Shri Vile Parle Kelavani Mandal's

Dwarkadas J. Sanghvi College of Engineering

(Autonomous College Affiliated to the University of Mumbai)

Course Structure and Syllabus

of

Third Year Engineering

in

Computer Engineering

Prepared by:- Board of Studies in Computer Engineering

Recommended by:- Academic Council of D. J. Sanghvi College of Engineering

Approved by:- Governing Body of D. J. Sanghvi College of Engineering

Revision: 3 (2023)

With effect from the Academic Year: 2025-2026

Scheme for Second Year Undergraduate Program in Computer Engineering (Autonomous) (DJS23 Scheme)

SEMESTER V

Sr. No.	Course Code	Course	Teaching Scheme				Semester End Examination (A)						Continuous Assessment						Aggregate (A+B)	Credits Earned	
			Theory (hrs)	Practical (hrs)	Tutorial (hrs)	Total Credits	Duration (hrs)	Theory	Oral	Pract.	Oral & Practical	SEE Total (A)	TT 1	TT 2	TT 3	Total	Term Work Total	Total (B)			
1	DJS23CCPC501	Data Warehousing and Mining	3			3	2	60				60	15	15	10	40		40	100	3	4
	DJS23CLPC501	Data Warehousing and Mining Laboratory		2		1	2		25			25					25	25	50	1	
2	DJS23CCPC502	Artificial Intelligence	3			3	2	60				60	15	15	10	40		40	100	3	4
	DJS23CLPC502	Artificial Intelligence Laboratory		2		1	2		25			25					25	25	50	1	
3	DJS23CCPC503	Automata Theory and Compiler Design	3			3	2	60				60	15	15	10	40		40	100	3	4
	DJS23CLPC503	Automata Theory and Compiler Design Laboratory		2		1	2		25			25					25	25	50	1	
4	DJS23CLPC504	Web Programming Laboratory		2		1	2				25	25					25	25	50	1	1
5	DJS23CCPE511	Advanced Algorithms	3			3	2	60				60	15	15	10	40		40	100	3	4
	DJS23CLPE511	Advanced Algorithms Laboratory		2		1	2		25			25					25	25	50	1	
	DJS23CCPE512	Advanced Operating Systems	3			3	2	60				60	15	15	10	40		40	100	3	
	DJS23CLPE512	Advanced Operating Systems Laboratory		2		1	2		25			25					25	25	50	1	
	DJS23CCPE513	Advanced Database Management Systems	3			3	2	60				60	15	15	10	40		40	100	3	
	DJS23CLPE513	Advanced Database Management Systems Laboratory		2		1	2		25			25					25	25	50	1	
	DJS23CCPE514	Computer Graphics and Image Processing	3			3	2	60				60	15	15	10	40		40	100	3	
	DJS23CLPE514	Computer Graphics and Image Processing Laboratory		2		1	2		25			25					25	25	50	1	
6	DJS23CCMD501	IoT-Centric Processor Organization and Architecture	2			2	2	60				60	15	15	10	40		40	100	2	3
	DJS23CLMD501	IoT-Centric Processor Organization and Architecture Laboratory		2		1	2		25			25					25	25	50	1	
7	DJS23IPSCX03	Innovative Product Development III	--	2	--	**1**	2	--	--	--	25	25	--				25	**25**	**50**	**1**	**1**
8	DJS23ICHSX09	Constitution of India (Audit Course)	1																		
		Total	**15**	**14**		**21**	**24**	**300**	**125**		**50**	**475**	**75**	**75**	**50**	**200**	**175**	**375**	**850**	**21**	**21**

Prepared by Checked by Head of the Department Principal

Scheme for Second Year Undergraduate Program in Computer Engineering (Autonomous) (DJS23 Scheme)

SEMESTER VI

Sr. No.	Course Code	Course	Teaching Scheme				Semester End Examination (A)						Continuous Assessment						Aggregate (A+B)	Credits Earned	
			Theory (hrs)	Practical (hrs)	Tutorial (hrs)	Total Credits	Duration (hrs)	Theory	Oral	Pract.	Oral & Practical	SEE Total (A)	TT 1	TT 2	TT 3	Total	Term Work Total	Total (B)			
1	DJS23CCPC601	Machine Learning	3			3	2	60				**60**	15	15	10	40		**40**	**100**	**3**	**4**
	DJS23CLPC601	Machine Learning Laboratory		2		1	2		25			**25**					25	**25**	**50**	**1**	
2	DJS23CCPC602	Software Engineering	3			3	2	60				**60**	15	15	10	40		**40**	**100**	**3**	**4**
	DJS23CLPC602	Software Engineering Laboratory		2		1	2		25			**25**					25	**25**	**50**	**1**	
3	DJS23CLPC603	DevOps Laboratory		2		**1**	2				25	**25**					25	**25**	**50**	**1**	**1**
4	DJS23CCPE611	Big Data Analytics	3			3	2	60				**60**	15	15	10	40		**40**	**100**	**3**	**4**
	DJS23CLPE611	Big Data Analytics Laboratory		2		1	2		25			**25**					25	**25**	**50**	**1**	
	DJS23CCPE612	Game Theory	3			3	2	60				**60**	15	15	10	40		**40**	**100**	**3**	
	DJS23CLPE612	Game Theory Laboratory		2		1	2		25			**25**					25	**25**	**50**	**1**	
	DJS23CCPE613	Computer Vision	3			3	2	60				**60**	15	15	10	40		**40**	**100**	**3**	
	DJS23CLPE613	Computer Vision Laboratory		2		1	2		25			**25**					25	**25**	**50**	**1**	
5	DJS23CCPE621	Advanced Network Design	3			3	2	60				**60**	15	15	10	40		**40**	**100**	**3**	**4**
	DJS23CLPE621	Advanced Network Design Laboratory		2		1	2		25			**25**					25	**25**	**50**	**1**	
	DJS23CCPE622	Distributed Databases	3			3	2	60				**60**	15	15	10	40		**40**	**100**	**3**	
	DJS23CLPE622	Distributed Databases Laboratory		2		1	2		25			**25**					25	**25**	**50**	**1**	
	DJS23CCPE623	High Performance Computing	3			3	2	60				**60**	15	15	10	40		**40**	**100**	**3**	
	DJS23CLPE623	High Performance Computing Laboratory		2		1	2		25			**25**					25	**25**	**50**	**1**	
6	DJS23CCMD601	Security in IOT Ecosystem	2			2	2	60				**60**	15	15	10	40		**40**	**100**	**2**	**3**
	DJS23CLMD601	Security in IOT Ecosystem Laboratory		2		1	2		25			**25**					25	**25**	**50**	**1**	
7	DJS23IPSCX04	Innovative Product Development IV	--	2	--	**1**	--	--		--	25	**25**			--		25	**25**	**50**	**1**	**1**
8	DJS23ITHSX10	Environmental Science Tutorial			1	**1**											25	**25**	**25**	**1**	**1**
		Total	**14**	**14**	**1**	**22**	**22**	**300**	**125**		**50**	**475**	**75**	**75**	**50**	**200**	**200**	**400**	**875**	**22**	**22**

Prepared by Checked by Head of the Department Principal

Program: Computer Engineering **T.Y B.Tech.** **Semester: V**

Course: Data Warehousing and Mining (DJS23CEC501)

Course: Data Warehousing and Mining Laboratory (DJS23CEL501)

Pre-requisite:

1. Basic database concepts
2. Concepts of algorithm design and analysis

Course Objectives:

1. To identify the scope and essentiality of Data Mining and Warehouse.
2. To analyze data, choose relevant models and algorithms for respective applications.
3. To develop research interest towards advances in data mining.

Outcomes: On successful completion of course, learner will be able to:

1. Design data warehouse models using dimension-modeling techniques.
2. Analyse the data by applying Online Analytical Processing (OLAP) operations for strategic decisions.
3. Apply preprocessing techniques to the given raw data.
4. Apply appropriate data mining techniques on data sets to retrieve relevant information.

Data Warehousing and Mining (DJS23CEC501)		
Unit	**Description**	**Duration**
1	**Introduction to Data Warehouse and Dimensional modelling:** Introduction to Strategic Information, Need for Strategic Information, Features of Data Warehouse, Data warehouse versus Data Marts, Data warehouse versus Data Lake, Top-down versus Bottom-up approach. Data warehouse architecture, metadata, E-R modelling versus Dimensional Modelling, Information Package Diagram, STAR schema, STAR schema keys, Snowflake Schema, Fact Constellation Schema, Factless Fact tables, Update to the dimension tables, Aggregate fact tables.	08
2	**ETL Process and OLAP:** Major steps in ETL process, Data extraction: Techniques, Data transformation: Basic tasks, Major transformation types, Data Loading: Applying Data, OLTP Vs OLAP, OLAP definition, Dimensional Analysis, Hypercubes, OLAP operations: Drill down, Roll up, Slice, Dice and Rotation, OLAP models: MOLAP, ROLAP.	06
3	**Introduction to Data Mining, Data Exploration and Preprocessing:** Data Mining Task and Techniques, KDD process, Issues in Data Mining, Applications of Data Mining, Data Exploration: Types of Attributes, Statistical Description of Data, Data Visualization, Measuring data similarity and dissimilarity. Data Preprocessing: Major tasks in preprocessing, Data Cleaning: Missing values, Noisy data; Data Integration: Entity Identification Problem, Redundancy and Correlation Analysis, Tuple Duplication, Data Value Conflict Detection and Resolution; Data Reduction: Attribute subset selection, Histograms, Clustering and Sampling; Data Transformation & Data Discretization: Data Transformation by Normalization, Discretization by Binning, Discretization by Histogram Analysis, Concept hierarchy generation for Nominal data	06
4	**Classification and Clustering:**	08

	Classification Basic Concepts of classification, Decision Tree Induction, Attribute Selection Measures using Information Gain, Tree pruning Bayes Classification Methods: Bayes' Theorem, Naïve Bayesian Classification Model Evaluation & Selection: Metrics for Evaluating Classifier Performance, Holdout Method and Random Subsampling, Cross Validation, Bootstrap, Improving Classification Accuracy: Ensemble classification, Bagging, Boosting and AdaBoost, Random Forests, **Clustering:** Cluster Analysis and Requirements of Cluster Analysis Partitioning Methods: k-Means, k-Medoids Hierarchical Methods: Agglomerative, Divisive Evaluation of Clustering: Assessing Clustering Tendency, Determining Number of Clusters and Measuring cluster quality: Intrinsic and Extrinsic methods	
5	**Mining Frequent Patterns and Association Rules:** Market Basket Analysis, Frequent Item sets, Closed Item sets, and Association Rule Frequent Item set Mining Methods: Apriori Algorithm, Association Rule Generation, FP growth, Mining frequent Itemsets using Vertical Data Format	05
6	**Spatial and Web Mining:** Spatial Data, Spatial Vs. Classical Data Mining, Spatial Data Structures, Mining Spatial Association and Co-location Patterns, Spatial Clustering Techniques: CLARANS Extension, **Web Mining**: Web Content Mining, Web Structure Mining, Web Usage mining, Applications of Web Mining	06
	Total	39

Data Warehousing and Mining Laboratory (DJS23CEL501)	
Exp.	**Suggested experiments**
1.	Build Data Warehouse/Data Mart for a given problem statement i. Identifying the source tables and populating sample data ii. Making information package diagram iii. Design dimensional data model i.e. Star schema, Snowflake schema and Fact Constellation schema (if applicable)
2.	Perform data Pre-processing task on your dataset
3.	To perform various OLAP operations such as slice, dice, drilldown, rollup, pivot
4.	Implementation of Classification algorithm i. Using Decision Tree ID3 ii. Naïve Bayes algorithm
5.	Implementation of Clustering algorithm i. K-means ii. Hierarchical clustering (single/complete/average) iii. DBScan
6.	Implementation of Association Rule Mining algorithm 1. Apriori algorithm 2. FP Tree algorithm
7.	Demonstrate performing Classification, Clustering, Association algorithm on data sets using data mining tool (WEKA, R tool, XL Miner, etc.)

8.	Implementation of page rank algorithm
9.	Implementation of HITS algorithm.
10.	Implementation of Spatial Clustering Algorithm- CLARANS Extensions.
11.	Case study on recent data mining applications

Batchwise laboratory work of minimum 8experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Books Recommended:

1. Paulraj Ponniah, "Data Warehousing: Fundamentals for IT Professionals", 2nd Edition, Wiley India, 2013.
2. Han, Kamber, "Data Mining Concepts and Techniques", 3rd Edition, Morgan Kaufmann, 2012.
3. H. Dunham, "Data Mining: Introductory and Advanced Topics", 1st Edition, Pearson Education, 2006.

Reference Books:

1. Theraja Reema, "Data Warehousing", 1st Edition, Oxford University Press, 2009.
2. P. N. Tan, M. Steinbach, Vipin Kumar, "Introduction to Data Mining", 2nd Edition, Pearson Education, 2018.

