analyze and compare** algorithms based on time and space complexity.

Laboratory Experiments:

1. Implement Binary Search and analyze its efficiency.
2. Implement and compare Merge Sort and Quick Sort.
3. Implement Strassen's algorithm and compare it with conventional multiplication.
4. Apply the Greedy strategy to solve optimization problems.
5. Implement Huffman Coding using a Greedy approach.
6. Implement Prim's and Kruskal's algorithms and compare them.
7. Solve LCS using Dynamic Programming.
8. Solve the 0/1 Knapsack problem using Dynamic Programming.
9. Implement the Backtracking technique to solve N-Queens
10. Implement Graph Coloring using Backtracking.
11. Implement BFS, DFS, and Topological Sorting.
12. Implement BFS, DFS, and Topological Sorting.
13. Implement Floyd–Warshall using Dynamic Programming.

DEPARTMENT OF INFORMATION TECHNOLOGY

Textbooks:
1. Introduction to Algorithms — Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein 2. Algorithm Design — Jon Kleinberg, Éva Tardos 3. Fundamentals of Computer Algorithms — Ellis Horowitz, Sartaj Sahni 4. The Design and Analysis of Computer Algorithms — Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman 5. Data Structures and Algorithm Analysis in C++ — Mark Allen Weiss
Reference Books:
1. Introduction to Algorithms — Cormen, Leiserson, Rivest & Stein (CLRS), 4th Ed., MIT Press 2. Algorithm Design — Kleinberg & Tardos, Pearson Education 3. Data Structures and Algorithm Analysis in C — Weiss, Pearson
E-Resources:
1. https://nptel.ac.in/courses/106101059 2. https://ocw.mit.edu/courses/6-046j-design-and-analysis-of-algorithms-spring-2012/resources/mit6_046js12_notes/

Program Elective Course I

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)							**Semester:** V		
Course: Program Elective Course - I **(**Computer Networking and Technology)							**Code:** ITPE501A		
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	-	-	03	40	60	-	-	-	100

Prerequisites:

Basics of Computer Networks and Fundamentals of Computer Systems Network

Course Objectives:

1. Understand fundamentals of computer networks, communication models, transmission media, and networking devices
2. Learn switched networks, VLAN concepts, and the configuration of network connecting devices
3. Study application layer protocols and client-server communication using TCP and UDP
4. Understand IP addressing, subnetting, IPv4/IPv6 concepts, and network layer operations
5. Learn routing fundamentals, routing protocols, and router configuration techniques
6. Understand network security concepts, cryptographic algorithms, and common network attacks

Course Outcomes: After completion of this course, students will be able to -

CO1	Explain networking models, transmission media, topologies, and networking devices
CO2	Configure switched networks, VLANs, and inter-VLAN routing for efficient communication
CO3	Analyze application layer protocols and communication paradigms in computer networks
CO4	Apply IPv4/IPv6 addressing and subnetting techniques in network design
CO5	Configure and analyze static and dynamic routing operations in computer networks
CO6	Identify network security threats and apply basic cryptographic techniques for secure communication

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to Networks:** Network goals and applications, TCP/IP models, Guided and wireless transmission media, Guided Media: Twisted-Pair, Coaxial, and Fiber-Optic Cable. Unguided Media: Radio Waves, Microwaves, Infrared. (RJ45, RJ11, BNC, SC/ST etc.) , Network Devices – Hub, Bridge, Switch, Router and Network topologies Switching techniques, Transmission Media and Switching	6
2.	**Connecting devices and Switched Networks:** Connecting devices: Hubs, Repeaters, Active hubs, Bridges, Switches, Routers, Gateways Switching: Switched and Converged Networks, Switching Domains, Switch Configuration and Security, Virtual LANs, Inter-VLAN Routing	6
3.	**Application Layers :** Client Server Paradigm: Communication using TCP and UDP, Peer to Peer Paradigm, Application Layer Protocols: DNS, FTP, TFTP, HTTP, SMTP, POP, IMAP, DHCP, TELNET	7

DEPARTMENT OF INFORMATION TECHNOLOGY

Unit	Description	Duration (Hrs.)
4.	**Network Layer:** Network Layer Protocols, Routing, Routers, IP Addressing: IPv4 Network Addresses, IPv6 Network Addresses, Subnetting IP Networks: Subnetting an IPv4 Network, Addressing Schemes, Design Considerations for IPv6	7
5.	**Routing Fundamentals & Essentials:** Introduction to Routing, Router Functions and Configuration, Routing Decisions and Operations, Routing Table, Static Routing and Default Routing, Dynamic Routing and Protocols, The Routing Table, Static vs. Dynamic Routing	7
6.	**Introduction To Network Security & Cryptographic Algorithm:** Importance and Need for Security, Network Attacks- Passive, Active Network Security Threats: Unauthorized access, Distributed Denial of Service (DDoS) attacks, Man in the middle attacks, Mathematical preliminaries: Groups, Rings, Fields, Prime numbers, Symmetric key algorithms: Data Encryption Standards, Advanced Encryption Standard, Public Key Encryption and Hash function: RSA, Digital signatures, Digital Certificates and Public Key Infrastructure	6
	TOTAL	**39**

Textbooks:

1. Andrew S. Tenenbaum," Computer Networks", Pearson
2. Douglas E. Comer, "Computer Networks and Internets", Pearson Education
3. Behrouz A. Forouzan, TCP/IP Protocol Suite, McGraw-Hill Education, ISBN: 978-0-07-070652-1, 4th Edition.

Reference Books:

1. S. Tanenbaum: "Computer, Networks", PHI Publication, 4th edition, ISBN: 8178087855
2. Fourauzan B., "Data Communications and Networking", 5th Edition, Tata McGraw-Hill, Publications, ISBN: 0 – 07 – 058408 – 7
3. Andrew S. Tanenbaum, David J. Wethrall, Computer Network, Pearson Education

E-Resources:

1. https://www.netacad.com/courses/networking/networking-basics?utm_source=chatgpt.com
2. https://gaia.cs.umass.edu/kurose_ross/index.php?utm_source=chatgpt.com

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)									**Semester:** V
Course: Program Elective Course - I **(**Computer Networking and Technology Lab.)									**Code:** ITPE501AP
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
-	02	-	01	-	-	-	25	-	25

Prerequisites:

Basics of Computer Networks and Fundamentals of Computer Systems Network

Course Objectives:

1. Apply networking commands and protocols for basic network configuration and troubleshooting.
2. Configure routers, DNS, DHCP, gateways, and remote-access services using network simulation tools.
3. Design and implement small computer networks and analyze real-world network configurations.

Course Outcomes: After completion of this course, students will be able to -

CO1	Apply TCP/IP commands and utilities for network configuration and troubleshooting.
CO2	Configure Telnet/SSH services for secure remote network access.
CO3	Configure routers, interfaces, and access control lists using network simulators.
CO4	Design and implement small computer networks using Packet Tracer.
CO5	Configure DNS, DHCP, and gateway services in wired and wireless networks.
CO6	Analyze an existing network and its IP addressing scheme for effective network management.

List of Experiments: (All Essential)

1. Study of basic TCP/IP network commands and utilities (eg: ping, ifconfig, iproute, tracert, arp, tcpdump, host, netsat, nslookup, ftp, telnet etc.)
2. Installation and Configuration of Remote Login Service, Telnet/SSH, and access it through Telnet/SSH client
3. Configure a router (Ethernet & Serial Interface) using router commands, including access lists on any network Simulator (eg, Packet Tracer)
4. Network design and implementation for a small network using Packet Tracer
5. Study & Configuration of DNS & Gateways
6. Configuring a Cisco Router in similar and different networks
7. Installation and configuration of a DHCP Server in a Wireless Environment using an Access Point (Packet Tracer)
8. Case Study of existing College network with IP Address Scheme.

Textbooks:

1. Andrew S. Tenenbaum," Computer Networks", Pearson
2. Douglas E. Comer, "Computer Networks and Internets", Pearson Education
3. Behrouz A. Forouzan, TCP/IP Protocol Suite, McGraw-Hill Education, ISBN: 978-0-07-070652-1, 4th Edition.

Reference Books:

1. S. Tanenbaum: "Computer, Networks", PHI Publication, 4th edition, ISBN: 8178087855
2. Fourauzan B., "Data Communications and Networking", 5th Edition, Tata McGraw-Hill, Publications, ISBN: 0 – 07 – 058408 – 7
3. Andrew S. Tanenbaum, David J. Wethrall, Computer Network, Pearson Education

E-Resources:
1. https://www.netacad.com/courses/networking/networking-basics?utm_source=chatgpt.com 2. https://gaia.cs.umass.edu/kurose_ross/index.php?utm_source=chatgpt.com

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)								**Semester:** V	
Course: Program Elective-I (Data Visualization with Python)								**Code:** ITPE501B	
Teaching Scheme (Hrs/week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	-	-	03	40	60	-	-	-	100

Prerequisites:

Basic knowledge of Python programming, statistics, mathematics, database management systems, data science, spreadsheets, data handling, and computer programming fundamentals.

Course Objectives:

1. To introduce students to the fundamentals of data visualization and its importance in data analysis.
2. To develop skills in collecting, pre-processing, and visualizing data using Python libraries.
3. To understand various visualization techniques for exploratory data analysis and business insights.
4. To create interactive and advanced visualizations using modern visualization libraries.
5. To design dashboards and storytelling-based visualizations for decision making.
6. To apply visualization techniques to real-world datasets and applications.

Course Outcomes: After completion of this course, students will be able to -

CO1	Explain data visualization concepts, principles, and the role of visualization in data analysis.
CO2	Perform data preprocessing, cleaning, and exploratory analysis using Python libraries.
CO3	Develop visualizations using Matplotlib and Seaborn to effectively represent data.
CO4	Apply statistical and intermediate visualization techniques for analyzing relationships within datasets.
CO5	Design interactive visualizations and dashboards using modern visualization tools.
CO6	Create advanced visualizations and storytelling-based analytical solutions for real-world applications.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to Data Visualization and Python Fundamentals:** Introduction to Data Visualization, Importance of Visualization, Data Types and Visualization Concepts, Introduction to Python for Visualization, Python Environment Setup, Introduction to NumPy and Pandas, Loading CSV and Excel Files, Data Structures for Visualization	6
2.	**Data Preparation and Exploratory Data Analysis (EDA):** Data Collection Techniques, Data Cleaning, Handling Missing Values, Data Transformation, Descriptive Statistics, Exploratory Data Analysis Concepts, Data Aggregation, Grouping and Filtering using Pandas	7
3.	**Visualization using Matplotlib and Seaborn:** Introduction to Matplotlib, Line Charts, Bar Charts, Scatter Plots, Histograms, Pie Charts, Boxplots, Introduction to Seaborn, Distribution Plots, Heatmaps, Pair Plots, Customization Techniques	7
4.	**Intermediate Visualization and Statistical Visualization:** Correlation Analysis Visualization, Time Series Visualization, Multi-variable Visualization, Statistical Visualization Techniques, Feature Relationship Visualization, Advanced Seaborn Techniques, Visualization Best Practices	7

DEPARTMENT OF INFORMATION TECHNOLOGY

Unit	Description	Duration (Hrs.)
5.	**Interactive Visualization and Dashboard Development:** Interactive Visualization Concepts, Introduction to Plotly, Interactive Charts, Dashboard Design Principles, Creating Dashboards using Plotly Dash or Streamlit, Visualization for Business Intelligence Applications	6
6.	**Advanced Visualization, Storytelling and Applications:** Data Storytelling Concepts, Visualization Ethics, Large Dataset Visualization Challenges, Geographic Visualization Basics, Real-world Case Studies, Healthcare/Retail/Social Media Visualization Applications, Mini Project / Case Study Development	6
	TOTAL	**39**

Textbooks:

1. Python for Data Analysis – O'Reilly Media
2. Practical Statistics for Data Scientists – O'Reilly Media
3. Data Visualization with Python and JavaScript – Packt Publishing
4. Interactive Data Visualization for the Web – O'Reilly Media

Reference Books:

1. Storytelling with Data – Wiley
2. Fundamentals of Data Visualization – O'Reilly Media
3. Hands-On Data Visualization with Python – Packt Publishing
4. Python Data Science Handbook – O'Reilly Media
5. Visualize This – Wiley Publications

E-Resources:

1. https://plotly.com/python
2. https://seaborn.pydata.org
3. https://pandas.pydata.org/docs
4. https://pandas.pydata.org/docs
5. https://numpy.org/doc/

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)								**Semester:** V	
Course: Program Elective-I(Data Visualization with Python)								**Code:** ITPE501BP	
Teaching Scheme (Hrs/week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
-	02	-	01	-	-	-	25	-	25

Prerequisites:

Basic knowledge of Python programming, statistics, mathematics, database management systems, data science, spreadsheets, data handling, and computer programming fundamentals.