Prepared by Checked by Head of the Department Principal

Program: Computer Engineering **T.Y B.Tech. Semester: V**

Course: Artificial Intelligence (DJS23CCPC502)

Course: Artificial Intelligence Laboratory (DJS23CLPC502)

Pre-requisite:

Knowledge of

1. Programming Language
2. Algorithms

Objectives:

1. Provide the basic ideas and techniques underlying the design of intelligent systems.
2. Impart the knowledge of various search techniques for problem solving.
3. Learn knowledge representation, planning and provide the knowledge to deal with uncertain and incomplete information.

Outcomes: On completion of the course, the learner will be able to:

1. Provide the basic ideas and techniques underlying the design of intelligent systems.
2. Impart the knowledge of various search techniques for problem solving.
3. Learn knowledge representation, planning and provide the knowledge to deal with uncertain and incomplete information.

Artificial Intelligence (DJS23CCPC502)		
Unit	**Description**	**Duration**
1	**Introduction to Artificial Intelligence:** Introduction, History of Artificial Intelligence, Intelligent Systems: Categorization of Intelligent System, Components of AI Program, Foundations of AI **Agents and Environments:** Concept of rationality, Nature of environment, Structure of Agents, Types of Agents, PEAS representation for an Agent	**05**
2	**Problem Solving:** Problem Solving Agent, Formulating Problems, Example Problems. **Search Methods:** **Uninformed Search:** Breadth First Search (BFS), Depth First Search (DFS), Depth Limited Search, Depth First Iterative Deepening (DFID) **Informed Search:** Greedy Best First Search, A* Search, Memory bounded heuristic Search. **Local Search Algorithms and Optimization Problems:** Hill Climbing search, Simulated Annealing, Genetic algorithms, Ant Colony Optimization **Adversarial Search:** Game Playing, The Minimax algorithm, Alpha-Beta Pruning	**12**
3	**Knowledge Representation and Reasoning:** Knowledge based Agents, The Wumpus World, Propositional Logic, First Order Logic, Inference in FOL, Conjunctive Normal Form, Forward Chaining, Backward Chaining,	**07**

	Unification, Resolution, Logic Programming (PROLOG), Semantic networks **Uncertain Knowledge and Reasoning:** Representing knowledge in an uncertain domain, The semantics of Bayesian Belief Network, Inference in Belief Network	
4	**Planning:** The planning problem, Planning with state space search, Planning graphs, Partial order planning, Hierarchical planning	**04**
5	**Learning:** Types of Learning, Inductive Learning **Artificial Neural Networks:** McCulloh Pitts Model, Perceptron, Feed Forward Network, Backpropagation Algorithm, Self-Organizing Map	**07**
6	**Expert System:** Introduction, Phases in building Expert Systems, ES Architecture, Case Study on MYCIN Rule based system **Advanced topic:** Introduction to Generative AI, Explainable AI,	**04**
	Total	**39**

Artificial Intelligence Laboratory (DJS23CLPC502)	
Suggested List of Experiments	
LAB	**Explanation of Activity**
Lab 1	Select a problem statement relevant to AI. i) Identify the problem ii) PEAS Description iii) Problem formulation
Lab 2	Identify and analyze Uninformed Search Algorithm to solve the problem. Implement BFS/DFS search algorithms to reach goal state.
Lab 3	Implement DFID search algorithms to reach goal state.
Lab 4	Identify and analyze Informed Search Algorithm to solve the problem. Implement A* search algorithm to reach goal state
Lab 5	Program to implement Local Search algorithm: Hill Climbing search
Lab 6	Program on Genetic Algorithm to solve a optimization problem in AI.
Lab 7	Program to implement learning: Perceptron Learning / Backpropagation Algorithm.
Lab 8	The laboratory will emphasize the use of PROLOG. (For example, Program to implement Family Tree in Prolog)
Lab 9	Implementation on any AI game: Wumpus world, Tic-tac-toe, 8-Queens Problem
Lab 10	Case study of an AI Application.

Batchwise laboratory work of minimum 8 experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Books Recommended:

Text Books

1. Stuart J. Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach" 3rd Edition, Pearson Education, 2010.
2. George F Luger, "Artificial Intelligence" 6th Edition, Pearson Education, 2021.
3. Deepak Khemani, "A First Course in Artificial Intelligence", 6th Reprint, McGraw Hill Education, 2018.
4. Saroj Kaushik, "Artificial Intelligence", 1st Edition, Cengage Learning, 2010

Reference Books

1. AP Engelbrecht, "Computational Intelligence", 3rd Edition, Wiley–Blackwell, 2021.
2. Elaine Rich and Kevin Knight, "Artificial Intelligence" 3rd Edition, McGraw Hill Education, 2017.
3. Ivan Bratko, "PROLOG Programming for Artificial Intelligence", 4th Edition, Addison-Wesley, 2011.
4. Hagan, Demuth, Beale, "Neural Network Design", 2nd Edition, Martin Hagan, 2014.
5. Ronald J. Martin, "The Age of Artificial Intelligence", Independently Published, 2023.

Prepared by Checked by Head of the Department Principal

Program: T.Y B.Tech **Semester: V**

Course: Automata Theory and Compiler Design (DJS23CCPC503)

Course: Automata Theory and Compiler Design Laboratory (DJS23CLPC503)

Pre-requisite:

1. Programming.
2. Algorithms and Data Structures.

Objectives:

1. To introduce the fundamental concepts of formal languages, grammar and automata theory.
2. Introduce the major concepts of language translation and compiler design and impart the knowledge of practical skills necessary for constructing a compiler.

Outcomes: On completion of the course, the learner will be able to:

1. Understand the basic properties of formal languages and differentiate between them.
2. Build grammar to produce strings from a specific language and apply different parsing algorithms.
3. Acquire concepts related to computational models including decidability and intractability.
4. Understand the basics of analysis phase of compilation and its relationship with automata.
5. Understand synthesis phases of compilation used to construct target language statements.

Automata Theory and Compiler Design (DJS23CCPC503)		
Unit	**Description**	**Duration**
1	**Introduction to Automata Theory**: Central Concepts of Automata-theory, Deterministic Finite Automata (DFA), Non- Deterministic Finite Automata (NFA), Epsilon- NFA, NFA to DFA Conversion, Minimization of DFA **Introduction to Compiler Design:** Language Processors, Phases of Compilers	08
2	**Regular Expressions and Languages:** Regular Expressions, Finite Automata and Regular Expressions, Proving Languages Not to Be Regular **Lexical Analysis Phase of compiler Design:** Role of Lexical Analyzer, Input Buffering, Specification of Token, Recognition of Token.	06
3	**Context Free Grammars:** Definition and designing CFGs, Derivations Using a Grammar, Parse Trees, Ambiguity and Elimination of Ambiguity, Elimination of Left Recursion, Left Factoring. **Syntax Analysis Phase of Compilers:** Role of Parser, Top-Down Parsing, First and Follow of a Variable, Bottom-up Parsing, Introduction to LR Parsing	08
4	**Push Down Automata:** Definition of the Pushdown Automata, The Languages of a PDA. **Syntax Directed Translation:** Introduction to Syntax-Directed Definitions	05

5	**Introduction to Turing Machine:** Problems that Computers Cannot Solve, The Turing machine, problems, Programming Techniques for Turing Machine, Extensions to the Basic Turing Machine **Undecidability:** A language That Is Not Recursively Enumerable, An Undecidable Problem That Is RE.	06
6	**Introduction to Synthesis Phase of Compilers:** **Intermediate Code Generation**: Variants of Syntax Trees, Three-Address Code. **Code optimization**: Scope of optimization, DAG representation, Basic blocks, Machine dependent and Machine independent optimization. **Code Generation**: Issues in the Design of a Code Generator	06
	Total	**39**

Automata Theory and Compiler Design Laboratory (DJS23CLPC503)	
Exp.	**Suggested experiments**
1	Write a program to find epsilon closure of all states of any given NFA with epsilon transitions.
2	Design and implement a lexical analyser for given language using C and the lexical analyser should ignore redundant spaces, tabs and new line.
3	Write a program to convert CFG to CNF.
4	Write a program to implement Left recursion and left factoring.
5	Write a program to find First and Follow of a Variable.
6	To create and simulate automata using JFLAP.
7	Any real-life application of automata theory.
8	Case Study: LLVM

Batchwise laboratory work of **08** experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Books Recommended:

Text Books:

1. John E Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, "Introduction to Automata Theory, Languages and Computation", Third Edition, Pearson.
2. Alfred V.Aho, Monica S.Lam,Ravi Sethi, Jeffrey D. Ullman, " Compilers Principles, Techniques and Tools", 2nd Edition Pearson, 2014.

Reference Books:

1. Elain Rich, "Automata, Computability and complexity", 1st Edition, Pearson Education, 2018.
2. K.L.P Mishra, N Chandrashekaran, 'Theory of Computer Science", 3rd Edition, PHI, 2012.
3. Peter Linz, "An introduction to Formal Languages and Automata ", 6th Edition, Narosa Publishers, 2017.
4. K Muneeswaran, "Compiler Design", Oxford University Press, 2013

Program: B. Tech. in Computer Engineering **T.Y B.Tech.** **Semester: V**

Course: Web Programming Laboratory (DJS23CLPC504)

Pre-requisite:

1. Basics of programming

Objectives:

1. To get familiar with the basics of Web Programming.
2. To expose students to Basics and Advanced concepts in REACT.
3. To orient students to Fundamentals of node.js and express framework.
4. To understand REST API and MongoDB for Frontend and Backend Connectivity.

Outcomes: On completion of the course, the learner will be able to:

1. Implement interactive web page(s) using HTML5, CSS3 and JavaScript.
2. Implement Single Page Application using React.js and Node.js Framework.
3. Construct web based Node.js applications using Express.
4. Apply MongoDB for frontend and backend connectivity using REST API.

Web Programming Laboratory (DJS23CLPC504)		
Unit	**Description**	**Duration**
1	**HTML5, CSS3 and Bootstrap5:** **HTML5:** Introduction and Advantages of HTML5, HTML formatting, Hyperlinks, Images, tables, Lists, Elements (Block & Inline), Attributes, Page Layout, Semantic Elements, HTML5 Web Forms, HTML5 Media (Video & Audio). **CSS3:** Introducing CSS3, Selectors, Border, Box Model, Margin & Padding, Background Images & Colors and Other Decorative (Texts, Fonts, Links, Lists, Tables), Positioning, Combinators, Pseudo-class and Pseudo-element, CSS Attribute Selectors, 2D and 3D Transformations, Transitions and Animations, @property, Flexbox, CSS3-Multi Column Layout, Media Queries. **Bootstrap5:** Introduction to Bootstrap, Containers, Bootstrap Grids, Bootstrap Cards, Bootstrap JS (Navbar, Offcanvas, Collapse, Modal, Carousel), Flex, Bootstrap Forms.	4
2	**JavaScript:** Introduction to JavaScript, JavaScript DOM Model, var, let, const, Operators, primitive data types & strings, conditional, loop, for each loop, operators, ternary operators, RegExp. Arrow functions, normal functions - Lexical this - Events, Handling events - Spread operator, Destructuring - named imports, default import, map, filter, reduce, Date and Objects. Call back system, Asynchronous, promises - Async, await, JSON Introduction, Syntax.	6
3	**React Fundamental:** Introduction to Vue.js vs. Angular vs. React, Installation, installing libraries, Folder and file structure, Components, Component lifecycle, Props, State, Events, React Conditional, map, keys, React Router and Single page applications, Forms, Form Handling. Refs, Use effects, Hooks, Flux.	6

4	**Node.js:** Node.js, Setup Development Environment: Installation of Node.js, Working in REPL, Node JS Console, apply concepts like data types, objects, methods, object-oriented programming, and classes in the context of backend development, Creating simple Node Server, Request and Response, Routing responses, NPM JavaScript Build Processes, Event Loop and Emitters, File System Interaction, Modules, Native Node drivers.	4
5	**Express.js:** Introduction, Installation, Express router, REST API, Generator, Authentication, sessions, Integrating with React, Commercial deployment.	4
6	**Database Connectivity:** MongoDB Installation, connecting to MongoDB, CRUD Operations, Frontend Integration with React, User Authentication (JWT), Role-based Access Control, connecting MongoDB to Node-RED. Hosting Backend (e.g., Heroku, Vercel), Hosting Frontend (e.g., Netlify, Vercel)	2
	Total	**26**

Web Programming Laboratory (DJS23CLPC504)	
Exp.	**Suggested experiments**
1	Using HTML5 layout tags develop informative page with sections which include various images, links to other pages for navigation, make use of all possible formatting (for example font, color etc.).
2	Create form in HTML5 with all form elements. Apply form validations (e.g., Email, mobile, Pin code, Password) using JavaScript.
3	Apply CSS properties, Border, margins, Padding, Navigation, dropdown list to page created in First and Second Experiments.
4	Create an application to implement a counter application in JavaScript.
5	Create an application to demonstrate JSX, Components, Props, State in React.
6	Create an application to demonstrate Forms, Events, Routers, Refs, Keys in React.
7	Create an application to demonstrate use of Conditional rendering in React JS.
8	Create an application to build a simple web server that serves static content. They can learn how to use the http module to create a server, and how to handle requests and responses.
9	Create an application to demonstrate the implementation of Call back system, Asynchronous, promises - Async, await in node js.
10	Create an application to demonstrate connection of Node-RED with MongoDB.
11	Build a RESTful API using MongoDB.

Batchwise laboratory work of minimum **8** experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Mini Project:
Develop website using MERN stack. Website must include home page, and at least 3 forms (with Validation), use at least HTML5, CSS/Bootstrap, JavaScript, React.js web technologies (Students can

also use Node-RED to create an IOT based project). Database support is needed. Deploy website on live webserver and access through URL.

Books Recommended:

Text Books:

1. John Dean, "Web Programming with HTML5, CSS3 and JavaScript", Jones & Bartlett Learning, 2019.
2. Glenn Johnson, "Programming in HTML5 with JavaScript and CSS3", Microsoft Press, 2013 Edition.
3. Adam Bretz and Colin J. Ihrig, "Full Stack JavaScript Development with MEAN", SitePoint Pty. Ltd., 2015.
4. Simon Holmes Clive Harber, "Getting MEAN with Mongo, Express, Angular, and Node", Manning Publications, 2015.
5. Venkat Subramaniam, "Rediscovering JavaScript, Master ES6, ES7, and ES8", The Pragmatic Bookshelf, 2018.
6. Alex Banks and Eve Porcello, "Learning React Functional Web Development with React and Redux", O'Reilly, 1st Edition, 2017 Edition 5.
7. Andrew Mead, "Learning Node.js Development", Packt Publishing, 2018 Edition 6.
8. Valentin Bojinov, "RESTful Web API Design with Node.js 10", Packt Publication, 2018.

Reference Books:

1. Ethan Brown, "Web Development with Node and Express", O'Reilly, 2019.
2. Shama Hoque "Full-Stack React Projects: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node.js", 2nd Edition Packt Publication, 2020.

Program: Computer Engineering **T.Y B.Tech. Semester: V**

Course: Advanced Algorithms (DJS23CCPE511)

Course: Advanced Algorithms Laboratory (DJS23CLPE511)

Pre-requisite:

1. Data structures
2. Analysis of Algorithms

Objectives:

To provide conceptual and practical knowledge of Advance Algorithm

Outcomes: On completion of the course, the learner will be able to:

1. Analyse the chosen algorithm.
2. Choose appropriate data structure and algorithm for given problem statement.
3. Design the algorithm.
4. Classify the algorithms based on P NP and NP hard parameters

Advanced Algorithms (DJS23CCPE511)		
Unit	**Description**	**Duration**
1	**Analysis of Algorithm Based on Time:** **Asymptotic notations:** Omega, Theta, Big-O, Small-o, small Omega and Tilde **Amortized Analysis:** Aggregate Method, Accounting Method, Potential Method RAM model analysis of algorithm	04
2	**Probabilistic and Randomized Algorithm:** Probabilistic approach to algorithm and Randomized Analysis, Indicator Random Variable (IRV), Randomized Quick Sort, Analysis of Hiring Problem, Las Vegas and Monte Carlo algorithm	05
3	**Advanced Data Structures:** Balanced Search Trees: Red-Black Tree, Randomized BST Heap and Operations: Binomial Tree, Binomial Heap, Treap Spatial Data Structure: KD Tree, R Tree Probabilistic Data Structure: LogLog and HyperLogLog, Count Min sketch.	11
4	**Graph Based Algorithms:** Flow Network Introduction: Residual Network, Augmenting Path, Ford-Fulkerson Method, Edmonds-Karp Method, Push-Relable Algorithm, Relable to Front algorithm. Bipartite Matching: Maximum Bipartite Matching.	07
5	**Computational Geometry:** Line Segment Properties, Convex Hull Graham's scan algorithm.	06

	Geometric Searching: Point Location in polygon using Ray Crossing Online Algorithms: Competitive Ratio, K-Server Special topic: Cuckoo search algorithm.	
6	**Algorithm Classes:** P, NP, NP Hardness and NP Completeness Np Completeness Proofs: Satisfiability (3 sat), Reducibility, TSP. Approximation Algorithms: Vertex Cover Problem, Travelling Sales Person problem Network Approximation: Randomized Rounding, Primal Dual algorithms	06
	Total	**39**

Advanced Algorithms Laboratory (DJS23CLPE511)	
Exp.	**Suggested experiments**
1	To perform Amortized Analysis
2	To implement Randomized Algorithms (Randomized Quick Sort)
3	To implement Randomized Algorithms (Hiring Problem)
4	To implement Advanced Data Structure (Red-black Tree Operations)
5	To implement Advanced Data Structure (Binomial Tree Operations)
6	To implement Advanced Data Structure (R Tree Operations)
7	To implement Advanced Data Structure (KD Tree Operations)
8	To implement Advanced Data Structure (MinHash implementation)
9	To implement Graph Based Algorithms (Ford Fulkerson Method)
10	To implement Graph Based Algorithms (Push Relable Method)
11	To implement Graph Based Algorithms (Maximum Bipartite Matching)
12	To implement Computational Geometry (Graham Scan Algorithm)
13	To implement Online Algorithms (K-Server algorithm)
14	To implement Approximation Algorithm (Approximate TSP implementation)

Batchwise laboratory work of minimum 10 experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Books Recommended:

Text books:

1. Thomas H Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein, Introduction to Algorithms, Third Edition, The MIT Press, 2009
2. S. Sridhar, Design and Analysis of Algorithms, Second Edition, Oxford University Press, 2014
3. Horowitz, Sahani and Rajsekaran, Fundamentals of Computer Algorithms, 2nd Edition, Universities Press, 2008.

4. Harsh Bhasin, Algorithms Design and Analysis, Oxford University Press, 2015.

Reference Books:

1. Rajeev Motwani, Prabhakar Raghavan, Randomized Algorithm, Cambridge University, 1995.
2. S. K. Basu, Design Methods and Analysis of Algorithm, 2nd Edition, PHI, 2013.
3. Vijay V. Vajirani, Approximation Algorithms, Springer, 2003.
4. Sanjeev Arora, Boaz Barak, Computational Complexity, Princeton University, 2007.

Prepared by Checked by Head of the Department Principal

Program: Computer Engineering **T.YB.Tech Semester: V**

Course: Advanced Operating Systems (DJS23CCPE512)

Course: Advanced Operating Systems Lab (DJS23CLPE512)

Pre-requisite: Operating System and Computer Organization.

Course Objectives:

1. To understand the difference between distributed, multiprocessor and virtualization concepts.
2. To explore Real time operating system concepts.
3. To explore mobile operating systems.

Outcomes: On successful completion of course, learner will be able to:

1. Understand different types of operating systems concepts to solve real life problems.
2. Analyze system performance by applying virtualization concepts.
3. Understand mobile operating systems concept.

Advanced Operating Systems (DJS23CCPE512)		
Unit	**Description**	**Duration**
1	**Introduction:** Functions of operating systems, design approaches: layered, kernel based and virtual machine approach, need for advanced operating systems, types of advanced operating systems (NOS, DOS, Multiprocessor OS, Mobile OS, RTOS, Cloud OS).	**04**
2	**Distributed operating Systems:** Architecture of distributed operating systems, system architecture types, issues in distributed operating systems, inherent limitation of distribute systems. Distributed mutual exclusion: classification of mutual exclusion algorithms, Lamport's, token- based algorithm, Suzuki-Kasami's Broadcast algorithm, Raymond's Tree based algorithm Distributed deadlock detection, Distributed file systems.	**09**
3	**Real Time Operating Systems:** Basic Model of Real time systems, Characteristics, Applications of Real time systems, Real time task scheduling, Types of tasks and their characteristics. Task Scheduling, Clock driven Scheduling, Hybrid Schedulers, Event driven Scheduling, EDF Scheduling, Rate Monotonic Algorithm, handling resource sharing Resource Handling: Resource Sharing, Priority Inversion, PIP, PCP, HLP, Scheduling real time tasks in distributed systems.	**09**
4	**Multiprocessor Operating Systems:**	**06**

	Introduction, Basic multiprocessor system architectures, design issues, Threads Process synchronization: the test and set instruction, the swap instruction, implementation of the process waits. Processor scheduling: Issues, Co- scheduling, Smart scheduling, Affinity Based scheduling	
5	**Virtualization:** Introduction to Virtualization, Types of Virtualizations, Bare Metal (XEN), Hosted (KVM) Virtualization, Para virtualization, Full virtualization, Emulation, Server Virtualization, Network Virtualization and Storage Virtualization.	**06**
6	**Mobile Operating System**: Mobile OS: Architecture, Android OS, iOS, Virtual OS, Cloud OS and their design issues, Application development using Android	**05**
	Total	**39**

Advanced Operating System Lab (DJS23CLPE512)	
Sr. No.	**Suggested Practical**
1	Implement concurrent client-server application.
2	Simulate Lamport's logical clock
3	Implement Ricart-Aggarwala Algorithm.
4	Demonstrate deadlock detection using Edge Chasing algorithm.
5	Demonstrate hosted virtualization using KVM.
6	Load a new operating system virtually on the client machine using the concept of bare metal virtualization by XEN.
7	Hello world, linking activities, passing data.
8	Create a simple list view with image and text.
9	Integrate a website inside an application, use of SQLite.
10	Application development using Android.

Batchwise laboratory work of minimum 8 experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Books Recommended:

Textbooks:

1. Mukesh Singhal, Niranjan G. Shivaratri, "Advanced Concepts in Operating Systems", Tata McGraw-Hill Edition, 2020.
2. Andrew S. Tanenbaum and Maarten Van Steen, "Distributed Systems: Principles and Paradigms", 2nd Edition, Pearson, 2016.

Reference Books:

1. K. C. Wang, “Embedded and Real Time Operating System”, Springer, January 2022.
2. Abraham Silberschatz, Peter B. Galvin, Greg Gagne, “Operating System Concepts”, 8th Edition by, 2017
3. Rajib Mall, “Real-Time Systems: Theory and Practice”, Pearson Education India, 2016
4. Cris Wolf and Eric M Halter,” Virtualization from Desktop to Enterprise”, Apress, 2006.
5. K.C. Wang, “Embedded and Real-Time Operating Systems Hardcover” Springer, 2017.

Prepared by Checked by Head of the Department Principal

Program: Computer Engineering **T.Y B.Tech. Semester: V**

Course: Advanced Database Management System (DJS23CCPE513)

Course: Advanced Database Management System Laboratory (DJS23CCPL513)

Pre-requisite:

1. Basic knowledge of Database Management System

Objectives:

1. To provide an overview of advancement in SQL and Database technology, distributed database systems and document-oriented database.
2. To impart knowledge of query processing and optimization.
3. To understand the usage of advanced data models for real life applications and secure them.

Outcomes: On completion of the course, the learner will be able to:

1. Discuss new developments in database technology, implement advanced data models for real life applications and secure them.
2. Optimize query execution and design advanced databases like graph database and distributed database for better resource management.
3. Demonstrate the understanding of the concepts related to document-oriented databases.

Advance Database Management System (DJS23CCPE513)		
Unit	**Description**	**Duration**
1	**Advance Databases** Indexing and Hashing: Types of Single-Level Ordered Indexes; Multilevel Indexes; Dynamic Multilevel Indexes Using B-Trees and B+-Trees; Advanced database applications and architectures: e.g., Data Warehousing; Multimedia database; NoSQL, Native XML databases (NXD), Graph database, Object Oriented Database: Need, Types of object-oriented database, Impedance matching problem between OO languages and Relational database.	07
2	**Query processing and Optimization** Query Processing: Overview, Measures of Query cost, Selection operation, Sorting, Join Operations, and other Operations, Evaluation of Expression Query Optimization: Translations of SQL Queries into relational algebra, Heuristic approach and cost-based optimization.	07
3	**Advanced database architectures** Graph Database: Introduction, Graph database architecture, Types of graph database, Graph vs Relational database, Data modeling with graph, Use cases of Graph database, ArangoDB, Giraph. Distributed Database: Introduction, Fragmentation and its types.	07
4	**Document oriented database** Need of Document Oriented database, difference between Document Oriented Database and Traditional database, Types of encoding XML, Query execution in XML: XPath, XSLT, XQuery, JSON, BSON,	06

	Representation XML, JSON Objects. Case study on document oriented database: MongoDB.	
5	**Advanced data models** Temporal data models: Aspects of valid time, Bitemporal time and bi-temporal time with examples of each. Spatial model: Types of spatial data models - Raster, Vector and Image. MYSQL Postgres, Mobile databases.	06
6	**Data Security** Introduction to Database Security Issues; Authentication and authorization, Database auditing, Discretionary Access Control Based on Granting and Revoking Privileges, Mandatory Access Control and Role-Based Access Control for Multilevel Security Introduction to Statistical Database Security.	06
	Total	**39**

Course: Advanced Database Management System Laboratory (DJS23CCPL513)	
Exp.	**Suggested Experiments**
1	Case study on Professional and Commercial Databases: Summary and Comparison. E.g. Oracle, MongoDB, Microsoft SQL Server
2	To implement ORDBMS features in MySQL
3	Implementation of Query monitor (QEP- Query Execution Plan, Query Statistics)
4	Simulate Query optimization by applying an SQL Query on any database
5	Implementation of B/B+ Tree
6	Perform Fragmentation (Range, List, Hash and Key) in DDBS design.
7	Implementation of simple graph database.
8	Data handing using JSON.
9	Query execution on XML database
10	Implementation of simple document-oriented database using MongoDB.
11	Case study on Database security issues and measures taken to handle those issues. (Study and document a research paper / patent / product. If possible, suggest an improvement.)