Course Objectives:

1. Apply Python tools for data handling, preprocessing, and exploratory analysis.
2. Develop effective static and interactive visualizations using Python libraries.
3. Analyze real-world datasets to identify patterns, trends, and insights through visualization.

Course Outcomes: After completion of this course, students will be able to -

CO1	Apply NumPy and Pandas for data manipulation, handling, and preprocessing.
CO2	Analyze datasets using descriptive statistics, grouping, filtering, and exploratory data analysis techniques.
CO3	Develop meaningful visualizations using Matplotlib and Seaborn to represent data effectively.
CO4	Analyze correlations, trends, distributions, and relationships among variables using statistical visualizations
CO5	Develop interactive visualizations and dashboards using Plotly, Dash, or Streamlit.
CO6	Apply data analysis, visualization, and storytelling techniques to solve problems using real-world datasets.

Course Contents:

Laboratory Assignments: (Any 10)

1. Develop Python programs using NumPy and Pandas to create arrays, manipulate datasets, and load data from CSV and Excel files.
2. Load a dataset and perform basic data preprocessing tasks such as handling missing values, removing duplicates, and transforming data using Pandas.
3. Perform Exploratory Data Analysis (EDA) by calculating descriptive statistics and applying grouping and filtering operations on datasets.
4. Create basic visualizations such as line charts, bar charts, pie charts, and histograms using Matplotlib.
5. Develop Python programs to create scatter plots, boxplots, and customized visualizations using Matplotlib.
6. Create statistical visualizations such as distribution plots, heatmaps, and pairplots using Seaborn.
7. Perform correlation analysis and visualize relationships between variables using correlation matrices and heatmaps.
8. Develop programs to create time-series visualizations and analyze trends using real-world datasets.
9. Create multi-variable visualizations and advanced Seaborn plots to analyze feature relationships within datasets.
10. Develop interactive visualizations using Plotly and create dynamic charts for data exploration.

11. Design and develop an interactive dashboard using Plotly Dash or Streamlit for visualizing business datasets. **12.** Develop a mini project using healthcare, retail, or social media datasets to perform preprocessing, visualization, dashboard creation, and data storytelling.
Textbooks:
1. Python for Data Analysis – O'Reilly Media 2. Practical Statistics for Data Scientists – O'Reilly Media 3. Data Visualization with Python and JavaScript – Packt Publishing 4. Interactive Data Visualization for the Web – O'Reilly Media
Reference Books:
1. Storytelling with Data – Wiley 2. Fundamentals of Data Visualization – O'Reilly Media 3. Hands-On Data Visualization with Python – Packt Publishing 4. Python Data Science Handbook – O'Reilly Media 5. Visualize This – Wiley Publications
E-Resources:
1. https://plotly.com/python 2. https://seaborn.pydata.org 3. https://pandas.pydata.org/docs 4. https://pandas.pydata.org/docs 5. https://numpy.org/doc/

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)							**Semester:** V		
Course: Program Elective-I (Human-Computer Interaction)							**Code:** ITPE501C		
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	-	-	03	40	60	-	-	-	100

Prerequisites:

Basic understanding of software engineering principles, SDLC, programming concepts (Python/Java/C++), computer graphics, digital media, operating systems, and basic web technologies like HTML/CSS.

Course Objectives:

1. To introduce the fundamental concepts, theories, and models underlying Human-Computer Interaction.
2. To develop skills in applying UI design principles, guidelines, and accessibility standards.
3. To equip students with user research methods to gather and analyze requirements effectively.
4. To train students in prototyping techniques and iterative interaction design processes.
5. To provide hands-on experience with usability evaluation methods and metrics.
6. To explore emerging HCI trends, including natural user interfaces, AR/VR, and AI-driven UX.

Course Outcomes: After completion of this course, students will be able to -

CO1	Explain the foundational concepts of HCI, the cognitive aspects of users, and identify usability issues in existing interfaces.
CO2	Apply UI design principles, visual design guidelines, and accessibility standards to create effective and inclusive user interfaces.
CO3	Conduct user research, develop personas and scenarios, and perform task analysis to derive meaningful design requirements.
CO4	Design and develop low-fidelity and high-fidelity prototypes for web and mobile applications using industry-standard tools.
CO5	Plan and execute usability evaluations using heuristic evaluation, usability testing, and quantitative metrics.
CO6	Analyse and design interfaces for emerging technologies, including NUI, AR/VR, voice assistants, and AI-driven personalization.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to Human-Computer Interaction:** Definition, Scope, and importance of HCI; History and evolution of HCI; The human in HCI: cognitive processes, memory, attention, and perception; The computer in HCI: input/output devices, interaction styles; The interaction: models of interaction, ergonomics; Usability: definitions, dimensions, and goals; Overview of HCI design principles (Nielsen's 10 Heuristics); Introduction to User-Centered Design (UCD) philosophy.	6

DEPARTMENT OF INFORMATION TECHNOLOGY

Unit	Description	Duration (Hrs.)
2.	**User Interface Design Principles and Guidelines:** Principles of good UI design: consistency, feedback, affordance, constraints, mappings; Design guidelines: platform-specific guidelines (Windows, iOS, Android, Web); Visual design principles: typography, color theory, layout, Gestalt principles; Interaction styles: command-line, menus, forms, direct manipulation, voice, gesture; Metaphors in UI design; Accessibility and inclusive design: WCAG standards, assistive technologies; Design patterns for UI: navigation patterns, content patterns, input patterns.	7
3.	**User Research and Requirements Gathering:** User research methods: interviews, surveys, focus groups, contextual inquiry; Personas: definition, creation, and usage in design; Scenarios and use design cases; Task analysis: HTA (Hierarchical Task Analysis), cognitive task analysis; Requirements gathering: functional and non-functional requirements from users; Affinity diagramming and data synthesis; Stakeholder analysis; Introduction to Design Thinking: Empathize, Define, Ideate, Prototype, Test.	7
4.	**Prototyping and Interaction Design:** Prototyping: low-fidelity (paper prototypes, wireframes) and high-fidelity prototypes; Wireframing tools, Interaction design principles: transitions, animations, micro-interactions; Information architecture: card sorting, site maps, navigation design; Responsive and adaptive design; Mobile-first design approach; Designing for multiple platforms: web, mobile, wearables, kiosks; Rapid prototyping techniques and iterative design**.**	7
5.	**Usability Evaluation Methods:** Usability testing: planning, conducting, and analyzing usability tests; Think-aloud protocol; Expert evaluation: heuristic evaluation, cognitive walkthrough; Inspection methods: pluralistic walkthrough, feature inspection; A/B Testing and multivariate testing; Eye-tracking and biometric evaluation; Quantitative and qualitative usability metrics; ISO 9241 usability standards; Formative vs. summative evaluation; Remote usability testing tools and methods	6
6.	**Emerging Trends in HCI:** Natural User Interfaces (NUI): touch, voice, gesture, and pen-based interaction; Augmented Reality (AR) and Virtual Reality (VR) interfaces; Wearable computing and IoT interfaces; Conversational interfaces: chatbots and voice assistants (Alexa, Siri, Google Assistant); Affective computing and emotion-aware interfaces; AI-powered personalization in UX; Universal Design and accessibility for diverse user populations; HCI in healthcare, education, and smart environments; Future directions: Brain-Computer Interfaces (BCI), Tangible UIs.	6
	TOTAL	**39**

Textbooks:

1. Alan Dix, Janet Finlay, Gregory D. Abowd, Russell Beale, "Human Computer Interaction," 3rd Edition, Pearson Education.
2. Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs, "Designing the User Interface: Strategies for Effective Human-Computer Interaction," 6th Edition, Pearson.
3. Jeff Johnson, "Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules," 3rd Edition, Morgan Kaufmann.

Reference Books:

1. Donald A. Norman, "The Design of Everyday Things," Revised and Expanded Edition, Basic Books.
2. Steve Krug, "Don't Make Me Think, Revisited: A Commonsense Approach to Web Usability," 3rd Edition, New Riders.
3. Jakob Nielsen, "Usability Engineering," Morgan Kaufmann Publishers.
4. Yvonne Rogers, Helen Sharp, Jenny Preece, "Interaction Design: Beyond Human-Computer Interaction," 5th Edition, Wiley.

E-Resources:

1. NPTEL – Human Computer Interaction (IIT): https://nptel.ac.in
2. Nielsen Norman Group – UX Research & Design Articles: https://www.nngroup.com
3. Interaction Design Foundation – HCI Courses: https://www.interaction-design.org
4. Google Material Design Guidelines: https://material.io/desisssgn
5. Apple Human Interface Guidelines: https://developer.apple.com/design/human-interface-guidelines
6. Figma Learning Resources: https://www.figma.com/resources/learn-design
7. Web Content Accessibility Guidelines (WCAG): https://www.w3.org/WAI/standards-guidelines/wcag

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)							**Semester:** V		
Course: Program Elective-I(Human-Computer Interaction Lab.)							**Code:** ITPE501CP		
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
-	02	-	01	-	-	-	25	-	25

Prerequisites:

Basic understanding of software engineering principles, SDLC, programming concepts (Python/Java/C++), computer graphics, digital media, operating systems, and basic web technologies like HTML/CSS.

Course Objectives:

1. Understand human-computer interaction principles, usability, and user-centered design.
2. Apply UX methods to analyze user requirements, interaction, accessibility, and usability.
3. Design and evaluate prototypes and interactive interfaces using appropriate HCI tools and techniques

Course Outcomes: After completion of this course, students will be able to -

CO1	**Apply** HCI principles, usability heuristics, and accessibility guidelines to interface design.
CO2	**Analyze** user requirements, tasks, and interaction problems using appropriate UX methods.
CO3	**Design** wireframes, prototypes, and user interfaces using HCI principles.
CO4	**Evaluate** interfaces using heuristic evaluation, usability testing, and usability metrics.
CO5	**Develop** interactive and accessible prototypes for web, mobile, and emerging applications.
CO6	**Recommend** improvements to interaction designs based on user feedback and usability findings.

Laboratory Assignments: (Any 10)

1. Evaluate the usability of an online shopping website using Nielsen's 10 heuristics and suggest improvements for navigation, feedback, and consistency.
2. Analyse the ordering process in a food delivery application to identify memory, attention, and usability issues faced by first-time users.
3. Create a visually appealing and accessible login/signup screen by applying color theory, typography, Gestalt principles, and WCAG 2.1 guidelines.
4. Design low-fidelity wireframes for a student portal with proper navigation, menu structure, and interaction consistency using any wireframe tool.
5. Conduct interviews and surveys to identify user requirements for an e-learning application and create personas and user scenarios.
6. Perform Hierarchical Task Analysis (HTA) for ATM cash withdrawal operations and identify usability issues in the transaction process.
7. Develop paper prototypes for searching and issuing books in a digital library application and improve the design through usability feedback.
8. Create a high-fidelity prototype for a travel booking system with responsive layouts, animations, and interactive transitions.
9. Conduct heuristic evaluation of a university website, assign severity ratings to identified issues, and prepare a usability report.
10. Plan and perform usability testing for common banking tasks such as fund transfer and balance enquiry, and compute usability metrics.