Batchwise laboratory work of minimum 8 experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Batchwise tutorial sessions are to be conducted on topics which would help the learner to identify/analyze the problem and to apply problem solving techniques learnt.

Books Recommended:

Text Books:

1. Abraham Silberschatz, Henry F. Korth, Sudarshan, "Database System Concepts", 7th Edition, Mc Graw Hill, 2021.

2. Sveta Smirnova and Alkin Tezuysal, “My SQL Cookbook” 4th Edition, O’Reilly Publication, 2022.
3. Shannon Bradshaw, Eoin Brazil, “MongoDB: The Definitive Guide - Powerful and Scalable Data Storage”, Third Edition, O’Reilly Publication, 2020.
4. Christos Tjortjis, “Graph Databases Applications on Social Media Analytics and Smart Cities” 1st Edition, CRC Press, 2023.

Reference Books:

1. Vinicius M. Grippa and Sergey Kuzmichev, “Learning MySQL” 2nd Edition, O’Reilly Publication, 2021.
2. Tamer OEzsu, Patrick V, “Principles of Distributed Database System”, Springer Publication, 2020.

Prepared by Checked by Head of the Department Principal

Program: Computer Engineering **T.Y B.Tech. Semester: V**

Course: Computer Graphics and Image Processing (DJS23CCPE514)

Course: Computer Graphics and Image Processing Laboratory (DJS23CLPE514)

Pre-requisite:

1. A sound knowledge of Mathematics and a programming language.

Objectives:

1. This course helps the learner to understand theoretical relationships between computer graphics and image processing.
2. This course helps the learner to understand three dimensional environment representation in a computer, transformation of 2D/3D objects, basic mathematical techniques and algorithms used to build useful applications, imaging, and image processing techniques.

Outcomes: On completion of the course, the learner will be able to:

1. Understand the fundamental of Computer Graphics and Image processing.
2. Illustrate line drawing, circle drawing, clipping and polygon filling algorithms and demonstrate geometric representations,
3. Demonstrate various transformations on 2D & 3D objects.
4. Solve image enhancement and segmentation problems using spatial domain Techniques.

Computer Graphics and Image Processing (DJS23CCPE514)		
Unit	**Description**	**Duration**
1	**Basics of Computer graphics and Algorithms** Basics of Computer Graphics and its applications. Random Scan Displays and systems, Raster scan displays and systems. **Fundamentals of Digital Image Processing** Image as 2D data, Image representation in Gray scale, Binary and Color images. Fundamental steps in image processing. Coordinate conventions. Sampling and quantization. Basic relationship between pixels–neighborhood, adjacency, connectivity.	05
2	**Filled Area Primitives and transformations** DDA line drawing, Bresenham's line drawing algorithm. Circle drawing algorithms- Midpoint Circle generation algorithm, Bezier Curves. **Filled Area Primitives**- Boundary filling and flood filling. **Two Dimensional Transformations**-Translation, Rotation, Scaling, Reflection and Shearing, Composite transformations, Matrix representations and homogeneous coordinates. Basic 3D transformations.	10
3	**Clipping and Projections**	08

	Window to viewport transformation. Cohen Sutherland Line clipping algorithm. Sutherland Hodgeman Polygon clipping algorithm. Three dimensional viewing pipeline. **Projections**- Parallel and Perspective projections. Visible surface detection algorithms- Depth buffer algorithm, Scan line algorithm.	
4	**Image Enhancement in Spatial Domain** Intensity transformations: Image negative, Gray level Slicing, Thresholding, contrast stretching, histogram equalization. Basics of spatial filtering - Smoothing spatial filter Linear and nonlinear filters, and Sharpening Spatial Filters-Gradient and Laplacian.	08
5	**Image Segmentation** Fundamentals of Image Segmentation. Thresholding - Basics of Intensity thresholding and Global Thresholding. Region based Approach - Region Growing, Region Splitting and Merging. Edge Detection - Edge Operators- Sobel and Prewitt.	08
	Total	**39**

Computer Graphics and Image Processing Laboratory (DJS23CLPE514)	
Exp.	**Suggested experiments**
1	Digital Differential Analyzer Algorithm
2	Bresenham's Line Drawing Algorithm
3	Midpoint Circle Generation Algorithm
4	Ellipse Generation Algorithm
5	Two Dimensional Transformations
6	Coloring the Pictures using polygon filling algorithms
7	Curve Generation
8	Implementation of Image negative, Gray level Slicing and Thresholding
9	Implementation of Contrast Stretching and histogram equalization
10	Implementation of Image smoothing/ Image sharpening
11	Implementation of Edge detection using Sobel and Prewitt masks

Laboratory work of minimum 10 experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Books Recommended:

Text Books:

1. Donald Hearn and M. Pauline Baker, Computer Graphics, PHI, 2e, 1996
2. Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing. Pearson, 4e, 2017

References:

1. William M. Newman and Robert F. Sproull, Principles of Interactive Computer Graphics, McGraw Hill, 2001.

2. Zhigang Xiang and Roy Plastock, Computer Graphics (Schaum's outline Series), McGraw Hill, 2019.
3. David F. Rogers, Procedural Elements for Computer Graphics, Tata McGraw Hill, 2001.
4. M. Sonka, V. Hlavac, and R. Boyle, Image Processing, Analysis, and Machine Vision, Thomson India Edition, 4e, 2017.

Prepared by Checked by Head of the Department Principal

Program: Computer Engineering	T.Y B.Tech.	Semester: V
Course: IoT-Centric Processor Organization and Architecture (DJS23CCMD501)		
Course: IoT-Centric Processor Organization and Architecture Laboratory (DJS23CLMD501)		

Pre-requisite:

1. Basic knowledge of digital electronics.
2. Fundamentals of programming.
3. Basic understanding of computer organization.

Course Objectives:

1. Understand the fundamental concepts of processor organization and its relevance to IoT systems.
2. Learn memory hierarchy and interfacing techniques between processors and IoT peripherals.
3. Analyze different processor architectures optimized for IoT applications and Develop skills in programming and optimizing IoT processors.

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

1. Analyze and evaluate different processor architectures for IoT applications.
2. Design and implement memory systems for resource-constrained IoT devices and Develop efficient programs for IoT processors considering hardware constraints.
3. Optimize processor performance using Pipelining and Parallel Processing concepts.
4. Interface various IoT sensors and actuators with processors for optimization and security.

IoT-Centric Processor Organization and Architecture (DJS23CCMD501)		
Unit	**Description**	**Duration**
1	**Introduction to Processor Architecture:** - Basic computer organization - Von Neumann vs Harvard architecture - RISC vs CISC architecture - IoT processor requirements and constraints - Introduction to popular IoT processors (ARM Cortex-M, ESP32, etc.).	03
2	**Memory Systems and Hierarchy:** - Memory hierarchy in IoT systems - Cache organization and management - Virtual memory concepts - Flash memory and storage systems	04
3	**Instruction Set Architecture:** - ISA design principles - Instruction formats and addressing modes - RISC-V architecture - ARM instruction set for IoT	06

	- Optimization techniques for IoT applications - Assembly language programming.	
4	**Pipelining and Parallel Processing:** - Basic pipeline concepts - Pipeline hazards and solutions - Parallel processing in IoT context - Multi-core processors - Hardware acceleration units	04
5	**IoT Interfaces and Peripherals:** - I/O organization - Serial communication protocols (I2C, SPI, UART) - Wireless interfaces (BLE, WiFi, LoRa) - Sensor interfacing - Real-time processing requirements - Interrupt handling	05
6	**Advanced Topics and Optimization:** - Security features in IoT processors - Power management techniques - Real-time operating systems integration - Hardware accelerators for ML/AI - Edge computing considerations - Future trends in IoT processors	04
Total		26

IoT-Centric Processor Organization and Architecture Laboratory (DJS23CLMD501)	
Sr. No.	**Suggested Experiments**
1	Introduction to development boards (ARM/ESP32) and tool chains
2	Assembly language programming for basic operations
3	Memory access and management experiments
4	Cache performance analysis
5	Implementation of pipeline concepts
6	I2C sensor interfacing and data acquisition
7	SPI communication with external devices
8	UART communication implementation
9	WiFi/BLE module interfacing
10	Interrupt handling and real-time processing
11	Complete IoT system integration project

Laboratory work of minimum 8 experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Books Recommended:

Textbook:

1. Patterson, D. A., & Hennessy, J. L., Computer Organization and Design RISC-V Edition: The Hardware Software Interface, 2nd Edition, Morgan Kaufmann, 2023.

2. Sarah Harris & David Harris, Digital Design and Computer Architecture: RISC-V Edition, Morgan Kaufmann, 2024.
3. Raj Kamal, Internet of Things: Architecture and Design Principles, McGraw Hill Education, 2023.

Reference Books:

1. William Stallings, Computer Organization and Architecture, 11th Edition, Pearson, 2023.
2. Perry Xiao, IoT System Design: A Practical Approach with Microcontrollers, CRC Press, 2023.
3. ARM Limited, ARM Architecture Reference Manual, ARM Holdings, 2024

Prepared by Checked by Head of the Department Principal

Program: Computer Engineering **T.Y B.Tech.** **Semester: V**

Course: Innovative Product Development III (DJS23IPSCX03)

Course Objectives:

1. To acquaint the students with the process of identifying the need (considering a societal requirement) and ensuring that a solution is found out to address the same by designing and developing an innovative product.
2. To familiarize the students with the process of designing and developing a product, while they work as part of a team.
3. To acquaint the students with the process of applying basic engineering fundamentals, so as to attempt at the design and development of a successful value added product.
4. To inculcate the basic concepts of entrepreneurship and the process of self-learning and research required to conceptualise and create a successful product.

Outcomes:

Learner will be able to:

1. Identify the requirement for a product based on societal/research needs.
2. Apply knowledge and skills required to solve a societal need by conceptualising a product, especially while working in a team.
3. Use standard norms of engineering concepts/practices in the design and development of an innovative product.
4. Draw proper inferences through theoretical/ experimental/simulations and analyse the impact of the proposed method of design and development of the product.
5. Develop interpersonal skills, while working as a member of the team or as the leader.
6. Demonstrate capabilities of self-learning as part of the team, leading to life-long learning, which could eventually prepare themselves to be successful entrepreneurs.
7. Demonstrate product/project management principles during the design and development work and also excel in written (Technical paper preparation) as well as oral communication.

Guidelines for the proposed product design and development:

- Students shall form a team of 3 to 4 students (max allowed: 5-6 in extraordinary cases, subject to the approval of the department review committee and the Head of the department).
- Students should carry out a survey and identify the need, which shall be converted into conceptualization of a product, in consultation with the faculty supervisor/head of department/internal committee of faculty members.
- Students in the team shall understand the effective need for product development and accordingly select the best possible design in consultation with the faculty supervisor.
- Students shall convert the best design solution into a working model, using various components drawn from their domain as well as related interdisciplinary areas.
- Faculty supervisor may provide inputs to students during the entire span of the activity, spread over 2 semesters, wherein the main focus shall be on self-learning.

- A record in the form of an activity log-book is to be prepared by each team, wherein the team can record weekly progress of work. The guide/supervisor should verify the recorded notes/comments and approve the same on a weekly basis.
- The design solution is to be validated with proper justification and the report is to be compiled in a standard format and submitted to the department. Efforts are to be made by the students to try and publish a technical paper, either in the institute journal, "Techno Focus: Journal for Budding Engineers" or at a suitable publication, approved by the department research committee/ Head of the department.
- The focus should be on self-learning, capability to design and innovate new products as well as on developing the ability to address societal problems. Advancement of entrepreneurial capabilities and quality development of the students through the year long course should ensure that the design and development of a product of appropriate level and quality is carried out, spread over two semesters, i.e. during the semesters V and VI.

Guidelines for Assessment of the work:

- The review/ progress monitoring committee shall be constituted by the Head of the Department. The progress of design and development of the product is to be evaluated on a continuous basis, holding a minimum of two reviews in each semester.
- In the continuous assessment, focus shall also be on each individual student's contribution to the team activity, their understanding and involvement as well as responses to the questions being raised at all points in time.
- Distribution of term work marks during the subsequent semester shall be as given below:
    - Marks awarded by the supervisor based on log-book: 10
    - Marks awarded by review committee: 10
    - Quality of the write-up: 05

In the last review of the semester VI, the term work marks will be awarded as follows.

- Marks awarded by the supervisor (Considering technical paper writing): 15
- Marks awarded by the review committee: 10

Review/progress monitoring committee may consider the following points during the assessment.

- In the semester V, the entire design proposal shall be ready, including components/system selection as well as the cost analysis. Two reviews will be conducted based on the presentation given by the student's team.
    - First shall be for finalisation of the product selected.
    - Second shall be on finalisation of the proposed design of the product.