11. Design a conversational chatbot that answers student queries related to admissions, timetable, and examination schedules using Dialogflow.
12. Evaluate the interaction design of a virtual classroom AR/VR application and recommend improvements for user comfort and interaction efficiency.

Textbooks:

1. Alan Dix, Janet Finlay, Gregory D. Abowd, Russell Beale, "Human Computer Interaction," 3rd Edition, Pearson Education.
2. Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs, "Designing the User Interface: Strategies for Effective Human-Computer Interaction," 6th Edition, Pearson.
3. Jeff Johnson, "Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules," 3rd Edition, Morgan Kaufmann.

Reference Books:

1. Donald A. Norman, "The Design of Everyday Things," Revised and Expanded Edition, Basic Books.
2. Steve Krug, "Don't Make Me Think, Revisited: A Commonsense Approach to Web Usability," 3rd Edition, New Riders.
3. Jakob Nielsen, "Usability Engineering," Morgan Kaufmann Publishers.
4. Yvonne Rogers, Helen Sharp, Jenny Preece, "Interaction Design: Beyond Human-Computer Interaction," 5th Edition, Wiley.

E-Resources:

1. NPTEL – Human Computer Interaction (IIT): https://nptel.ac.in
2. Nielsen Norman Group – UX Research & Design Articles: https://www.nngroup.com
3. Interaction Design Foundation – HCI Courses: https://www.interaction-design.org
4. Google Material Design Guidelines: https://material.io/desisssgn
5. Apple Human Interface Guidelines: https://developer.apple.com/design/human-interface-guidelines
6. Figma Learning Resources: https://www.figma.com/resources/learn-design
7. Web Content Accessibility Guidelines (WCAG): https://www.w3.org/WAI/standards-guidelines/wcag

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)								**Semester:** V	
Course: Web Technology								**Code:** ITMD503	
Teaching Scheme (Hrs/week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
01	04	-	03	-	-	25	-	50	75

Prerequisites: Students should have basic knowledge of

- Core Java Programming
- Object-Oriented Programming Concepts
- HTML and CSS Basics
- Database Management Systems
- SQL Queries
- Internet and Web Technology Concepts

Course Objectives:

- To understand frontend and backend technologies used in Java Full Stack Development.
- To develop responsive and interactive web applications.
- To implement server-side applications using Java technologies.
- To integrate databases with web applications using JDBC and MySQL.
- To develop RESTful APIs and full-stack applications using Spring Boot.
- To deploy and maintain full-stack web applications.

Course Outcomes: After completion of this course, students will be able to -

CO1	Understand frontend and backend concepts used in Java Full Stack Development.
CO2	Design responsive web pages using HTML5, CSS3 and Bootstrap.
CO3	Develop dynamic web applications using Servlets and JSP.
CO4	Implement database connectivity using JDBC and MySQL.
CO5	Build RESTful APIs and full-stack applications using Spring Boot.
CO6	Deploy and manage Java Full Stack applications on hosting platforms.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	Introduction to Full Stack Development, Client-Server Architecture, Frontend vs Backend, Web Application Lifecycle	2
2.	HTML5 Fundamentals: Structure of HTML document, Semantic Tags, Lists, Tables, Forms and Multimedia Elements	2
3.	CSS3 Concepts: Selectors, Box Model, Flexbox, Grid Layout, Responsive Web Design	2
4.	Bootstrap Framework: Grid System, Components, Navigation Bar, Cards, Forms and Responsive UI	2
5.	JavaScript Fundamentals: Variables, Operators, Loops, Arrays, Functions and Objects	2
6.	JavaScript Advanced Concepts: DOM Manipulation, Events, Validation, JSON and AJAX	2
7.	Introduction to Java Web Technologies: Servlet Architecture, Servlet Lifecycle and Web ContSSainers	2

DEPARTMENT OF INFORMATION TECHNOLOGY

Unit	Description	Duration (Hrs.)
8.	JSP Concepts: JSP Tags, Directives, Session Management and Implicit Objects	2
9.	JDBC and Database Connectivity: JDBC Drivers, Connection Establishment, CRUD Operations using MySQL	2
10.	Spring Boot Fundamentals: Architecture, REST APIs, Controllers, Dependency Injection	2
11.	Web Security and Authentication: Session Handling, Cookies, SQL Injection Prevention, Authentication and Authorization	2
12.	Deployment Concepts: GitHub, Version Control, Hosting, FTP Deployment and Cloud Deployment Basics	2
	TOTAL	**24**

List of Experiments:

1. Create static webpages using HTML5 semantic tags, lists, tables and forms
2. Design responsive webpages using CSS3 and Bootstrap components
3. Implement JavaScript programs using functions, arrays and objects
4. Develop webpage validation using JavaScript and DOM manipulation
5. Create dynamic webpages using AJAX and JSON
6. Develop basic Servlet applications using Apache Tomcat Server
7. Create JSP-based applications for session handling and form processing
8. Implement CRUD operations using JDBC and MySQL database
9. Develop REST APIs using Spring Boot Framework
10. Develop and deploy a Mini Java Full Stack Project integrating frontend, backend and database

Mini Project (Any Two)

a. Online Student Management System
b. Library Management System
c. Employee Management Portal
d. Online Food Ordering System
e. E-Commerce Website
f. Online Examination System
g. Hospital Management System
h. Event Registration Portal

Text Books:

1. Herbert Schildt, Java: The Complete Reference, McGraw Hill.
2. Cay S. Horstmann, Core Java Volume I, Pearson Education.
3. Jon Duckett, HTML and CSS: Design and Build Websites.
4. Deitel & Deitel, Internet and World Wide Web – How to Program.
5. Craig Walls, Spring Boot in Action, Manning Publications.

DEPARTMENT OF INFORMATION TECHNOLOGY

Reference Books:

1. Marty Hall, Core Servlets and JavaServer Pages.
2. V.K. Jain, Advanced Java Programming, DreamTech Press.
3. Kathy Sierra & Bert Bates, Head First Java, O'Reilly Publications.
4. Bryan Basham, Kathy Sierra & Bert Bates, Head First Servlets and JSP.

E-Resources:

1. Oracle Java Documentation
2. Spring Boot Documentation
3. MDN Web Docs
4. Bootstrap Documentation
5. W3Schools Java Tutorials
6. NPTEL Java Courses

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)									**Semester:** V
Course: Internship - IV									**Code:** ITIN504
Teaching Scheme				**Evaluation Scheme**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
4 Week			02		-	25	-	-	25
Kindly refer to the policy of internship on the College Website									

SYLLABUS

SEMESTER - VI

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)							**Semester:** VI		
Course: Deep Learning							**Code:** ITPC611		
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	-	-	03	40	60	-	-	-	100

Prerequisites: Students should have basic knowledge

- Artificial Intelligence
- Machine Learning
- Python Programming
- Probability and Statistics
- Linear Algebra
- Calculus Fundamentals

Course Objectives:

1. To understand advanced neural network architectures and deep learning fundamentals.
2. To develop deep learning models for computer vision applications.
3. To apply sequence models, transformers, and attention mechanisms in NLP tasks.
4. To explore generative AI models, including GANs, VAEs, and Large Language Models.
5. To implement transfer learning and model optimization techniques.
6. To analyse, explain, and deployment challenges of deep learning systems.

Course Outcomes: After completion of this course, students will be able to -

CO1	Explain advanced deep learning concepts, architectures, and optimization techniques.
CO2	Develop and evaluate deep learning models for image classification and computer vision tasks.
CO3	Apply sequence models, attention mechanisms, and transformers for language processing applications.
CO4	Implement generative AI models, including GANs and diffusion-based architectures.
CO5	Utilize transfer learning, model fine-tuning, and deployment frameworks for real-world applications.
CO6	Evaluate ethical, explainability, and sustainability aspects of deep learning systems.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Foundations of Deep Learning:** Review of Neural Networks, Deep Learning Ecosystem, Forward and Backpropagation, Gradient Descent Variants, Loss Functions, Regularization Techniques, Batch Normalization, Hyperparameter Tuning, Deep Learning Frameworks (TensorFlow, PyTorch).	6
2.	**Deep Learning for Computer Vision:** CNN Architecture Review, Advanced CNN Architectures: AlexNet, VGG, ResNet, Efficient Net, Image Classification, Object Detection, Transfer Learning in Vision Tasks, Vision Applications.	6

3.	**Sequence Models and Natural Language Processing**: Sequence Modeling, LSTM and GRU Networks, Attention Mechanism, Transformer Architecture Word Embeddings, BERT Fundamentals, NLP Applications	6
4.	**Generative AI and Foundation Models:** Generative AI Overview, Autoencoders, Variational Autoencoders (VAE), GAN Architecture, Diffusion Models, Large Language Models (LLMs), Prompt Engineering, Retrieval-Augmented Generation (RAG).	7
5.	**Model Optimization and Deployment:** Transfer Learning, Fine-Tuning Techniques, Model Compression, Quantization, Edge AI Concepts, Lops Fundamentals, Deployment using Streamlit and FastAPI, Monitoring Deep Learning Models	7
6.	**Explainable and Responsible Deep Learning:** Explainable AI (XAI), SHAP and LIME, Bias and Fairness, Responsible AI, AI Governance, Privacy-Preserving Deep Learning, Green AI, Future Trends in Deep Learning	6
	TOTAL	**39**

Textbooks:

1. François Chollet, Deep Learning with Python, Manning Publications.
2. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Publications.
3. Nikhil Buduma, Fundamentals of Deep Learning, O'Reilly Publications.
4. Charu C. Aggarwal, Neural Networks and Deep Learning, Springer Publications.

Reference Books:

1. David Foster, Generative Deep Learning, O'Reilly Publications.
2. Lewis Tunstall, Leandro von Werra, Thomas Wolf, Natural Language Processing with Transformers, O'Reilly Publications.
3. Mohamed Elgendy, Deep Learning for Vision Systems, Manning Publications.
4. Andriy Burkov, Machine Learning Engineering, True Positive Publications.
5. Sebastian Raschka, Yuxi (Hayden) Liu, Vahid Mirjalili, Machine Learning with PyTorch and Scikit-Learn, Packt Publications.
6. Josh Patterson, Adam Gibson, Deep Learning: A Practitioner's Approach, O'Reilly Publications.

E-Resources:

1. TensorFlow Documentation – https://www.tensorflow.org
2. PyTorch Documentation – https://pytorch.org
3. DeepLearning.AI Courses – https://www.deeplearning.ai
4. fast.ai Practical Deep Learning Courses – https://www.fast.ai
5. Kaggle Learn and Datasets – https://www.kaggle.com
6. Google Colab Documentation – https://colab.research.google.com
7. NPTEL Courses on Deep Learning and Artificial Intelligence – https://nptel.ac.in
8. SWAYAM Online Courses on AI and Deep Learning – https://swayam.gov.in

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)								**Semester:** VI	
Course: Deep Learning Laboratory								**Code:** ITPC611P	
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
-	02	-	01	-	-	-	-	25	25

Prerequisites: Students should have basic knowledge

Students should have basic knowledge of Python programming, linear algebra, probability, statistics, and machine learning fundamentals.

Course Objectives:

1. Understand fundamental deep learning concepts, architectures, and frameworks.
2. Implement deep learning models for image, text, and generative applications.
3. Evaluate and deploy deep learning models using modern tools and explainability techniques.