- In the semester VI, the expected work shall be procurement of components/systems, building of the working prototype, testing and validation of the results based on work completed in semester III.
    - First review is based on readiness of building the working prototype.
    - Second review shall be based on a presentation as well as the demonstration of the working model, during the last month of semester IV. This review will also look at the readiness of the proposed technical paper presentation of the team.

The overall work done by the team shall be assessed based on the following criteria;

1. Quality of survey/ need identification of the product.
2. Clarity of Problem definition (design and development) based on need.
3. Innovativeness in the proposed design.
4. Feasibility of the proposed design and selection of the best solution.
5. Cost effectiveness of the product.
6. Societal impact of the product.
7. Functioning of the working model as per stated requirements.
8. Effective use of standard engineering norms.
9. Contribution of each individual as a member or the team leader.
10. Clarity on the write-up and the technical paper prepared.

- The semester reviews (V and VI) may be based on relevant points listed above, as applicable.

Guidelines for Assessment of Semester Reviews:

- The write-up should be prepared as per the guidelines given by the department.
- The design and the development of the product shall be assessed through a presentation and demonstration of the working model by the student team to a panel of Internal and External Examiners, preferably from industry or any research organisations having an experience of more than five years, approved by the Head of the Institution. The presence of the external examiner is desirable only for the 2nd presentation in semester IV. Students are compulsorily required to present the outline of the technical paper prepared by them during the final review in semester VI.

Program: Computer Engineering **T.Y B.Tech.** **Semester: V**

Course: Constitution of India (DJS23ICHSX09)

Course:

Course Objectives:

1. To provide basic information about Indian constitution.
2. To identify individual role and ethical responsibility towards society.
3. To understand human rights and its implications.

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

1. Have general knowledge and legal literacy and thereby to take up competitive examinations.
2. Understand state and central policies, fundamental duties.
3. Understand Electoral Process, special provisions.
4. Understand powers and functions of Municipalities, Panchayats and Co-operative Societies.
5. Understand Engineering ethics and responsibilities of Engineers.
6. Understand Engineering Integrity & Reliability.

Constitution of India (DJS23ICHSX09)		
Unit	**Syllabus Content**	**Duration**
1	**Introduction to the Constitution of India** The Making of the Constitution and Salient features of the Constitution. Preamble to the Indian Constitution. Fundamental Rights & its limitations.	02
2	**Directive Principles of State Policy:** Relevance of Directive Principles, State Policy, Fundamental Duties. Union Executives – President, Prime Minister, Parliament, Supreme Court of India.	02
3	**State Executives:** Governor, Chief Minister, State Legislature, High Court of State. Electoral Process in India, Amendment Procedures, 42nd, 44th, 74th, 76th, 86th & 91st Amendments.	03
4	**Special Provisions:** For SC & ST, Special Provision for Women, Children & Backward Classes, Emergency Provisions.	02
5	**Human Rights:** Meaning and Definitions, Legislation Specific Themes in Human Rights, Working of National Human Rights Commission in India, Powers and functions of Municipalities, Panchayats and Co-Operative Societies.	02
6	**Scope & Aims of Engineering Ethics**: Responsibility of Engineers and Impediments to Responsibility. Risks, Safety and liability of Engineers. Honesty, Integrity & Reliability in Engineering.	02
	Total hours	13

Books Recommended:

Text books:

1. Durga Das Basu, *"Introduction to the Constitution on India",* (Students Edition) Prentice

Hall EEE, 19th / 20th Edition, 2001.
2. Charles E. Haries, Michael S. Pritchard and Michael J. Robins, *"Engineering Ethics",* Thompson Asia, 2003.

***Reference Books**:*

1. M. V. Pylee, *"An Introduction to Constitution of India",* Vikas Publishing, 3rd Edition, 2003.
2. M. Govindarajan, S. Natarajan, V. S. Senthilkumar, *"Engineering Ethics",* Prentice Hall of India Pvt. Ltd. New Delhi, 2013.
3. Brij Kishore Sharma, *"Introduction to the Constitution of India",* PHI Learning Pvt. Ltd., New Delhi, 7th Edition 2015.
4. Latest Publications of *Indian Institute of Human Rights*, New Delhi

Website Resources:

1. www.nptel.ac.in
2. www.hnlu.ac.in
3. www.nspe.org
4. www.preservearticles.com

Program: Computer Engineering **T.Y B.Tech.** **Semester: VI**

Course: Machine Learning (DJS23CCPC601)

Course: Machine Learning Laboratory (DJS23CLPC601)

Pre-requisite:

Data Structures, Basic Probability and Statistics, Data Mining

Objectives:

1. To introduce students to the basic concepts and techniques of Machine Learning.
2. To become familiar with regression, classification and clustering tasks.
3. To become familiar with Dimensionality Reduction Techniques.

Outcomes: On completion of the course, the learner will be able to:

1. Gain knowledge about basic concepts of Machine Learning
2. Understand and apply supervised learning algorithms
3. Understand and apply unsupervised learning algorithms
4. Design application using machine learning techniques

Machine Learning (DJS23CCPC601)		
Unit	**Description**	**Duration**
1	**Introduction to Machine Learning:** Types of Machine Learning, Steps involved in developing a Machine Learning Application, Evaluating a Learning Algorithm: Deciding what to try next, Evaluating Hypothesis, Model Selection and Train/ Validation/ Test Sets, Bias Vs variance: Regularization and Bias/ Variance, Learning Curve, Error Analysis, Handling Skewed Data: Error Matrices for Skewed Classes, Tradeoff between Precision and recall, Issues in Machine Learning, Application of Machine Learning	06
2	**Dimensionality Reduction**: Dimensionality Reduction Techniques: Principal components analysis (Eigen values, Eigen vectors, Orthogonality), Independent Component Analysis, Single value decomposition	07
3	**Supervised Learning:** **Learning with Regression and trees:** Learning with Regression: Simple Linear Regression, Multiple Linear Regression, Logistic Regression. Learning with Trees: Decision Trees, Constructing Decision Trees using Gini Index, Classification and Regression Trees (CART). **Classification:** Classification using Bayesian Belief networks, Hidden Markov Models Support Vector Machine: Maximum Margin Linear Separators, Quadratic Programming solution to finding maximum margin separators, Kernels for learning non-linear functions. Classification using k Nearest Neighbor Algorithm	14
4	**Unsupervised Learning:** **Clustering:** Basics of clustering, Hard vs Soft Clustering, Density Based Clustering: DBSCAN, Expectation maximization (EM) for soft clustering, **Semi-supervised learning** with EM using labelled and unlabeled data, Radial Basis functions.	07
5	**Applications of Machine Learning:**	05

	Recommender Systems, Machine Learning for Image Recognition, Sentiment Analysis, Machine Learning for video surveillance	
	Total	**39**

Machine Learning Laboratory (DJS23CLPC601)	
Exp.	**Suggested experiments**
1	To implement Linear Regression
2	To implement Logistic Regression
3	To implement CART decision tree algorithm.
4	To implement Support Vector Machine.
5	To implement Bayesian Classification.
6	To implement PCA.
7	To implement K-Nearest Neighbour.
8	To Implement Radial basis functions.
9	Mini project based on any machine learning application.

Batchwise laboratory work of minimum 08 experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Batchwise tutorial sessions are to be conducted on topics which would help the learner to identify/analyze the problem and to apply problem solving techniques learnt.

Books Recommended:

Text Books:

1. Ethem Alpaydın, Introduction to Machine Learning, 4th Edition, The MIT Press 2020
2. Peter Harrington, "Machine Learning In Action", 1 st Edition, Dreamtech Press 2012
3. Tom Mitchell , "Machine Learning", 1st Edition, McGraw Hill 2017
4. Andreas C, Müller and Sarah Guido, "Introduction to Machine Learning with Python: A Guide for Data Scientists", 1 st Edition, O'reilly 2016
5. Kevin P. Murphy, "Machine Learning: A Probabilistic Perspective" MIT Press 2012

Reference Books:

1. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition, 2019
2. Witten Ian H., Eibe Frank, Mark A. Hall, and Christopher J. Pal. "Data Mining: Practical machine learning tools and techniques", 1 st Edition, Morgan Kaufmann, 2016.
3. Han, Kamber, "Data Mining Concepts and Techniques", 3rd Edition, Morgan Kaufmann, 2012.
4. Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. Foundations of Machine Learning, The MIT Press, 2012 5.
5. H. Dunham, "Data Mining: Introductory and Advanced Topics", 1st Edition, Pearson Education, 2006.

Program: Computer Engineering **T.Y B. Tech Semester: VI**

Course: Software Engineering (DJS23CCPC602)

Course: Software Engineering Laboratory (DJS23CLPC602)

Pre-requisite:

1. Concepts of Object Oriented Programming & Methodology
2. Knowledge of developing applications with front end & back end connectivity.

Objectives:

1. To explore the essential phases and critical aspects of an overall software development process in order to design a high-quality software solution in cost-effective manner for a real-world problem.

Outcomes: On completion of the course, the learner will be able to:

1. Understand basic concepts of Software Engineering process and models.
2. Identify requirements, analyse and design for real world software projects.
3. Plan, schedule and track the progress of the project using various software project management tools.
4. Identify risks and create mitigation plans for dealing with real world risks, manage the configuration and change in software.
5. Apply testing principles on software projects.

Software Engineering (DJS23CCPC602)		
Unit	**Description**	**Duration**
1	**Introduction to Software Engineering and Process Models:** Nature of Software, Software Engineering, Software Process, CMM, Generic Process Model. **Prescriptive Process Models:** The Waterfall Model, V Model. **Incremental Process Model:** Incremental Model, RAD Model **Evolutionary Process Models:** Prototyping Paradigm, The Spiral Model **Concurrent Process Models:** Concurrent Process Model **Agile Methodology:** Agility Principals, Agile Process Models: Extreme Programming (XP), Adaptive Software Development (ASD), Dynamic Systems Development Method (DSDM), Scrum, Crystal, Feature Driven Development (FDD), Agile Modeling (AM), Kanban Model, LEAN models	08
2	**Requirement Analysis:** Requirement Elicitation, Software Requirement Specification (SRS). **Requirement Models:** Scenario Based Models, Class Based Models, Behavioural Models and Flow Models.	07
3	**Design Engineering and Analysis:** Design Principles, Design Concepts, Effective Modular Design-Cohesion and Coupling. **Translating the requirement models into the design model.** **Designs** Architectural Design, Component Level Design, User Interface Design.	06
4	**Software Project Management:** **Project Management Concepts:** Management Spectrum, 3Ps **Process and Project Metrics:**	08

	Metrics in the Process and Project Domains, software measurement, metrics for software quality. **Software Project Estimation:** LOC, FP, Empirical Estimation Models COCOMO I COCOMO II, Specialized Estimation Techniques. **Software Project Scheduling:** Work Breakdown Structure, Network Diagram, Gantt Chart, PERT, CPM, Stakeholders and Communication plan, Introduction to Project Management Information System (PMIS).	
5	**Software Risk Management:** Risk Identification, Risk Assessment, Risk Projection, Risk Refinement, RMMM Plan. **Software Configuration Management:** SCM, SCM Repositories, SCM Process, Change Control and Version Control.	05
6	**Software Testing Fundamentals:** Strategic Approach to Software Testing, Unit Testing, Integration Testing, Verification, Validation Testing, System Testing, Test Strategies for WebApps **Software Testing Techniques:** White Box Testing, Basis Path Testing, Control Structure Testing and Black Box Testing. TDD	05
	Total	**39**

Software Engineering Laboratory (DJS23CLPC602)	
Exp.	**Suggested experiments**
1	Prepare detailed statement of problem for the selected / allotted mini project and identify suitable process model for the same with justification.
2	Develop Software Requirement Specification (SRS) document in IEEE format for the project.
3	Identify scenarios & develop UML Use case and Class Diagram for the project.
4	Draw DFD (upto 2 levels) and prepare Data Dictionary for the project.
5	Develop Activity / State Transition diagram for the project.
6	Develop Sequence and Collaboration diagram for the project.
7	Use project management tool to prepare schedule and estimation for the project.
8	Prepare RMMM plan for the project.
9	Change specification and make different versions using any SCM Tool.
10	Develop test cases for the project using testing techniques.

Batchwise laboratory work of minimum 8 experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Batchwise tutorial sessions are to be conducted on topics which would help the learner to identify/analyze the problem and to apply problem solving techniques learnt.

Books Recommended:

Text Books:

1. Roger Pressman, "Software Engineering: A Practitioner's Approach", McGraw-Hill Publications 8th Edition, 2015

2. Ian Sommerville, "Software Engineering", Pearson Education 9th Edition, 2011
3. Ali Behfrooz and Fredeick J. Hudson, "Software Engineering Fundamentals", Oxford University Press.
4. Ugrasen Suman, "Software Engineering-Concepts and Practices", Cengage Learning, 2022

Reference Books:

1. Pankaj Jalote, "Software Engineering: A Precise Approach", Wiley Publications, 2010
2. Pankaj Jalote, "An integrated approach to Software Engineering", Springer, 2005
3. Jibitesh Mishra and Ashok Mohanty, "Software Engineering", Pearson, 2012
4. Rajib Mall, "Fundamentals of Software Engineering", Prentice Hall India, 4th edition, 2014

Prepared by Checked by Head of the Department Principal

Program: B. Tech. in Computer Engineering		T.Y B.Tech.	Semester: VI
Course: DevOps Laboratory (DJS23CLPC603)			

Pre-requisite:

1. Knowledge of Linux Operating system, installation and configuration of services and command line basics.
2. Software Development Life cycle.