Course Outcomes: After completion of this course, students will be able to -

CO1	**Implement** basic deep learning models using TensorFlow and PyTorch.
CO2	**Develop** CNN and transfer learning models for image classification.
CO3	**Implement** LSTM and Transformer models for NLP applications.
CO4	Develop autoencoders and GANs for generative and representation learning.
CO5	**Evaluate** and fine-tune deep learning models using suitable performance measures.
CO6	**Deploy** and **explain** deep learning models using Streamlit, SHAP, or LIME.

List of Experiments:

1. Environment setup using TensorFlow and PyTorch.
2. Implement a Multilayer Perceptron (MLP) for classification.
3. Build a CNN model for image classification using CIFAR-10.
4. Apply Transfer Learning using ResNet or MobileNet.
5. Perform Image Augmentation and evaluate model performance.
6. Implement LSTM for sentiment analysis.
7. Develop a Transformer-based text classification model.
8. Build an Autoencoder for dimensionality reduction.
9. Implement GAN for image generation.
10. Fine-tune a pre-trained BERT model for NLP tasks.
11. Deploy a deep learning model using Streamlit.
12. Explain model predictions using SHAP or LIME.

Textbooks:

1. François Chollet, Deep Learning with Python, Manning Publications.
2. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Publications.
3. Nikhil Buduma, Fundamentals of Deep Learning, O'Reilly Publications.
4. Charu C. Aggarwal, Neural Networks and Deep Learning, Springer Publications.

DEPARTMENT OF INFORMATION TECHNOLOGY

Reference Books:

1. David Foster, Generative Deep Learning, O'Reilly Publications.
2. Lewis Tunstall, Leandro von Werra, Thomas Wolf, Natural Language Processing with Transformers, O'Reilly Publications.
3. Mohamed Elgendy, Deep Learning for Vision Systems, Manning Publications.
4. Andriy Burkov, Machine Learning Engineering, True Positive Publications.
5. Sebastian Raschka, Yuxi (Hayden) Liu, Vahid Mirjalili, Machine Learning with PyTorch and Scikit-Learn, Packt Publications.
6. Josh Patterson, Adam Gibson, Deep Learning: A Practitioner's Approach, O'Reilly Publications.

E-Resources:

1. TensorFlow Documentation – https://www.tensorflow.org
2. PyTorch Documentation – https://pytorch.org
3. DeepLearning.AI Courses – https://www.deeplearning.ai
4. fast.ai Practical Deep Learning Courses – https://www.fast.ai
5. Kaggle Learn and Datasets – https://www.kaggle.com
6. Google Colab Documentation – https://colab.research.google.com
7. NPTEL Courses on Deep Learning and Artificial Intelligence – https://nptel.ac.in
8. SWAYAM Online Courses on AI and Deep Learning – https://swayam.gov.in

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)								**Semester:** VI	
Course: Software Quality Engineering and Project Management								**Code:** ITPC612	
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	-	-	03	40	60	-	-	-	100

Prerequisites: Students should have prior knowledge of:

- Fundamentals of Software Engineering and SDLC.
- Programming concepts and Object-Oriented Programming.
- Database Management Systems.
- Agile and Waterfall Development Models.
- Software Requirement Engineering.
- Basic Project Planning and Documentation.
- Software Development Tools and Version Control Systems.

Course Objectives:

1. To understand software project management principles and methodologies.
2. To apply project planning, scheduling, estimation, and risk management techniques.
3. To understand software testing processes and quality assurance practices.
4. To design effective test cases using industry-standard testing techniques.
5. To use project management and testing tools for software development projects.
6. To evaluate software quality through testing, metrics, and quality standards

Course Outcomes: After completion of this course, students will be able to -

CO1	Explain software project management concepts and software testing fundamentals.
CO2	Prepare project plans, schedules, and effort estimations for software projects.
CO3	Identify and manage software project risks and quality assurance activities.
CO4	Design test cases using black-box and white-box testing techniques.
CO5	Utilize project management and defect tracking tools effectively.
CO6	Evaluate software quality through automation, performance testing, and quality metrics.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to Software Project Management and Software Testing**: Software Project Management Overview, Project Life Cycle, Role of Project Manager, Software Development Life Cycle (SDLC)	6
2.	**Data Collection and Preprocessing**: Scope Management, Work Breakdown Structure (WBS), Effort Estimation Techniques (COCOMO, Function Point Analysis), Project Scheduling using Gantt Charts and Network Diagrams, Resource Allocation, Cost Estimation and Budgeting.	6
3.	**Exploratory Data Analysis and Visualization**: Software Risk Identification, Risk Assessment Matrix, Risk Mitigation Strategies, Software Quality Assurance (SQA), Quality Planning, Quality Control vs Quality Assurance, ISO 9001, ISO/IEC 25010 Quality Model, Capability Maturity Model Integration (CMMI).	6

4.	**Test Case Design and Defect Management:** **Introduction to Software Testing:** Software Testing Fundamentals, Verification vs Validation, Software Testing Life Cycle (STLC), Levels of Testing (Unit, Integration, System, Acceptance), Software Quality Concepts**.** Black-Box Testing (Equivalence Partitioning, Boundary Value Analysis, Decision Tables, State Transition Testing), White-Box Testing (Statement, Branch and Path Coverage), Test Plan Preparation, Defect Life Cycle, Severity and Priority Classification, Test Metrics and Reporting.	7
5.	**Agile Project Management and Test Automation:** Agile Principles, Scrum Framework, Sprint Planning, Product Backlog, User Stories, Burndown Charts, Introduction to Test Automation, Automation Frameworks, Selenium WebDriver, TestNG/JUnit, Continuous Integration and Continuous Testing using Jenkins and GitHub Actions.	7
6.	**Project Monitoring, Quality Metrics and Advanced Testing:** Project Monitoring and Control, Earned Value Management (EVM), Software Quality Metrics, Reliability Metrics, Performance Testing, Load Testing, Stress Testing, Security Testing Fundamentals, OWASP Top 10, Software Maintenance, and Project Closure.	7
	TOTAL	**39**

Textbooks:

1. Roger S. Pressman – Software Engineering: A Practitioner's Approach, McGraw-Hill.
2. Bob Hughes, Mike Cotterell – Software Project Management, McGraw-Hill.
3. Glenford J. Myers – The Art of Software Testing, Wiley.
4. Rex Black – Managing the Testing Process, Wile

Reference Books:

1. Kathy Schwalbe – Information Technology Project Management.
2. Ian Sommerville – Software Engineering.
3. Elfriede Dustin – Automated Software Testing.
4. PMBOK Guide – Project Management Institute (PMI).

E-Resources:

1. PMI Project Management Institute
2. ISTQB Foundation Level Syllabus
3. Selenium Documentation
4. Apache JMeter Documentation
5. OWASP Testing Guide
6. NPTEL Software Project Management Course

Program Elective Course II

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)							**Semester:** VI		
Course: Program Elective – II (Advanced Business Intelligence and Analytics)							**Code:** ITPE602A		
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
02	-	-	02	40	60	-	-	-	100

Prerequisites:

Basic knowledge of business processes, computer applications, spreadsheets, and databases is recommended. Familiarity with SQL or Python is beneficial but not mandatory.

Course Objectives:

1. **Understand** the concepts, architecture, components, and applications of Business Intelligence and Business Analytics in organizational environments.
2. **Design and implement** data warehouse architectures and ETL solutions for business reporting and analytical requirements.
3. **Develop** interactive dashboards, scorecards, and business reports using modern BI tools and visualization techniques.
4. **Apply** descriptive and predictive analytics techniques to support data-driven business decision-making.
5. **Perform** data mining and sentiment analysis on structured and unstructured datasets to extract meaningful business insights.
6. **Build and evaluate** decision-support analytical solutions and effectively communicate analytical findings using real-world business datasets.

Course Outcomes: After completion of this course, students will be able to -

CO1	**Explain** BI architecture, components, and business applications in organizational decision-making systems.
CO2	**Design and develop** dimensional data models and ETL workflows for efficient data integration and warehousing
CO3	**Create** interactive dashboards, scorecards, and business reports using modern BI tools and visualization techniques.
CO4	**Apply** descriptive and predictive analytics techniques to analyze business problems and support managerial decisions
CO5	**Perform and analyze** data mining and sentiment analysis for extracting actionable business insights from structured and unstructured data
CO6	**Develop and present** decision-support analytical solutions using real-world datasets and effective data visualization methods.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**BI Fundamentals and Architecture:** Introduction to Business Intelligence, BI Architecture, Applications of BIKPIs, and Metrics	4
2.	**Data Warehousing and ETL:** Database and OLAP Concepts, Data Warehouse Architecture, Dimensional Modeling, ETL Processes, Data Quality and Governance	5

3.	**Dash boarding and Business Reporting:** Business Reporting Concepts, Data Visualization Principles, BI Tools and Platforms, Interactive Reporting	4
4.	**Business Analytics Techniques:** Descriptive Analytics, Predictive Analytics, Data Visualization for Analytics, Business Use Cases	4
5.	**Data Mining and Sentiment Analysis:** Introduction to Data Mining, Sentiment Analysis Concepts, Business Applications	5
6.	**Decision Support and Capstone Project:** Decision Support Systems (DSS), BI Governance and Strategy, BI Project Lifecycle	4
	TOTAL	**26**

Textbooks:

1. Business Intelligence, Analytics, and Data Science: A Managerial Perspective by Ramesh Sharda, Dursun Delen, and Efraim Turban, Pearson.
2. Data Warehousing Fundamentals By Paulraj Ponniah, Wiley Publications.
3. Microsoft Power BI Data Analyst Certification Guide By Thomas LeBlanc, Packt Publishing.
4. Fundamentals of Business Analytics By RN Prasad and Seema Acharya, Wiley India.

Reference Books:

1. Competing on Analytics, Harvard Business Review Press.
2. The Data Warehouse Toolkit Wiley Publications.
3. Business Analytics Pearson Education.
4. Practical Tableau O'Reilly Media.
5. Data Mining Concepts and Techniques, Morgan Kaufmann Publishers.

E-Resources:

1. https://learn.microsoft.com/en-us/training/modules/get-started-with-power-bi/2-using-power-bi
2. https://www.tableau.com/learn/training?utm
3. https://www.coursera.org/learn/introduction-to-tableau

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)							**Semester:** VI		
Course: Program Elective – II (Advanced Business Intelligence and Analytics) Lab.							**Code:** ITPE602AP		
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
-	02	-	01	-	-	-	25	-	25

Prerequisites:

Basic knowledge of business processes, computer applications, spreadsheets, and databases is recommended. Familiarity with SQL or Python is beneficial but not mandatory.

Course Objectives:

1. Understand business intelligence concepts, data warehousing, ETL, and OLAP techniques.
2. Apply data analytics and visualization techniques to generate meaningful business insights.
3. Develop interactive BI dashboards and analytical solutions for business decision-making.

Course Outcomes: After completion of this course, students will be able to -

CO1	**Apply** BI concepts, ETL processes, and data warehousing techniques to business data.
CO2	**Perform** OLAP operations to analyze multidimensional business data.
CO3	**Develop** interactive dashboards and reports using suitable BI and visualization tools
CO4	**Analyze** business data using descriptive and predictive analytics techniques.
CO5	**Apply** machine learning techniques for customer segmentation, market basket analysis, and forecasting.
CO6	**Develop** an end-to-end BI solution to support data-driven business decision-making.

Laboratory Assignment:

1. A retail company maintains sales data in Excel spreadsheets but lacks meaningful insights. Develop a basic BI report using a BI tool to identify key business performance indicators (KPIs) such as total sales, profit, and regional performance
2. An organization stores business transactions in a relational database. Connect a BI tool to the database, import the data, and prepare it for reporting by validating the imported records.
3. A supermarket chain wants to analyze sales across products, customers, stores, and time. Design a data warehouse using Star or Snowflake schema to support efficient analytical queries.
4. Business data is collected from multiple sources containing duplicate, inconsistent, and missing values. Perform ETL operations to clean, transform, and integrate the data into a unified dataset for analysis.
5. A sales manager needs to analyze business performance across different dimensions such as region, product category, and year. Perform OLAP operations such as Slice, Dice, Drill-down, Roll-up, and Pivot to generate business insights.
6. A company's management requires a real-time dashboard to monitor sales, profit, customer growth, and regional performance. Design an interactive dashboard using appropriate charts, KPIs, filters, and slicers.
7. A business analyst is required to prepare monthly performance reports for management. Develop professional reports using appropriate visualization techniques to present trends, comparisons, and business performance.
8. A retail organization wants to understand historical sales trends and customer behavior. Perform descriptive analytics using Python and visualize business insights through statistical summaries and graphical representations.

9. A company wants to forecast future sales and predict customer purchasing behavior using historical data. Build and evaluate suitable predictive models using machine learning algorithms.
10. A marketing department wants to identify groups of customers with similar purchasing patterns to design targeted marketing campaigns. Apply clustering and association rule mining techniques for customer segmentation and market basket analysis.
11. An e-commerce company receives thousands of customer reviews every day. Analyze customer opinions using sentiment analysis techniques to classify reviews as positive, negative, or neutral and summarize customer satisfaction.
12. Develop an end-to-end Business Intelligence solution for a real-world business problem by integrating data collection, ETL, data warehousing, dashboard development, business analytics, and decision support to provide actionable business insights.

Textbooks:

1. Business Intelligence, Analytics, and Data Science: A Managerial Perspective By Ramesh Sharda, Dursun Delen, and Efraim Turban, Pearson.
2. Data Warehousing Fundamentals By Paulraj Ponniah, Wiley Publications.
3. Microsoft Power BI Data Analyst Certification Guide By Thomas LeBlanc, Packt Publishing.
4. Fundamentals of Business Analytics By RN Prasad and Seema Acharya, Wiley India.

Reference Books:

1. Competing on Analytics, Harvard Business Review Press.
2. The Data Warehouse Toolkit Wiley Publications.
3. Business Analytics Pearson Education.
4. Practical Tableau O'Reilly Media.
5. Data Mining Concepts and Techniques, Morgan Kaufmann Publishers.

E-Resources:

1. https://learn.microsoft.com/en-us/training/modules/get-started-with-power-bi/2-using-power-bi
2. https://www.tableau.com/learn/training?utm
3. https://www.coursera.org/learn/introduction-to-tableau

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)							**Semester:** VI		
Course: Program Elective-II (High Performance Computing)							**Code:** ITPE602B		
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
02	-	-	02	40	60	-	-	-	100

Prerequisites:

Basic understanding of Data Structures and Algorithms, Operating Systems, Computer Organization and Architecture, Programming in C/C++ or Python.

Course Objectives:

1. To understand the principles and need for parallel and high-performance computing.
2. To study modern parallel computing architectures and programming models.
3. To learn message passing and shared memory programming techniques.
4. To analyze performance metrics and scalability of parallel systems.
5. To develop parallel applications using OpenMP and GPU programming concepts.
6. To explore contemporary HPC applications in AI, cloud computing, and data analytics.

Course Outcomes: After completion of this course, students will be able to -

CO1	Explain parallel computing concepts, architectures, and HPC platforms.
CO2	Apply decomposition and parallel programming strategies to solve computational problems.
CO3	Develop shared memory and distributed memory parallel programs.
CO4	Evaluate performance metrics such as speedup, efficiency, and scalability.
CO5	Implement GPU-based parallel solutions using modern accelerator programming concepts.
CO6	Analyze HPC applications in scientific computing, AI, big data, and cloud environments.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to High Performance Computing:** Evolution of HPC, Need for Parallel Computing, Flynn's Classification, Shared Memory Systems, Distributed Memory Systems, Hybrid Architectures, Multicore and Manycore Processors, HPC Applications.	4
2.	**Parallel Programming Fundamentals:** Task and Data Parallelism, Decomposition Techniques, Load Balancing, Dependency Analysis, Parallel Algorithm Design Strategies, Pipeline Processing, Divide and Conquer Paradigm	4
3.	**Parallel Programming Models:** Message Passing Model, MPI Concepts, Shared Memory Programming, OpenMP Fundamentals, Synchronization Techniques, Race Conditions, and Deadlocks.	5
4.	**Performance Analysis of Parallel Systems:** Performance Metrics, Speedup, Efficiency, Cost Metrics, Amdahl's Law, Gustafson's Law, Scalability Analysis, Benchmarking Concepts	5

5.	**GPU and Accelerator Computing:** Introduction to GPU Computing, CUDA Architecture, GPU Memory Hierarchy, Kernel Execution Concepts, Heterogeneous Computing, AI Accelerators and TPUs, Introduction to OpenCV.	4
6.	**Modern HPC Applications**: Parallel Matrix Operations, Scientific Simulations, Big Data Processing, AI and Machine Learning Workloads, Cloud-Based HPC, Edge HPC Concepts, Future Trends in HPC.	4
	TOTAL	**26**

Textbooks:

1. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, "Introduction to Parallel Computing", Addison-Wesley, Second Edition.
2. Jason Sanders, Edward Kandrot, "CUDA by Example", Addison-Wesley.
3. Peter Pacheco, "An Introduction to Parallel Programming", Morgan Kaufmann.

Reference Books:

1. Kai Hwang, "Scalable Parallel Computing", McGraw Hill.
2. Shane Cook, "CUDA Programming: A Developer's Guide to Parallel Computing with GPUs", Morgan Kaufmann.
3. David Culler, Jaswinder Pal Singh, "Parallel Computer Architecture: A Hardware/Software Approach", Morgan Kaufmann.
4. Michael J. Quinn, "Parallel Programming in C with MPI and OpenMP", McGraw Hill.
5. Georg Hager, Gerhard Wellein, "Introduction to High Performance Computing for Scientists and Engineers", CRC Press

E-Resources:

1. https://swayam.gov.in
2. https://www.openmp.org
3. https://developer.nvidia.com/cuda-zoneS

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)							**Semester:** VI		
Course: Program Elective-II (High Performance Computing) Lab							**Code:** ITPE602BP		
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
-	02	-	01	-	-	-	25	-	25

Prerequisites:

Basic understanding of Data Structures and Algorithms, Operating Systems, Computer Organization and Architecture, Programming in C/C++ or Python.

Course Objectives:

1. Understand parallel computing concepts, architectures, and performance measures.
2. Implement parallel programs using OpenMP, MPI, and CUDA.
3. Analyze parallel algorithms and evaluate their performance on multicore and GPU systems.

Course Outcomes: After completion of this course, students will be able to -

CO1	**Explain** parallel computing concepts and multicore architectures.
CO2	**Implement** parallel algorithms using OpenMP.
CO3	**Implement** distributed programs using MPI communication techniques.
CO4	**Develop** GPU-based applications using CUDA.
CO5	Analyze and compare the performance of sequential and parallel programs.
CO6	**Evaluate** scalability and speedup of HPC applications using suitable performance measures.

Laboratory Assignments:

1. Performance comparison between sequential and parallel execution.
2. Parallel implementation of matrix addition.
3. Parallel implementation of matrix multiplication using OpenMP.
4. Parallel reduction operations (sum, maximum, minimum).
5. Parallel sorting algorithm implementation.
6. MPI program for message passing.
7. MPI broadcast and reduction operations.
8. Study of multicore processor utilization.
9. CUDA program for vector addition.
10. CUDA program for matrix multiplication.
11. Performance measurement using Amdahl's Law.
12. Mini project on HPC application

Textbooks:

1. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, "Introduction to Parallel Computing", Addison-Wesley, Second Edition.
2. Jason Sanders, Edward Kandrot, "CUDA by Example", Addison-Wesley.
3. Peter Pacheco, "An Introduction to Parallel Programming", Morgan Kaufmann.

Reference Books:

1. Kai Hwang, "Scalable Parallel Computing", McGraw Hill.
2. Shane Cook, "CUDA Programming: A Developer's Guide to Parallel Computing with GPUs", Morgan Kaufmann.
3. David Culler, Jaswinder Pal Singh, "Parallel Computer Architecture: A Hardware/Software Approach", Morgan Kaufmann.
4. Michael J. Quinn, "Parallel Programming in C with MPI and OpenMP", McGraw Hill.
5. Georg Hager, Gerhard Wellein, "Introduction to High Performance Computing for Scientists and Engineers", CRC Press

E-Resources:

1. https://swayam.gov.in
2. https://www.openmp.org
3. https://developer.nvidia.com/cuda-zoneS

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)							**Semester:** VI		
Course: Program Elective-II (System Programming and Compiler Design)							**Code:** ITPE602C		
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
02	-	-	02	40	60	-	-	-	100

Prerequisites:

Fundamentals of C/C++ programming and data structures.
Basic knowledge of computer organization and operating systems.
Understanding of discrete mathematics, especially sets, relations, graphs, and logic.
Basic familiarity with formal languages and automata is desirable.

Course Objectives: The course aims to enable students to:

- To understand system programming and compiler design.
- To apply compiler analysis and intermediate code techniques.
- To apply optimization and modern compiler technologies.

Course Outcomes: After completion of this course, students will be able to -

CO1	**Explain** the architecture and working of system software and different phases of a compiler.
CO2	**Analyze** the working of two-pass assemblers, macro processors, linkers, and loaders.
CO3	**Apply** regular expressions, CFGs, FIRST/FOLLOW, LL(1), and shift-reduce techniques to construct lexical analyzers and parsers.
CO4	**Generate** syntax-directed intermediate representations using three-address code, quadruples, triples, syntax trees, and DAGs.
CO5	**Apply and analyze** runtime storage concepts and code optimization techniques to improve program efficiency.
CO6	**Apply** basic code generation techniques and **evaluate** modern compiler technologies such as GCC, LLVM, and JIT compilation.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to System Programming and Compiler Design:** System software and application software, language processing system, assembler, compiler, interpreter, linker and loader. **Phases of compiler**, front-end and back-end, compiler construction tools, bootstrapping and cross compiler.	4
2.	**Assemblers, Macro Processors and Linkers: Assembler:** Assembly language, assembler directives, symbol table, literal table, forward reference, **two-pass assembler**. **Macro Processor:** Macro definition, invocation and expansion. **Linker and Loader:** Linking, relocation and loading.	5
3.	**Lexical and Syntax Analysis**: **Lexical Analysis:** Tokens, patterns, lexemes, regular expressions, finite automata, lexical analyzer and LEX/FLEX. **Syntax Analysis:** Context-Free Grammar, parse tree, FIRST and FOLLOW, recursive descent, **LL(1) and shift-reduce parsing**, overview of LR/SLR/LALR.	7
4.	Semantic Analysis and Intermediate Code Generation: **Semantic Analysis:** Syntax-directed definitions, type checking and symbol table. **Intermediate Code:** Three-address code, quadruples, triples, syntax trees and DAG.	4

5.	**Runtime Environment and Code Optimization:** Runtime storage, **activation records, stack and heap**. Basic blocks and flow graphs. **Code optimization:** constant folding, common subexpression elimination, dead-code elimination, loop optimization and peephole optimization.	3
6.	**Code Generation and Modern Compiler Technologies: Code Generation:** Instruction selection and register allocation. **Modern Technologies:** GCC, LLVM, **JIT compilation** and applications of modern compilers.	3
	TOTAL	**26**

Textbooks :

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D. Ullman, Compilers: Principles, Techniques, and Tools, 2nd Edition, Pearson.
2. John R. Levine, Tony Mason and Doug Brown, Lex & Yacc, O'Reilly Media.
3. D. M. Dhamdhere, Systems Programming and Operating Systems, McGraw-Hill.
4. John J. Donovan, Systems Programming, McGraw-Hill.