Objectives:

1. To understand the fundamentals of DevOps engineering.
2. To be proficient with DevOps terminologies, concepts, benefits, and deployment options to meet real world software development requirements.

Outcomes: On completion of the course, the learner will be able to:

1. Interpret and apply various principles, phases and activities of Agile methodology and DevOps principles to meet software development requirements.
2. Understand and implement DevOps principles for CI/CD.
3. Apply testing process for application development and configuration management tools.

DevOps Laboratory (DJS23CLPC603)		
Unit	**Description**	**Duration**
1	**Introduction to DevOps:** Phases of Software Lifecycle, Minimum Viable Product (MVP) & Cross-functional Teams, Lean, ITIL, Agile development methodologies, DevOps as a prominent culture to achieve agility in the software development process, DevOps Stakeholders, Goals, DevOps and Agile, DevOps Tools.	02
2	**Version Control:** Introduction, Overview of Version Control Systems, Role of Version Control System, Types of Control Systems and their Supporting Tools, Importance of version control in CICD pipeline.	03
3	**Continuous Integration:** Introduction to Jenkins (With Master –Slave Architecture), Choosing a launch method, Administering Jenkins slaves, Labels, groups and load balancing. Creating Views and Jobs in Jenkins: The Jenkins user interface, Jobs in Jenkins, Creating Views, Managing Views and Jobs in Jenkins: Managing Views in Jenkins, Navigating a job's project page, Job Execution, The Job Execution Configuration Panel, The Status Panel, Console Panel.	06
4	**Continuous Deployment:** Overview of Docker, Benefits of Docker Workflow, Process Simplification, Architecture, Docker Containers, Docker Workflow, Anatomy of Dockerfile, Building an Image, Running an Image, Custom base Images, Storing Images.	04
5	**Continuous Testing:** Introducing WebDriver and WebElements, Selenium Testing Tools, Differences between Selenium 2 and Selenium 3, Setting up a project in Eclipse with Maven and TestNG using Java, WebElements, Locating WebElements using WebDriver, Interacting with WebElements, Different Available WebDrivers, Using Java 8 Features with Selenium. Introducing Java 8 Stream API, Using Stream API with Selenium WebDriver.	06
6	**Continuous Management:** The Parts of an Infrastructure System, Infrastructure Platforms, Infrastructure Resources, Compute Resources, Storage Resources, Network Resources.	05

	Puppet: Puppet Architecture, The Puppet Server, setting up the Puppet Agent, Performance Optimizations, Ansible: Ansible Architecture, Ansible and Infrastructure Management, Local Infrastructure Development: Ansible and Vagrant. Introduction to open-source tools for data gathering and management, AWS	
	Total	**26**

Web Programming Laboratory (DJS23CLPC504)	
Exp.	**Suggested experiments**
1	To understand Version Control System / Source Code Management, install git and create a GitHub account.
2	Perform various GIT operations on local and Remote repositories using GIT Cheat-Sheet.
3	To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job.
4	To Build the pipeline of jobs using Maven / Gradle / Ant in Jenkins, create a pipeline script to Test and deploy an application over the tomcat server.
5	To understand Jenkins Master-Slave Architecture and scale your Jenkins standalone implementation by implementing slave nodes.
6	To Setup and Run Selenium Tests in Jenkins Using Maven.
7	To understand Docker Architecture and Container Life Cycle, install Docker and execute docker commands to manage images and interact with containers.
8	To learn Dockerfile instructions, build an image for a sample web application using Dockerfile.
9	To install and Configure Pull based Software Configuration Management and provisioning tools using Puppet /Ansible.
10	To learn Software Configuration Management and provisioning using Puppet Blocks (Manifest, Modules, Classes, Function).
11	To provision a LAMP/MEAN Stack using Puppet Manifest.

Batchwise laboratory work of minimum **8** experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Books Recommended:

Text Books:

1. John Dean, “Web Programming with HTML5, CSS3 and JavaScript”, Jones & Bartlett Learning, 2019.
2. Karl Matthias & Sean P. Kane, “Docker: Up and Running”, 3rd Edition, O'Reilly Publication, 2022.
3. Craig Berg, “DevOps For Beginners: A Complete Guide To DevOps Best Practices” 2020.
4. Mikael Krief, “Learning DevOps: A comprehensive guide to accelerating DevOps culture adoption with Terraform, Azure DevOps, Kubernetes, and Jenkins”, Packt Publication, 2nd Edition, 2022.
5. Gene Kim, Jez Humble, et.el, “The DevOps Handbook: How to Create World-Class Agility,

6. Reliability, & Security in Technology Organizations", IT Revolution Press; 2nd edition 2021.
7. Mark Reed, "DevOps: The Ultimate Beginners Guide to Learn DevOps Step-By Step", LLC Publication, 2020.
8. Deepak Gaikwad, Viral Thakkar, "DevOps Tools from Practitioner's Viewpoint", Wiley, 2019.

Reference Books:

1. Mark S. Merkow, "Practical Security for Agile and DevOps", CRC Press Taylor & Francis, 2022.
2. Emily Freeman, "DevOps for Dummies", 3rd Edition, Wiley Publication, 2019.
3. Martin Alfke, Felix Frank, "Puppet 5 Essentials", 3rd Edition, O'Reilly Publication, 2017.

Prepared by Checked by Head of the Department Principal

Program: Computer Engineering **T.Y B.Tech.** **Semester: VI**

Course: Big Data Analytics (DJS23CCPE611)

Course: Big Data Analytics Laboratory (DJS23CLPE611)

Pre-requisite:

1. Database Management Systems
2. Data Warehousing and Mining

Objectives:

1. Analyze Big Data characteristics and evaluate appropriate analytical approaches for different business scenarios.
2. Design MapReduce solutions and implement stream processing techniques for large-scale data handling.
3. Construct Big Data mining solutions and apply analytics frameworks to solve industry problems.

Outcomes: On completion of the course, the learner will be able to:

1. Differentiate between various Big Data frameworks and choose appropriate tools for specific data processing needs.
2. Develop MapReduce programs and optimize stream processing algorithms for real-time data analysis.
3. Examine complex datasets and apply mining algorithms to extract meaningful patterns.
4. Create scalable Big Data applications and integrate recommendation systems and social network analytics.

Big Data Analytics (DJS23CCPE611)		
Unit	**Description**	**Duration**
1	**Introduction to Big Data and Big Data Analytics** What is Big Data?, Big Data Characteristics, Drivers, Big Data Characteristics, Traditional vs Big Data, Types of Big Data, Big Data Challenges, Examples of Big Data in Real Life, Big Data Applications. Introduction to Big Data Analytics, Types of Big Data Analytics, Big Data Analytics Process Models, Advantages of Big Data Analytics, Challenges in Big Data Analytics.	04
2	**Big Data Frameworks** **Hadoop**: Hadoop Core Components, Hadoop Ecosystem, Hadoop Physical Architecture, Working of Hadoop, Advantages and Limitations of Hadoop **HDFS**: Overview of HDFS, HDFS Architecture, HDFS Commands **Hive**: Features of Hive, Hive Architecture, Hive Vs Traditional Database, HiveQL **HBase**: Introduction to HBase, HBase Architecture, HBase Vs RDBMS, HBase Schema, Indexing in HBase, HBase Shell Queries **Pig**: Features, Execution Modes of Pig, Pig Vs SQL, Pig Architecture, Pig Data Model **Zookeeper**: Features, Architecture of Zookeeper, Working of Zookeeper	08
3	**MapReduce Paradigm** The Map Tasks, Grouping by Key, The Reduce Tasks, Combiners, Details of MapReduce Execution, Coping With Node Failures. Algorithms Using MapReduce: Word count by MapReduce, Matrix-Vector Multiplication by MapReduce,	04

	Relational-Algebra Operations, Computing Selections by MapReduce, Computing Projections by MapReduce, Union ,Intersection, and Difference by MapReduce	
4	**Mining Big Data Streams** **The Stream Data Model**: A DataStream-Management System, Examples of Stream Sources, Issues in Stream Processing, Sampling Data in a Stream: Sampling Techniques, Data Ingestion **Spark**: Features of Spark, Components of Spark, Architecture of Spark, RRD in Spark, SparkQL, Schedulers in Spark, Shared Variables in Spark **Kafka:** Features of Kafka, Components of Kafka, Kafka Cluster Architecture, Kafka Workflow **Filtering Streams**: The Bloom Filter **Counting Distinct Elements in a Stream**: The Count-Distinct Problem, The Flajolet-Martin Algorithm, **Counting Ones in a Window**: The Cost of Exact Counts, The Datar-Gionis-Indyk, Motwani Algorithm, Query Answering in the DGIM Algorithm.	08
5	**Big Data Mining Algorithms** **Frequent Pattern Mining Algorithms**: Handling Larger Datasets in Main Memory, Park, Chen, and Yu (PCY) Algorithm, The SON Algorithm and MapReduce. **Clustering Algorithms**: CURE Algorithm. Canopy Clustering, Clustering with MapReduce **Classification Algorithms**: Overview SVM classifiers, Parallel SVM, K-Nearest Neighbor classifications for Big Data, One Nearest Neighbor	08
6	**Big Data Analytics Applications** **Link Analysis**: PageRank Definition, Structure of the web, dead ends, Using Page rank in a search engine, Efficient computation of Page Rank, Topic sensitive Page Rank, Link Spam, Hubs and Authorities, HITS Algorithm. **Recommendation Engines**: A Model for Recommendation Systems, Content-Based Recommendations, Collaborative Filtering **Mining Social- Network Graphs**: Social Networks as Graphs, Types , Clustering of Social Network Graphs, Direct Discovery of Communities	07
	Total	**39**

Big Data Analytics Laboratory (DJS23CLPE611)	
Exp.	**Suggested experiments**
1	Installation of Hadoop on a single node cluster.
2	Execution of HDFS Commands.
3	Implementation of Map Reduce program to count words in a text file, matrix multiplication
4	Execute HIVE commands to load, insert, retrieve, update, or delete data in the tables.
5	Execute HBASE commands to perform basic CRUD operations and joins.
6	To create RDD, perform various operations and find occurrence of each word.
7	To create SparkQL and execute various SQL commands.
8	Perform Sentiment analysis using Spark Streaming.
9	Implementation of DGIM algorithm.

10	Implementation of PCY, SON algorithm.
11	Implementation of PageRank, HITS algorithm.
12	Implementation of Recommendation System.

Batchwise laboratory work of minimum 8 experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Batchwise tutorial sessions are to be conducted on topics which would help the learner to identify/analyze the problem and to apply problem solving techniques learnt.

Books Recommended:

Text Books:

1. Seema Acharya, Subhashini Chellappan, Big Data and Analytics, 2nd Edition, Wiley, 2022.
2. Radha Shankarmani, M. Vijayalakshmi, Big Data Analytics, 2nd Edition, Wiley, 2016.
3. Tom White, Hadoop: The Definitive Guide, 4th Edition, O'Reilly, 2015

Reference Books:

1. Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman, Mining of Massive Datasets, 3rd Edition, Cambridge University Press, 2020.
2. Rajkumar Buyya, Rodrigo N. Calheiros, Amir Vahid Dastjerdi, Big Data Principles and Paradigms, Morgan Kaufmann, 2016.
3. Vignesh Prajapati, Big Data Analytics with R and Hadoop, Packt Publishing Limited, 2013.