Reference Books :

1. Alfred V. Aho and Jeffrey D. Ullman, *Principles of Compiler Design*, Addison-Wesley.
2. Steven S. Muchnick, *Advanced Compiler Design and Implementation*, Morgan Kaufmann.
3. Keith D. Cooper and Linda Torczon, *Engineering a Compiler*, Morgan Kaufmann.
4. Andrew W. Appel, *Modern Compiler Implementation in C*, Cambridge University Press.
5. D. M. Dhamdhere, *System Programming and Operating System*, Tata McGraw-Hill.

E-Resources :

1. https://nptel.ac.in/courses/106105190
2. https://market.tutorialspoint.com/course/compiler-design-online-training/index.asp

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)							**Semester:** VI		
Course: Program Elective-II (System Programming and Compiler Design) Lab							**Code:** ITPE602CP		
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
-	02	-	01	-	-	-	25	-	25

Prerequisites:

Fundamental knowledge of C/C++ programming and data structures, Computer organization and architecture.

Course Objectives: The course aims to enable students to:

1. To understand the **structure and phases of compiler design**.
2. To implement **assemblers, macro processors, and lexical analyzers**.
3. To apply **parsing techniques** for context-free grammars.
4. To generate and represent **intermediate code**.
5. To apply **code optimization techniques** for improving compiler output.
6. To understand and implement basic **code generation and compilation tools**

Course Outcomes: After completion of this course, students will be able to -

CO1	**Explain** the phases and components of a compiler and assembler.
CO2	**Implement** lexical analyzers, assemblers, and macro processors for language processing.
CO3	**Apply** parsing techniques such as LL(1), recursive descent, and shift-reduce parsing.
CO4	**Generate** intermediate representations such as three-address code, quadruples, and triples.
CO5	**Analyze** basic blocks, flow graphs, and apply code optimization techniques.
CO6	**Develop** basic code generation and register allocation solutions using compiler tools such as GCC/LLVM.

Laboratory Assignment (any 10):

1. Study the structure and phases of a compiler and demonstrate the compilation process of a simple program.
2. Implement a two-pass assembler with symbol table and literal table generation.
3. Implement a macro processor for macro definition, invocation, and expansion.
4. Implement a lexical analyzer to identify tokens, keywords, identifiers, operators, and literals.
5. Design a lexical analyzer using LEX/FLEX.
6. Compute FIRST and FOLLOW sets for a given Context-Free Grammar.
7. Implement an LL(1) parser / recursive descent parser.
8. Implement a shift-reduce parser for a given grammar.
9. Generate three-address code for arithmetic and conditional expressions.
10. Generate and represent quadruples and triples from intermediate code.
11. Identify basic blocks and flow graphs and apply code optimization techniques such as constant folding, common subexpression elimination, and dead-code elimination.
12. Demonstrate basic code generation/register allocation and study compilation using GCC/LLVM or a simple JIT compilation example.

DEPARTMENT OF INFORMATION TECHNOLOGY

Textbooks :

5. Alfred V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D. Ullman, Compilers: Principles, Techniques, and Tools, 2nd Edition, Pearson.
6. John R. Levine, Tony Mason and Doug Brown, Lex & Yacc, O'Reilly Media.
7. D. M. Dhamdhere, Systems Programming and Operating Systems, McGraw-Hill.
8. John J. Donovan, Systems Programming, McGraw-Hill.

Reference Books :

6. Alfred V. Aho and Jeffrey D. Ullman, *Principles of Compiler Design*, Addison-Wesley.
7. Steven S. Muchnick, *Advanced Compiler Design and Implementation*, Morgan Kaufmann.
8. Keith D. Cooper and Linda Torczon, *Engineering a Compiler*, Morgan Kaufmann.
9. Andrew W. Appel, *Modern Compiler Implementation in C*, Cambridge University Press.
10. D. M. Dhamdhere, *System Programming and Operating System*, Tata McGraw-Hill.

E-Resources :

3. https://nptel.ac.in/courses/106105190
4. https://market.tutorialspoint.com/course/compiler-design-online-training/index.asp

Program Elective Course III

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)							**Semester:** VI		
Course: Program Elective Course - III **(**Big Data Analytics)							**Code:** ITPE603A		
Teaching Scheme (Hrs/week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	-	-	03	40	60	-	-	-	100

Prerequisites:

Basic knowledge of Programming using Python/C, Database Management Systems, Data Structures and Algorithms, Probability and Statistics, Computer Networks, Operating Systems, and fundamentals of Data Science.

Course Objectives:

1. To introduce the concepts, characteristics, and ecosystem of Big Data Analytics.
2. To understand the mathematical and statistical foundations required for processing large-scale data. SSSS
3. To learn Hadoop architecture, distributed processing, and NoSQL technologies for handling Big Data.
4. To perform data collection, preprocessing, analytics, and visualization using modern Big Data tools.
5. To develop skills in business intelligence, dashboard creation, and data-driven decision making.
6. To explore real-world applications, emerging trends, ethical issues, and security challenges in Big Data Analytics.

Course Outcomes: After completion of this course, students will be able to -

CO1	Explain Big Data concepts, architectures, applications, and challenges in modern industries.
CO2	Apply mathematical, probabilistic, and statistical techniques for Big Data processing and analysis.
CO3	Implement Hadoop ecosystem tools, including HDFS, MapReduce, and NoSQL databases, for distributed data processing.
CO4	Perform data collection, preprocessing, transformation, analytics, and querying using Big Data tools and technologies.
CO5	Design effective visualizations, dashboards, and business intelligence solutions using modern visualization platforms.
CO6	Analyze real-world Big Data applications, emerging trends, ethical concerns, and security challenges in Big Data Analytics.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to Big Data and Data Science:** Introduction to Big Data, Characteristics of Big Data (Volume, Velocity, Variety, Veracity), Data Science Lifecycle, Big Data Applications in Industry, Big Data Challenges, Data Warehousing, Shared Nothing and Shared Everything Architectures, Overview of Hadoop Ecosystem.	6
2.	**Mathematical Foundations for Big Data:** Probability Basics, Conditional Probability, Markov Chains, Random Variables, Hashing Techniques, Bloom Filters, Approximate Counting Algorithms, Data Streaming Models, Statistical Concepts: Mean, Median, Mode, Variance, Standard Deviation and Correlation Analysis.	6

3.	**Hadoop and Distributed Data Processing:** Hadoop Architecture, HDFS Architecture, Name Node and Data Node, Hadoop Cluster Setup, Hadoop Shell Commands, File Read and Write Operations, MapReduce Framework, Job Scheduling, Introduction to NoSQL Databases and MongoDB, ETL Processing Concepts	7
4.	**Big Data Analytics and Data Processing:** Big Data Analytics Lifecycle, Data Collection from CSV, JSON, Excel, MySQL and MongoDB, Data Cleaning and Data Transformation, Handling Missing Values, Data Standardization, Feature Engineering, Statistical Analysis, Data Visualization Basics, Hive Query Language (HQL).	7
5.	**Big Data Visualization and Business Intelligence:** Introduction to Data Visualization, Challenges in Big Data Visualization, Visualization Techniques, Tableau Fundamentals, Dashboard Design, D3.js Basics, Google Charts API, Business Intelligence Applications, Case Study using Zomato or E-commerce Dataset.	7
6.	**Applications of Big Data Analytics and Emerging Trends:** Social Media Analytics, Text Mining, Mobile Analytics, Recommendation Systems, Big Data in Healthcare, Finance and Retail, Big Data Strategy, Decision Theory, Ethical Issues in Big Data, Big Data Security Challenges, Research Directions in Big Data Analytics.	6
	TOTAL	**39**

Textbooks:

1. Big Data Analytics – Raj Kamal, McGraw-Hill Education.
2. Big Data: Principles and Best Practices of Scalable Real-Time Data Systems – Manning Publications.
3. Hadoop: The Definitive Guide – O'Reilly Media.
4. Data Science for Business – O'Reilly Media.

Reference Books:

1. Mining of Massive Datasets – Cambridge University Press.
2. NoSQL Distilled – Addison-Wesley.
3. Practical Statistics for Data Scientists – O'Reilly Media.
4. Data Visualization with Python and JavaScript – Packt Publishing.
5. Business Intelligence Guidebook – Morgan Kaufmann.

E-Resources:

1. https://hadoop.apache.org/docs
2. https://learn.mongodb.com/
3. https://www.tableau.com/learn
4. https://developers.google.com/chart
5. https://nptel.ac.in/courses

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)							**Semester:** VI		
Course: Program Elective Course - III **(**Big Data Analytics) Lab.							**Code:** ITPE603AP		
Teaching Scheme (Hrs/week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
-	02	-	01	-	-	-	-	25	25

Prerequisites:

Basic knowledge of Programming using Python/C, Database Management Systems, Data Structures and Algorithms, Probability and Statistics, Computer Networks, Operating Systems, and fundamentals of Data Science.

Course Objectives:

1. To develop skills in data generation and analysis using Python.
2. To apply probability, statistics, hashing, and Bloom Filters.
3. To understand Hadoop HDFS and MapReduce for Big Data processing.
4. To use MongoDB for NoSQL data storage and processing.
5. To perform data collection, preprocessing, and visualization.
6. To develop dashboards and end-to-end data analytics projects.

Course Outcomes: After completion of this course, students will be able to -

CO1	Analyze datasets using **Python, NumPy, and Pandas**.
CO2	Implement **hashing and Bloom Filters** for efficient data search.
CO3	Perform Big Data processing using **Hadoop HDFS and MapReduce**.
CO4	**Apply** MongoDB operations for NoSQL data storage and CRUD processing.
CO5	**Analyze** and visualize data after performing data collection and preprocessing.
CO6	**Create** dashboards and end-to-end analytics projects to derive insights from real-world datasets.

Laboratory Assignments: (Any 10)

1. Develop a Python program to generate large datasets and perform basic data analysis using NumPy and Pandas.
2. Develop Python programs to perform probability calculations and conditional probability analysis using sample datasets.
3. Write Python programs to calculate statistical measures such as mean, median, mode, variance, standard deviation, and correlation analysis.
4. Implement hashing techniques and Bloom Filters using Python for efficient searching and membership testing.
5. Install Hadoop and perform basic HDFS operations such as file creation, upload, download, copy, and deletion using Hadoop shell commands.
6. Implement a MapReduce program such as Word Count or Frequency Count to process large text datasets.
7. Create a MongoDB database and perform CRUD (Create, Read, Update, Delete) operations on sample datasets.
8. Develop programs to collect and load data from CSV, JSON, Excel, and database sources using Python.
9. Perform data preprocessing tasks such as handling missing values, removing duplicates, data transformation, and standardization using Python.

10. Create visualizations such as bar charts, histograms, pie charts, scatter plots, and correlation plots using Python libraries.
11. Create interactive dashboards and business intelligence reports using real-world datasets such as sales, restaurant, or e-commerce data.
12. Develop a mini analytics project using real-world datasets to perform data collection, preprocessing, analysis, visualization, and generate useful insights.

Textbooks:

1. Big Data Analytics – Raj Kamal, McGraw-Hill Education.
2. Big Data: Principles and Best Practices of Scalable Real-Time Data Systems – Manning Publications.
3. Hadoop: The Definitive Guide – O'Reilly Media.
4. Data Science for Business – O'Reilly Media.

Reference Books:

1. Mining of Massive Datasets – Cambridge University Press.
2. NoSQL Distilled – Addison-Wesley.
3. Practical Statistics for Data Scientists – O'Reilly Media.
4. Data Visualization with Python and JavaScript – Packt Publishing.
5. Business Intelligence Guidebook – Morgan Kaufmann.

E-Resources:

1. https://hadoop.apache.org/docs
2. https://learn.mongodb.com/
3. https://www.tableau.com/learn
4. https://developers.google.com/chart
5. https://nptel.ac.in/courses

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)							**Semester:** VI		
Course: Program Elective-III **(**Cryptography and Network Security)							**Code:** ITPE603B		
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	-	-	03	40	60	-	-	-	100

Prerequisites: Students should have basic knowledge

- Basic knowledge of Computer Networks
- Fundamental understanding of Computer Systems and Operating Systems
- Familiarity with basic mathematical concepts such as modular arithmetic and logic (desirable)

Course Objectives:

1. Understand the fundamental concepts of information security, threats, vulnerabilities, and security services.
2. Learn the principles and techniques of classical and symmetric key cryptography.
3. Understand public key cryptography, key exchange mechanisms, and key management techniques.
4. Examine authentication mechanisms, cryptographic hash functions, digital signatures, and Public Key Infrastructure (PKI).
5. Explore network security protocols and technologies used to secure communication networks.
6. Analyze emerging security technologies, cybersecurity governance, risk management, and current security challenges

Course Outcomes: After completion of this course, students will be able to -

CO1	Explain the fundamental concepts of information security and identify various security threats and attacks.
CO2	Apply the principles of classical and symmetric cryptographic techniques to secure information.
CO3	Analyze public key cryptography algorithms and key management approaches used in secure communication.
CO4	Evaluate authentication mechanisms, digital signatures, hash functions, and PKI frameworks.
CO5	Describe network security protocols, firewalls, VPNs, and intrusion detection mechanisms used to protect networks.
CO6	Assess security issues in cloud, wireless, and emerging computing environments and recommend appropriate security controls.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to Information Security:** Information Security Concepts Security Goals: Confidentiality, Integrity, Availability (CIA), Threats, Vulnerabilities, and Attacks. Security Services and Mechanisms, Security Models and Challenges	6

2.	**Classical and Symmetric Key Cryptography:** Principles of Cryptography Classical Ciphers: Caesar, Playfair, Hill Cipher, Symmetric Key Encryption, Block and Stream Ciphers, DES and AES Overview, Advantages and Limitations of Symmetric Cryptography	7
3.	**Public Key Cryptography and Key Management:** Asymmetric Cryptography Concepts, Public and Private Keys, RSA Algorithm, Diffie-Hellman Key Exchange, Elliptic Curve Cryptography (ECC), Key Distribution and Key Management	7
4.	**Authentication and Digital Signatures:** Cryptographic Hash Functions, Message Authentication Codes (MAC), Digital Signatures, Public Key Infrastructure (PKI), Digital Certificates, Authentication Protocols	7
5.	**Network Security Protocols and Mechanisms:** Network Security Fundamentals, Firewalls and Intrusion Detection Systems, Virtual Private Networks (VPNs), Secure Socket Layer (SSL), Transport Layer Security (TLS), Secure Email and Web Security	6
6.	**Emerging Trends and Security Management:** Wireless and Mobile Security, Cloud Security Concepts, Blockchain Security Fundamentals, Cybersecurity Governance, Risk Management and Security Policies, Emerging Security Challenges and Trends	6
	TOTAL	**39**

Textbooks:

1. Cryptography and Network Security: Principles and Practice, Pearson Education, Latest Edition.
2. Cryptography and Network Security, McGraw-Hill Education, Latest Edition.
3. Network Security Essentials: Applications and Standards, Pearson Education, Latest Edition.

Reference Books:

1. Computer Security: Principles and Practice, Pearson Education.
2. Cryptography and Network Security, McGraw-Hill Education.
3. Network Security Bible, Wiley Publications.
4. Security in Computing, Pearson Education.
5. Introduction to Modern Cryptography, Chapman & Hall/CRC.

E-Resources:

1. https://nptel.ac.in/
2. https://swayam.gov.in/
3. https://www.nist.gov/

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)								**Semester:** VI	
Course: Program Elective-III **(**Cryptography and Network Security) Lab.								**Code:** ITPE603BP	
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
-	02	-	01	-	-	-	-	25	100

Prerequisites: Students should have basic knowledge

- Basic Python programming and programming logic.
- Basic computer networking concepts and protocols.
- Fundamental concepts of databases and web applications.

Course Objectives:

1. To introduce students to the fundamental concepts and principles of information security.
2. To develop practical understanding of authentication, authorization, data protection, and activity logging.
3. To implement and analyze classical and modern cryptographic techniques.
4. To understand password security, hashing, and secure authentication mechanisms.
5. To develop skills in network traffic monitoring and security analysis using appropriate tools.
6. To identify security risks in cloud-based applications and develop appropriate mitigation strategies.

Course Outcomes: After completion of this course, students will be able to -

CO1	Develop a basic **secure information system** implementing authentication, role-based access control, data protection, and activity logging.
CO2	Implement and analyze **classical cryptographic algorithms** such as Caesar, Playfair, and Hill ciphers for encryption and decryption.
CO3	Implement **RSA public-key cryptography** and demonstrate key generation, encryption, and decryption.
CO4	Implement secure **password hashing and authentication mechanisms** and differentiate between hashing and encryption.
CO5	Capture and analyze **network traffic** using Wireshark and identify common network protocols and secured/unsecured communication.
CO6	Perform a basic **cloud security risk assessment** and recommend suitable security controls and mitigation strategies.

Laboratory Assignment:

1. Develop a simple Secure Student Information System demonstrating login authentication, role-based access, data protection, and activity logging.
2. Implementation of Classical Ciphers – Implement Caesar Cipher, Playfair Cipher, and Hill Cipher using Python. Provide both encryption and decryption functionality.
3. RSA Encryption and Decryption – Implement RSA key generation, encryption, and decryption for a small demonstration message. Explain the role of public and private keys.
4. Password Hashing and Authentication System **–** Develop a login system that stores passwords using a secure cryptographic hash function rather than plaintext. Demonstrate password verification and discuss hashing vs encryption
5. Network Traffic Analysis using Wireshark – Capture and analyze network traffic using Wireshark. Identify protocols such as HTTP, DNS, TCP, and TLS, and compare characteristics of unsecured and secured communication.

6. Cloud Security Risk Assessment – Select a cloud-based application/service and identify possible security risks related to data privacy, authentication, access control, storage, and availability. Prepare a risk assessment and mitigation plan.

Textbooks:

1. Cryptography and Network Security: Principles and Practice, Pearson Education, Latest Edition.
2. Cryptography and Network Security, McGraw-Hill Education, Latest Edition.
3. Network Security Essentials: Applications and Standards, Pearson Education, Latest Edition.

Reference Books:

1. Computer Security: Principles and Practice, Pearson Education.
2. Cryptography and Network Security, McGraw-Hill Education.
3. Network Security Bible, Wiley Publications.
4. Security in Computing, Pearson Education.
5. Introduction to Modern Cryptography, Chapman & Hall/CRC.

E-Resources:

1. https://nptel.ac.in/
2. https://swayam.gov.in/
3. https://www.nist.gov/

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)							**Semester:** VI		
Course: Program Elective Course - I (Wireless Communication)							**Code:** ITPE603C		
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	-	-	03	40	60	-	-	-	100

Prerequisites: Students should have basic knowledge

- Computer Networks
- Digital Communication Fundamentals
- Basic Signals and Systems

Course Objectives:

1. To understand the fundamentals of wireless communication systems and propagation mechanisms.
2. To study modern wireless transmission and multiple access techniques.
3. To analyse cellular network architectures and mobile communication technologies.
4. To understand wireless LAN, PAN, and emerging wireless technologies.
5. To examine security and quality of service issues in wireless networks.
6. To explore next-generation wireless communication systems, including 5G and 6G.

Course Outcomes: After completion of this course, students will be able to -

CO1	Explain wireless communication principles and propagation characteristics.
CO2	Analyze wireless transmission and multiple access techniques.
CO3	Describe cellular network architectures and mobility management mechanisms.
CO4	Evaluate wireless networking technologies and standards.
CO5	Assess wireless security challenges and quality of service mechanisms.
CO6	Examine emerging trends including 5G, IoT, satellite communication, and 6G networks.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Fundamentals of Wireless Communication:** Introduction to Wireless Communication, Evolution of Wireless Systems, Electromagnetic Spectrum, Wireless Channel Characteristics, Radio Wave Propagation, Path Loss Models, Fading and Shadowing, Antennas and Signal Transmission.	4
2.	**Wireless Transmission Techniques:** Multiple Access Techniques-FDMA, TDMA, CDMA, OFDMA, MIMO Systems, Beamforming Fundamentals.	4
3.	**Cellular Communication Systems:** Cellular Concept, Frequency Reuse, Handoff Mechanisms, Mobility Management, LTE Architecture, LTE-Advanced, Cellular Network Planning, Performance Metrics	5

4.	**Wireless Networks and Standards:** Wireless PAN (Bluetooth, ZigBee), Wireless MAN, Wi-Fi, , LoRaWAN , Wireless Sensor Networks.	4
5.	**Wireless Security and QoS:** Wireless Network Security Challenges, Authentication and Access Control, WPA2 and WPA3, Wireless Threats and Attacks, Quality of Service (QoS).	5
6.	**Next Generation Wireless Systems:** Edge Computing in Wireless Networks, Satellite and Non-Terrestrial Networks, Introduction to 6G, AI for Wireless Networks, Future Trends	4
	TOTAL	**26**

Textbooks:

1. Andrea Goldsmith, **"Wireless Communications"**, Cambridge University Press.
5. Theodore S. Rappaport, **"Wireless Communications: Principles and Practice"**, Pearson Education.
6. William Stallings, **"Wireless Communications and Networks"**, Pearson Education.

Reference Books:

1. Andreas F. Molisch, "Wireless Communications", Wiley Publications.
2. Vijay K. Garg, "Wireless Communications and Networking", Morgan Kaufmann.
3. Jochen Schiller, "Mobile Communications", Pearson Education.
4. Mischa Schwartz, "Mobile Wireless Communications", Cambridge University Press.
5. David Tse and Pramod Viswanath, "Fundamentals of Wireless Communication", Cambridge University Press.

E-Resources:

1. https://nptel.ac.in/
2. https://www.gsma.com/solutions-and-impact/connectivity-for-good/mobile-for-development/impact-report/

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)									**Semester:** VI
Course: Program Elective Course - I (Wireless Communication)									**Code:** ITPE603CP
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
-	02	-	01	-	-	-	25	-	25

Prerequisites: Students should have basic knowledge

- Digital communication fundamentals and data transmission.
- Basic concepts of computer networks and wireless networks.

Course Objectives:

1. To introduce students to the fundamental concepts and components of wireless communication systems.
2. To develop an understanding of electromagnetic spectrum, frequency bands, and wireless applications.
3. To analyse wireless propagation and path loss using mathematical models and simulation.
4. To understand and implement basic multiple access techniques such as FDMA, TDMA, and CDMA.
5. To study cellular communication concepts, including frequency reuse and handoff mechanisms.
6. To introduce wireless network security, performance analysis, and practical wireless communication system design.

Course Outcomes: After completion of this course, students will be able to -

CO1	Identify and explain the **basic components and working principles of wireless communication systems**.
CO2	Analyze different **frequency bands, wireless technologies, and their applications**.
CO3	Calculate and simulate **path loss and wireless channel characteristics** for different communication conditions.
CO4	Implement and compare **FDMA, TDMA, and CDMA** techniques for multiple-user communication.
CO5	Simulate basic **cellular handoff mechanisms** and analyze wireless network performance based on signal strength.
CO6	Configure and evaluate a basic **secure wireless network** and design a simple wireless communication system

Laboratory Assignment:

1. Identify and explain the basic components of a wireless communication system. Develop a simple block diagram showing transmitter, wireless channel, and receiver.
2. Study different frequency bands used in wireless communication and identify their applications such as Wi-Fi, Bluetooth, cellular communication, and satellite communication.
3. Write a program to calculate free-space path loss for different distances and frequencies. Plot **Path Loss vs. Distance**.
4. Implement a simple example of **FDMA, TDMA, and CDMA** and demonstrate how multiple users can share a wireless communication channel.
5. Simulate a mobile user moving from one cell to another and demonstrate the basic handoff process based on signal strength.
6. Configure a basic wireless network and study authentication and security mechanisms such as **WPA2/WPA3**.
7. Design a small wireless communication system and demonstrate data transmission between multiple wireless nodes. Measure basic performance parameters.