Prepared by Checked by Head of the Department Principal

Program: Computer Engineering **T.Y B.Tech.** **Semester: VI**

Course: Game Theory **(**DJS23CCPE612**)**

Course: Game Theory Laboratory **(**DJS23CLPE612**)**

Pre-requisite:

1. Analysis of Algorithms

Objectives:

1. To explain and predict how individuals behave in a specific strategic situation, and therefore help improve decision making
2. To explain in depth, the standard equilibrium concepts in Game Theory
3. To illustrate the concepts, real-world examples and case studies
4. To design Repeated Games with public information
5. To design static and Dynamic games with incomplete information

Outcomes: On completion of the course, the learner will be able to:

1. Identify strategic situations and represent them as games
2. Solve simple games using various techniques
3. Recommend and prescribe which strategies to implement
4. Develop Static and Dynamic Games
5. Develop Repeated Games

Game Theory (DJS23CCPE5612)		
Unit	**Description**	**Duration**
1	**Introduction to Game Theory** Games and solutions - Game theory and mechanism design - Examples from networks - Strategic form games - Matrix and continuous games - Iterated strict dominance - Rationalizability - Nash Equilibrium - existence and uniqueness - Mixed and correlated equilibrium – Super modular games - Potential/congestion games - Existence and Properties of Nash Equilibria.	07
2	**Extensive-Form Games** Definition - Strategies and Equilibria in Extensive Form Games - Backward Induction and Subgame Perfection and its Critiques.	04
3	**Repeated Games** Infinitely/finitely repeated games - Pareto Perfection and Renegotiation - Proofness in Repeated Games - Repeated Games with incomplete Public Information - Trigger strategies - Fork Theorem with Imperfect Public Information.	07
4	**Static Games with incomplete information** Mixed and Behavioral strategies - Bayesian Nash equilibrium - Applications in auctions - Different auction formats - Revenue and efficiency properties of different auctions - Bayesian Games	06
5	**Mechanism Design** Mechanism Design Principle - Single Agent - Several Agents, social choice functions and mechanisms, incentive compatibility and revelation theorem, the Gibbard-Satterthwaite impossibility theorem, quasilinear mechanisms, Vickrey- Clarke-Groves (VCG) mechanisms	08
6	**Dynamic Games with incomplete information** Introduction - Perfect Bayesian Equilibrium in Multi-stage games - Extensive-Form and Strategic-Form Refinements - Reputation Effects - Sequential Bargaining under Incomplete Information.	07
	Total	**39**

Game Theory Laboratory (DJS23CLPE612)	
Exp.	**Suggested experiments**
1	Construct and evaluate Nash equilibrium for Prisoner's Dilemma using nashpy, without using nashpy
2	Construct and evaluate the Game tree for the Battle of Sexes in Gambit
3	Construct and Evaluate Nash equilibrium for Hunting Game using nashpy, without using nashpy
4	Matching Pennies: Two payers, each having a penny, are asked to choose from among two strategies – heads (H) and tails (T). The row player wins if the two pennies match, while the column player wins if they do not. Find Nash Equilibrium nashpy, without using nashpy.
5	Routing congestion game: player 1 is interested in getting good service, hence would like theothers to choose different routes, while player 2 is interested only in disrupting player 1's serviceby trying to choose the same route. Find Nash Equilibrium nashpy, without using nashpy
6	SENATE RACE: An incumbent senator (from a rightist party) runs against a challenger (from a leftist party). They first choose a political platform, leftist or rightist, where the senator has to move first. If both choose the same platform, the incumbent wins, otherwise the challenger wins. Assume that the value of winning is 10, the value of compromising their political views (by choosing a platform not consistent with them) is 5, and the payoff is the sum of these values. Use Gambit to illustrate a Game tree.
7	BACKWARD INDUCTION: Six stones lie on the board. Black and White alternate to remove either one or two stones from the board, beginning with White. Whoever first faces an empty board when having to move loses. The winner gets $1, the loser loses $1. What are the best strategies for the players? Use Gambit to illustrate a Game tree
8	CENTIPEDE GAME: In this game, two players alternately face two stacks of money. To make a move, a player has the choice either to pass, in which case both stacks grow slightly and the other player now must make a move facing slightly larger stacks, or to take the larger stack, in which case the other player gets the smaller one and the game ends. If it didn't end before, the game ends after a fixed number of rounds, in which case both players share the accumulated money evenly. Use Gambit to illustrate a Game tree. Implement the CENTIPEDE GAME and find out Nash equilibrium
9	Mini Project

Batchwise laboratory work of minimum 08 experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Batchwise tutorial sessions are to be conducted on topics which would help the learner to identify/analyze the problem and to apply problem solving techniques learnt.

Books Recommended:

Text Books:

1. Ana Espinola-Arredondo, Felix Muñoz-Garcia, Game Theory-An Introduction with Step-by-Step Examples, Springer Nature Link, 2023.
2. Fudenberg, Drew, Jean Tirole, "Game Theory", Cambridge, MA: MIT Press, 1991
3. Osborne, M. J., and Rubinstein, A., "*A Course in Game Theory", Cambridge*, MA: MIT Press, 1994.
4. D. Fudenberg and J. Tirole, "*Game Theory*", The MIT Press, 2005

Reference Books:

1. Nisan, Noam, Tim Roughgarden, Eva Tardos, Vijay V. Vazirani, “Algorithmic Game Theory”, Cambridge, UK: Cambridge University Press, 2007.
2. Shoham, Y. and Leyton–Brown, K., "*Multiagent Systems: Algorithmic, Game Theoretic, and Logical Foundations"*. Cambridge University Press, 2008.

Prepared by Checked by Head of the Department Principal

Program: Computer Engineering **T.Y B.Tech.** **Semester: VI**

Course: Computer Vision (DJS23CCPE613)

Course: Computer Vision Laboratory (DJS23CLPE613)

Pre-requisite:

1. A sound knowledge of Computer Graphics and Image Processing.

Objectives:

1. To introduce fundamental concepts of computer vision and equip students with foundational knowledge in image formation, photometric processing, and geometric transformations.
2. To develop practical skills in feature extraction, recognition methodologies, and morphological image processing techniques for object segmentation and detection.
3. **To explore advanced applications in feature-based alignment, 3D vision, and recognition, including face detection, pose estimation, and scene understanding.**

Outcomes: On completion of the course, the learner will be able to:

1. Understand and apply key principles of image formation and photometric processing to analyze images and digital camera functions.
2. Implement and evaluate feature extraction techniques and morphological operations for conditioning, labeling, and matching objects in images.
3. Apply segmentation and detection algorithms to identify and isolate objects and regions of interest in complex images.
4. Demonstrate competence in advanced feature-based alignment, 3D object recognition, and scene understanding for real-world applications in computer vision.

Computer Vision (DJS23CCPE616)		
Unit	**Description**	**Duration**
1	**Introduction** What is computer Vision, Image Formation: Geometric Primitives, 2D transformation, 3 D transformation, 3D rotation, 3D to 2D Projection, Lens Distortion **Photometric Image Formation:** Lighting, reflectance and shading **The digital camera:** sampling and quantizition	04
2	**Recognition Methodology:** Conditioning, Labeling, Grouping, Extracting, Matching. **Morphological Image Processing:** Introduction, Dilation, Erosion, Opening, Closing,Hit-or-Miss transformation, Morphological algorithm operations on binary images, Morphological algorithm operations on gray-scale images, Thinning, Thickining , Region growing, region shrinking	06
3	**Feature Extraction and Model Fitting:** Edges - Canny, LOG, DOG, Line detectors (Hough Transform), Corners – Harris and Hessian Affine, Orientation Histogram, SIFT, SURF, HOG, GLOH, Deformation, RANSAC, Scale-Space Analysis- Image Pyramids and Gaussian derivative filters, Gabor Filters and DWT.	08

4	**Object Segmentation and Detection:** Active contours, Split and merge, Mean shift and mode finding, Normalized cuts, Graph cuts and energy-based methods, Application: Medical image segmentation.	08
5	**Feature-based alignment**: 2D and 3D feature-based alignment, Pose estimation, Geometric intrinsic calibration, Calibration patterns, Vanishing points, Application: Single view metrology, Rotational motion, Radial distortion.	06
6	**Recognition**: Object detection, Face detection, Pedestrian detection, Face recognition, 3D shape models, Application: Personal photo collections, Instance recognition, Category recognition, Context and scene understanding.	07
	Total	**39**

Computer Vision Laboratory (DJS23CLPE616)	
Exp.	**Suggested experiments**
1	Image Acquisition and Display
2	Image Transformations
3	Image Denoising and enhancement techniques
4	Feature Detection using OpenCV- Corner, Edge, Pyramid
5	Object Detection
6	Object Tracking
7	Pattern Recognition
8	Face Recognition
9	Perform medical image segmentation
10	Implementation of Image smoothing/ Image sharpening
11	Mini Project Based Learning
12	Research Article Review

Laboratory work of minimum 08 experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Books Recommended:

Text Books:

1. Richard Szeliski, Computer Vision: Algorithms and Applications, 2nd Edition, Springer Verlag London Limited, 2022.
2. Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing. Pearson, 4e, 2017.

Reference Books:

1. David Forsyth, Jean Ponce, Computer Vision: A Modern Approach, Pearson Education, 2nd Edition, 2015.
2. Katsushi Ikeuchi, Computer Vision: A Reference Guide, 2nd edition, Springer Publishing, 2021.
3. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
4. M. Sonka, V. Hlavac, and R. Boyle, Image Processing, Analysis, and Machine Vision, Thomson India Edition, 4e, 2017

Prepared by Checked by Head of the Department Principal

Program: Computer Engineering	T.Y B.Tech.	Semester: VI
Course: Advanced Network Design (DJS23CCPE621)		
Course: Advanced Network Design Laboratory (DJS23CLPE621)		

Pre-requisite: Data Structures, Computer Networks, Operating Systems

Course Objectives:

1. To develop a comprehensive understanding of advanced network design principles and theirstrategic implications.
2. To acquire proficiency in the practical implementation of network designs, including physicalinfrastructure, configurations, and lifecycle management.
3. To apply analytical skills to assess network data traffic, security threats, and performance metricsfor informed decision-making.

Outcomes: On completion of the course, learner will be able to:

1. Analyze the need for advanced networks and standards in various scenarios, evaluating specific design requirements for topology, cabling, and physical component placement.
2. Develop frameworks and standards for the implementation of network designs, incorporating bestpractices in configuration management. Also apply human factor concepts to enhance usability and troubleshoot network issues
3. Integrate IPv6 in network design, design and implement security systems, and analyze real-worldexamples of advanced network design. Also compare performance metrics in Internet routing andVoIP

Advanced Network Design (DJS23CCPE621)		
Unit	**Description**	**Duration**
1	**Introduction to Advanced Network Design:** Overview of Advanced Network Design Principles: Definition of advanced network design, Importance of strategic network planning Review of Basic Networking Concepts: OSI Model revisited, TCP/IP fundamentals, Network addressing and subnetting Emerging Trends in Networking: Software-Defined Networking (SDN), Network Function Virtualization (NFV), Internet of Things (IoT) in networking	05
2	**Physical Network Design and Infrastructure:** Physical Network Design: Topology design considerations, Cabling andphysical component placement, Redundancy and fault tolerance in physical design Advanced Router Configuration: In-depth configuration of routers, Routing protocols (EIGRP, OSPF, BGP), Router optimization and scalability	09
3	**Configuring and Managing the Network Infrastructure:** Network Configuration Best Practices: Implementation of network designs, Configuration management and version control Network Lifecycle Management: Maintenance and troubleshooting strategies, Network monitoring and performance optimization	06

4	**Analyzing Network Data Traffic:** Traffic Analysis Fundamentals: Packet capture and analysis tools, Understanding network protocols Quality of Service (QoS) Implementation, QoS requirements and strategies, Traffic classification and shaping	06
5	**Network Security and IPv6:** Threats and Vulnerabilities in Networks: Common network security threats, Vulnerability assessment and risk analysis Firewall and Intrusion Detection/Prevention Systems, Designing and placing firewalls strategically, Intrusion detection and prevention strategies Virtual Private Networks (VPNs) and IPv6: Implementing VPNs for security, Integration of IPv6 in network design	06
6	**Internet Routing and VOIP:** Internet Routing with BGP: BGP essentials and best practices, Internet routing considerations Voice over IP (VoIP) Basics: Introduction to VoIP technologies, Design considerations for VoIP in networks Case Studies and Best Practices: Examining real-world examples of advanced network design, best practices for implementing advanced network solutions	07
	Total	39

Advanced Network Design Laboratory (DJS23CLPE621)	
Sr. No.	**Suggested Experiment List**
1	Implement advanced network design principles through case study analysis.
2	Design a physical network layout using simulation tools like Cisco Packet Tracer or GNS3.
3	Configure routers for EIGRP, OSPF, and BGP, optimizing routing tables.
4	Set up a network infrastructure adhering to industry best practices.
5	Develop troubleshooting skills and perform routine maintenance tasks.
6	Analyze network traffic using tools like Wireshark.
7	Implement Quality of Service (QoS) strategies for network traffic.
8	Simulate and analyze common network security threats using tools like Metasploit.
9	Design and implement effective firewall and intrusion detection/prevention systems.
10	Configure and deploy Virtual Private Networks (VPNs) for secure communication

Laboratory work of minimum 08 experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Books Recommended:

Text Books:

1. Olivier Bonaventure, "Computer Networking: Principles, Protocols and Practice," No Starch Press,2013.
2. Kevin Dooley, "Designing Large Scale LANs," O'Reilly Media,2011.

3. Mani Subramanian, "Network Management: Principles and Practice," Addison-Wesley,2000.
4. Ilya Grigorik, "High-Performance Browser Networking," O'Reilly Media,2013.
5. William Stallings, "Network Security Essentials," Pearson,2016.
6. William A. Flanagan, "VoIP and Unified Communications: Internet Telephony and the Future Voice Network," Wiley,2012.

Reference books:

1. Laura Chappell, "Wireshark Network Analysis," Protocol Analysis Institute,2012.
2. Silvia Hagen, "IPv6 Essentials," O'Reilly Media,2006.
3. Iljitsch Van Beijnum, "BGP: Building Reliable Networks with the Border Gateway Protocol," O'Reilly Media,2002.
4. Tim Szigeti, "End-to-End QoS Network Design," Cisco Press,2004.
5. Jeffrey S. Beasley and Piyasat Nilkaew, "A Practical Guide to Advanced Networking," Pearson, 2017.
6. Gary A. Donahue, "Network Warrior," O'Reilly Media,2011.