DEPARTMENT OF INFORMATION TECHNOLOGY

Textbooks:

1. Andrea Goldsmith, **"Wireless Communications"**, Cambridge University Press.
2. Theodore S. Rappaport, **"Wireless Communications: Principles and Practice"**, Pearson Education.
3. William Stallings, **"Wireless Communications and Networks"**, Pearson Education.

Reference Books:

1. Andreas F. Molisch, "Wireless Communications", Wiley Publications.
2. Vijay K. Garg, "Wireless Communications and Networking", Morgan Kaufmann.
3. Jochen Schiller, "Mobile Communications", Pearson Education.
4. Mischa Schwartz, "Mobile Wireless Communications", Cambridge University Press.
5. David Tse and Pramod Viswanath, "Fundamentals of Wireless Communication", Cambridge University Press.

E-Resources:

1. https://nptel.ac.in/
2. https://www.gsma.com/solutions-and-impact/connectivity-for-good/mobile-for-development/impact-report/

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)							**Semester:** VI		
Course: Internet of Things (IoT)							**Code:** ITMD604		
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
02	-	-	02	40	60	-	-	-	100

Prerequisites:

1. Basics of Computer Networks 2. Processor Architecture (Computer Organization)

Course Objectives:

1. To understand the fundamentals of IoT
2. To implement small IoT systems using building blocks of IoT
3. To gain knowledge of IoT protocols
4. To comprehend the fundamentals of security in IoT
5. To learn how secure infrastructure for IoT is supported by IoT platforms
6. To learn real-world application scenarios of IoT along with its societal and economic impact using case studies

Course Outcomes: After completion of this course, students will be able to -

CO1	Demonstrate the fundamentals of IoT
CO2	Understand Things in IoT
CO3	Comprehend the IoT protocols
CO4	Apply concepts of IoT to build small IoT Systems
CO5	Analyze the security issues in IoT and various IoT platforms
CO6	Design the real-world applications of IoT

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Fundamentals of IoT:** Definition and characteristics of IoT, Internet of Things: Emerging Trends, Economic Significance, Technical Building Blocks, Physical design of IoT, Things of IoT, IoT Protocols, Logical design of IoT, IoT functional blocks, IoT communication models, IoT enabling technologies, IoT levels and deployment templates, IoT Network Issues and Challenges	4
2.	**Introduction to IoT Sensors & Actuators**: Introduction to IoT platforms, Sensors, Actuators, and Smart Objects, Classification of things, Sensor Networks, Connecting Smart Objects: Communications Criteria, IoT Access Technologies: IEEE 802.15.4	4
3.	**IoT Communication Protocols:** Existing protocols for M2N (TCP/IP) IoT Protocols Organization, IoT Protocols overview: CoAP, MQTT, IPv6, ZigBee, Bluetooth, Wi-fi, Comparison of Traditional Networking Protocols and IoT Protocols, Issues with IoT Standardization.	4

4.	**IoT & M2M:** Machine-to-Machine, Difference between IoT and M2M, Software-defined Network, Software-defined Network for IoT, IoT Physical Devices and Endpoints: Basic building blocks and IoT device, Interoperability in IoT, Introduction to Arduino Programming, Integration of Sensors and Actuators with microcontroller.	6
5.	**Microcontrollers and IoT Security:** Microcontrollers: Arduino, Raspberry Pi, ESP8266, ESP32 and interfaces. IoT Security: Vulnerabilities of IoT, Security Requirements, Challenges for Secure IoT, Key elements of IoT Security, Introduction to Cloud Computing, Cloud of Things: Grid/SOA and Cloud Computing	6
6.	**IoT in daily Use:** Home Intrusion Detection, Weather Monitoring System, Air Pollution Monitoring, Smart Irrigation, Smart Cities, Health Care IoT design, Applications of Raspberry Pi and Arduino architecture.	6
	TOTAL	**36**

Textbooks:

1. Internet of Things: A Hands-On Approach Arshdeep Bahga, Vijay Madisetti VPT – Paperback 2015 978- 0996025515 628/- 2.
2. IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things David Hanes, Gonzalo Salgueiro, Patrick Grossetete Cisco Press – Paperback – 16 Aug 2017 978-1- 58714-456- 1 599.

Reference Books:

1. Smart Internet of Things Projects Agus Kurniawan Packt - Sep 2016 978- 1- 78646- 651-8
2. The Internet of Things Key Olivier Willy Publication 2nd Edition 978
3. Applications and protocols Hersent s 119- 99435-0,
4. The Internet of Things Connecting Objects to the Web Hakima Chaouchi, Willy Publications 978-1-84821- 140-7.

E-Resources:

1. https://developer.ibm.com/technologies/iot/?utm_source=chatgpt.com
2. https://docs.arduino.cc/iot/?utm_source=chatgpt.com

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)							**Semester:** VI		
Course: Internet of Things (IoT)							**Code:** ITMD604P		
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
-	02	-	01	-	-	-	25	-	25

Prerequisites:

1. Basics of Computer Network 2. Processor Architecture (Computer Organization)

Course Objectives:

Computer fundamentals and operating systems, C/C++ or Python programming, Basic digital electronics and microcontrollers.

Course Outcomes: After completion of this course, students will be able to -

CO1	To introduce students to **Raspberry Pi, BeagleBoard, Arduino, and ESP8266** as IoT prototyping platforms.
CO2	To understand the **operating systems, installation, and configuration** of IoT development boards.
CO3	To develop skills in **GPIO programming and peripheral interfacing**.
CO4	To implement basic programs for **controlling LEDs, buzzers, and other devices**.
CO5	To understand and implement **client-server communication** using IoT platforms.
CO6	To develop simple **web-based applications for remote/local control of IoT devices**.

List of Experiments

1. Study of Raspberry Pi, Beagle Board, Arduino, and ESP8266.
2. Study of different operating systems for Raspberry Pi/Beagle board/Arduino. Understanding the process of OS installation on Raspberry Pi/Beagle board/Arduino
3. Study of Connectivity and configuration of Raspberry Pi/Beagle board circuit with basic Peripherals, LEDS. Understanding GPIO and its use in programming
4. Use an open-source prototype platform- Raspberry Pi/Beagle board/Arduino and write a simple program to blink the LED
5. Demonstrate client applications to get services from the server application using Raspberry Pi/Beagle board
6. Create a simple web interface for Raspberry Pi to control the connected LEDs remotely through the interface
7. Create a web application to control the on/off state of the buzzer on localhost

Textbooks:

1. Internet of Things: A Hands-On Approach Arshdeep Bahga, Vijay Madisetti VPT – Paperback 2015 978- 0996025515 628/- 2.
2. IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things David Hanes, Gonzalo Salgueiro, Patrick Grossetete Cisco Press – Paperback – 16 Aug 2017 978-1- 58714-456- 1 599.

DEPARTMENT OF INFORMATION TECHNOLOGY

Reference Books:

1. Smart Internet of Things Projects Agus Kurniawan Packt - Sep 2016 978- 1- 78646- 651-8
2. The Internet of Things Key Olivier Willy Publication 2nd Edition 978
3. Applications and protocols Hersent s 119- 99435-0,
4. The Internet of Things Connecting Objects to the Web Hakima Chaouchi, Willy Publications 978-1-84821- 140-7.

E-Resources:

1. https://developer.ibm.com/technologies/iot/?utm_source=chatgpt.com
2. https://docs.arduino.cc/iot/?utm_source=chatgpt.com

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)							**Semester:** VI		
Course: Programming Laboratory – III (Software Testing Laboratory)							**Code:** ITVS605		
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
-	04	-	02	-	-	25	-	50	75

Prerequisites: Students should have basic knowledge

- Software Engineering and Software Development Life Cycle (SDLC).
- Basic understanding of software quality assurance concepts.
- Web Technologies (HTML, CSS, JavaScript).
- Programming using C, C++, Java, or Python.

Course Objectives:

1. Understand the fundamentals of software testing and software quality assurance.
2. Design and execute manual test cases using standard testing techniques.
3. Perform functional and non-functional testing of software applications.
4. Develop automated test scripts using Selenium WebDriver.
5. Apply API testing, unit testing, and test reporting techniques.
6. Use industry-standard testing tools and frameworks for software validation

Course Outcomes: After completion of this course, students will be able to -

CO1	Design software test cases using black-box and white-box testing techniques.
CO2	Execute manual testing and report software defects using standard templates.
CO3	Develop automated test scripts using Selenium WebDriver.
CO4	Perform unit, web, and API testing using appropriate testing frameworks.
CO5	Generate test reports and evaluate software quality using testing metrics.
CO6	Apply software testing methodologies in real-world software development projects.

List of Experiments:

1. Study the Software Requirement Specification (SRS) of a web application (Library Management, Student Management, or Online Shopping System) and identify functional requirements. Prepare test scenarios for the selected module.
2. Design comprehensive test cases for the Login and Registration module using
   - Equivalence Partitioning
   - Boundary Value Analysis
3. Analyze a given program and perform
   - Statement Coverage
   - Branch Coverage
   - Path Coverage Generate suitable test cases.
4. Execute the prepared test cases on a sample web application and prepare
   - Bug Report
   - Severity
   - Priority

Defect Life Cycle using a standard bug report format.

5. Perform functional testing on an Online Shopping System.
   - Validate
   - Login
   - Search
   - Add to Cart
   - Checkout
   - Payment
6. Prepare
   - Test Plan
   - Test Strategy
   - Test Summary Report
   - for an assigned application.
7. Install Selenium, configure WebDriver, and automate
   - Browser Launch
   - URL Navigation
   - Browser Close
   - using Java/Python.
8. Automate a Login Page using Selenium.
   - Perform
   - Textbox
   - Radio Button
   - Checkbox
   - Dropdown
   - Button Click
   - Assertions
9. Develop Selenium scripts to identify web elements using
   - ID
   - Name
   - XPath
   - CSS Selector
   - Perform automated login validation.
10. Implement Data Driven Testing by reading multiple login credentials from an Excel file or CSV file and automate login validation.
11. Test REST APIs using Postman.
    - Perform
    - GET
    - POST
    - PUT
    - DELETE
      Validate
    - Response Code
    - Response Time
    - JSON Response

DEPARTMENT OF INFORMATION TECHNOLOGY

Textbooks:

1. Software Testing: Principles and Practices, Srinivasan Desikan and Gopalaswamy Ramesh, Pearson Education.
2. Foundations of Software Testing, Cengage Learning.
3. Software Testing, Sams Publishing.
4. Selenium WebDriver Practical Guide, Packt Publishing.

Reference Books:

1. Software Testing and Analysis: Process, Principles and Techniques, Wiley India.
2. Effective Methods for Software Testing, Wiley Publications.
3. Lessons Learned in Software Testing, Wiley.
4. How Google Tests Software, Addison-Wesley.
5. The Art of Software Testing, Wiley India.
6. Agile Testing, Addison-Wesley.

E-Resources:

1. https://www.selenium.dev/documentation
3. https://docs.pytest.org
4. https://nptel.ac.in
5. https://www.istqb.org

DEPARTMENT OF INFORMATION TECHNOLOGY

Program: B. Tech. (Information Technology)									**Semester:** VI
Course: Internship - V									**Code:** ITIN605
Teaching Scheme				**Evaluation Scheme**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
4 Week			02		-	25	-	-	25
Kindly refer to the policy of internship on the College Website									