Prepared by Checked by Head of the Department Principal

Program: Computer Engineering **T.Y B.Tech. Semester: VI**

Course: Distributed Databases (DJS23CCPE622)

Course: Distributed Databases Laboratory (DJS23CLPE622)

Pre-requisite:

1. Database Management Systems, ADBMS

Objectives:

1. To provide students with a comprehensive understanding of the design and management of distributed databases
2. To familiarize students with the technical challenges and solutions in distributed database systems

Outcomes: On completion of the course, the learner will be able to:

1. Design distributed databases, including data fragmentation and query processing, while optimizing for performance for real life applications.
2. To apply distributed transaction management protocols to ensure consistency and reliability in distributed systems under various failure conditions.

Distributed Databases (DJS23CCPE622)		
Unit	**Description**	**Duration**
1	**Concept and Overview of Distributed Database System**: What is Distributed Database System (DDBS), Features of DDBS, promises of DDBS, Design issue in DDBS, Distributed DBMS architecture: Client/server System, Peerto-Peer, MutliDatabase system, Types of Distributed Databases: Homogeneous vs. Heterogeneous	05
2	**Distributed Database Design**: Distributed database design concept, Objective of Data Distribution, Data Fragmentation (Horizontal, Vertical and Hybrid), Fragment Allocation, Transparencies in Distributed Database Design, Data Replication Concept and Techniques	07
3	**Overview of Query Processing**: Query Processing Problem, Objectives of Query Processing, Complexity of Relational Algebra Operations, Characterization of Query Processors, Layers of Query Processing **Query Decomposition and Data Localization**: Query Decomposition-Normalization, Analysis, Elimination of Redundancy, Rewriting, Localization concept	08
4	**Distributed Query Optimization**: Query Optimization, Join Ordering in Distributed Queries- Join Ordering, Semijoin Based Algorithms, Join versus Semijoin, Distributed Query Optimization – Dynamic Approach, Static Approach, Semijoin-based Approach, Hybrid Approach	08
5	**Distributed Transaction, Concurrency control and Recovery**:	07

	Distributed Transaction Management - Definition of a Transaction, Properties of Transactions, Model for Transaction management Distributed Concurrency control: Locking-Based Concurrency Control Algorithms - Distributed 2PL, Timestamp-Based Concurrency Control Algorithms Distributed Deadlock and Recovery: Distributed Deadlock prevention, avoidance, detection and recovery, Two-Phase and Three-Phase Commit Protocol.	
6	**Distributed Database Security and Emerging Trends**: Security Challenges in Distributed Databases, NoSQL distributed databases, NewSQL, and cloud-native databases, Database as a service (DBaaS)	04
	Total	**39**

Distributed Databases Laboratory (DJS23CLPE622)	
Exp.	**Suggested experiments**
1	Creation of centralized homogeneous database.
2	Perform Horizontal fragmentation and allocation in DDBS design.
3	Perform Vertical fragmentation and allocation in DDBS design.
4	Perform Hybrid fragmentation and allocation in DDBS design.
5	Implementation of Replication transparency.
6	Implementation of Query monitor (QEP- Query Execution Plan, Query Statistics)
7	Implementation of query optimization.
8	Implementations of deadlock detection in DDBS. (Use of table lock/unlock)
9	Simulation of two phase / three phases commit protocol.
10	Case study on Distributed Database Security and Emerging Trends.

Laboratory work of minimum 08 experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Books Recommended:

Text Books:

1. Chhanda Ray, "Distributed Database System", Pearson Education India, 2009
2. Korth, Silberchatz, Sudarshan, "Database System Concepts", 7th Edition, McGraw, 2019
3. Seed K. Rahimi and Frank S. Haug, "Distributed Database Management System", Wiley India, 2015

Reference Books:

1. M. Tamer Ozsu, Patrick Valduriez, "Principles of Distributed Database", Pearson Education India, 2020
2. Elmasri and Navathe, Fundamentals of Database Systems, 7th Edition, Pearson education, 2016

Prepared by Checked by Head of the Department Principal

Program: Department of Computer Engineering **T.Y B.Tech.** **Semester: VI**

Course: High Performance Computing (DJS23CCPE623)

Course: High Performance Computing Laboratory (DJS23CLPE623)

Pre-requisite:

1. IoT-Centric Processor Architecture and Organization.

Objectives:

1. To learn concepts of parallel processing as it pertains to high-performance computing.
2. To design, develop and analyse parallel programs on high performance computing resources
   using parallel programming paradigm.

Outcomes: On completion of the course, the learner will be able to:

1. Understand different parallel processing approaches and platforms involved in achieving High Performance Computing.
2. Explore Programming for High Performance Computing.
3. Understand the principles of Grid and Cloud Computing with practical examples and applications.
4. Analyse the performance measures in high performance computing.
5. Discover the advanced topic in GPU including libraries and framework.

High Performance Computing (DJS23CCPE623)		
Unit	**Description**	**Duration**
1	**Introduction to Parallel Processing:** Parallel processing, Levels of Parallelism, Models: SIMD, MIMD, SIMT, SPMD, Data Flow Models, Demand-driven Computation. Loosely coupled and tightly coupled. Parallel Architecture: Interconnection network, processor Array, Multiprocessor, Challenges in Parallel Computing, Performance Metrics, Distributed vs. Parallel architectures.	04
2	**Introduction to High Performance Computing:** Principles of HPC, HPC Architectures, HPC vs Parallel Processing, Data partitioning Techniques: Block, cyclic, and block-cyclic partitioning, Domain Decomposition: Spatial, temporal, and functional decomposition, Load balancing, Case Study: Partitioning strategies for matrix multiplication. Communication Models: Shared memory vs. message passing. Point-to-Point Communication: Send/Receive operations in MPI. Collective Communication: Broadcast, scatter, gather, and reduction operations in MPI (MPI_Reduce)	06
3	**GPU and CUDA Programming:** Overview of GPU, evolution of GPU, CPU vs. GPU, overview of CUDA: Features, Benefits, Architecture. Programming Model CUDA: Kernels and kernel launches, Thread and block indexing, CUDA Memory Management: Memory Hierarchy and Memory Management, Case Studies: simulation, data analytics and machine learning.	08

4	**Grid and Cloud Computing:** Data & Computational Grids, Grid Architectures and its relation to various Distributed Technologies, Examples of The Grid Computing, Cloud Computing, High Performance Cloud Computing, Cloud Tensor Processing Units (TPUs). Case studies: Analyzing impact of TPUs	07
5	**Performance Optimization:** Speedup, Efficiency and Scalability, Amdahl's Law, Gustafson's Law, Weak vs. Strong Scaling, Performance Bottlenecks, Data Races and Determinism, Data Race Avoidance, Profiling and performance analysis tools for GPUs, Techniques for optimizing GPU performance (warp divergence, loop unrolling, vectorization), Memory bandwidth optimization techniques, Advanced GPU programming concepts (shared memory atomics, warp shuffling). Case Studies: Scientific Computing with CUDA/Real-life application	08
6	**Advanced Topics in GPU:** Introduction to GPU accelerated libraries (cuBLAS, cuDNN, cuGraph), GPU computing frameworks (TensorFlow, PyTorch) and their integration with GPUs, Introduction to GPU clusters and distributed GPU computing, Cluster Setup & its Advantages. Case studies: Realworld applications of GPU computing.	06
	Total	**39**

High Performance Computing Laboratory (DJS23CLPE623)	
Exp.	**Suggested experiments**
1	Set up the CUDA environment, install the CUDA Toolkit, and write a basic CUDA program to understand the CUDA development environment.
2	Implement vector addition using CUDA to introduce students to parallelism, thread management, and memory allocation in GPU programming.
3	Develop a CUDA program for matrix multiplication to understand parallelism and optimization techniques in GPU computing.
4	Apply CUDA for image processing tasks, like blurring and edge detection, to learn how to process images efficiently using GPU parallelism.
5	Implement parallel reduction operations (e.g., sum, min, max) to grasp the concept of efficient parallel reduction.
6	Explore parallel sorting algorithms using CUDA, comparing their performance with CPU based sorting and optimizing CUDA sorting.
7	Employ CUDA for solving real-world problems to understand the power of GPU parallelism.
8	Experiment with CUDA to implement concurrent data structures using locks and atomic operations to learn how to manage data concurrently.
9	Optimize the reduction step in machine learning algorithms using CUDA, focusing on techniques for efficient large-scale data processing.
10	Integrate CUDA-accelerated code with data science frameworks like TensorFlow or PyTorch to develop and run GPU-accelerated machine learning models for practical applications.
11	Perform the Log Analysis-Based Resource and Execution Time Improvement

Laboratory work of minimum 08 experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Books Recommended:

Text Books:

1. "High Performance Computing in Clouds: Moving HPC Applications to a Scalable and Cost Effective Environment", Edson Borin, Lúcia Maria A. Drummond, Jean-Luc Gaudiot, Alba Melo, Maicon Melo Alves, Philippe Olivier Alexandre Navaux, Springer, ISBN-13 978-3031297687, 2023.
2. "High Performance Computing for Drug Discovery and Biomedicine", Alexander Heifetz, Springer Nature, ISBN, 1071634496, 9781071634493, 2023.
3. "Programming in Parallel with CUDA", Richard Ansorge, Cambridge University Press, ISBN-13 978-1108479530, 2022.
4. "Parallel and High Performance Computing", Robert Robey, Yuliana Zamora, Manning publisher, ISBN-13 978-1617296468, 2021
5. "The Practice of Parallel Programming", Sergey A. Babkin, CreateSpace Publisher ISBN-13:978-1451536614, Online Edition 2021.
6. "Hands-On GPU Programming with Python and CUDA" , Dr Brian Tuomanen, Packt Publishing, ISBN-13 978-1788993913, 2018.

Reference Books:

1. "Programming Massively Parallel Processors: A Handson Approach: David B. Kirk and Wenmei W. Hwu, Morgan Kaufmann, 4th Edition, 2022.
2. "CUDA by Example: An Introduction to General-Purpose GPU Programming", Jason Sanders and Edward Kandrot, Addison-Wesley, 1st Edition, 2010.
3. "Introduction to High Performance computing for Scientist and Engineers", Georg Hager, Gerhard Wellein,, CRC press, 2019.

Prepared by Checked by Head of the Department Principal

Program: B. Tech. in Computer Engineering **T.Y B.Tech.** **Semester: VI**

Course: Security in IoT Ecosystem (DJS23CCMD601)

Course: Security in IoT Ecosystem Laboratory (DJS23CLMD601)

Pre-requisite:

1. Computer Networks with IoT Fundamentals
2. IoT-Centric Processor Architecture and Organization

Objectives:

1. To understand and apply classic cryptographic techniques and modular arithmetic principles essential for modern cryptographic systems.
2. To gain knowledge of modern cryptographic algorithms, including symmetric and asymmetric encryption, and their practical applications in ensuring data confidentiality.
3. To explore mechanisms for ensuring integrity and availability in cryptographic systems, focusing on hashing, message authentication, and digital signatures.
4. To analyze network security protocols and IoT-specific authentication mechanisms that enhance security in connected environments.

Outcomes: On completion of the course, the learner will be able to:

1. Demonstrate proficiency in applying classic cryptographic techniques and modular arithmetic to solve cryptographic problems.
2. Implement and evaluate modern symmetric and asymmetric encryption algorithms to secure data transmission in various applications.
3. Utilize secure hash functions and digital signatures to ensure data integrity and authenticity in cryptographic communications.
4. Design and assess secure network architectures for IoT environments, implementing robust authentication protocols to safeguard user and device identities.

Security in IoT Ecosystem (DJS23CCMD601)		
Unit	**Description**	**Duration**
1	**Modular Arithmetic and Classic Cryptographic Techniques:** **Classic Cryptographic Techniques:** Caesar Cipher, Railfence Cipher, Vigenère Cipher, and Affine Cipher, One-Time Pad and Enigma Machine, Transposition Ciphers. **Modular Arithmetic and Number Theory:** Modular exponentiation, Fermat's Little Theorem and applications, Chinese Remainder Theorem, Euler's Theorem, Euclid's Algorithm, Groups , Rings and Fields.	04
2	**Modern Cryptographic Techniques** **Fundamentals:** Block Ciphers and Stream Ciphers, Block Cipher Modes of Operations. **Symmetric Encryption Algorithms:** Historical Symmetric Encryption Techniques (DES), Transition from DES to 3DES, Advanced Encryption Standard, ChaCha20, Lightweight Cryptography. **Asymmetric Encryption Algorithms:** RSA Algorithm, Elliptic Curve Cryptography, Curve 25519.	05
3	**Digital Trust Mechanisms:**	04

	Secure Hash Algorithms: Cryptographic Hashes, SHA-256 **Message Authentication Codes (MAC) and HMAC:** MAC, HMAC, CMAC **Digital Signatures:** DSA, ECDSA **Fault Tolerance and High Availability in Cryptographic Systems:** Fault Tolerance in Cryptography, Redundancy Mechanisms, Failover Systems	
4	**Network Security:** **Secure Network Protocols:** SSL/TLS, IPSec, HTTPS **Firewalls and Intrusion Detection Systems (IDS/IPS):** Firewalls,Intrusion Detection Systems (IDS), Prevention Systems (IPS) **VPNs and Zero Trust Architecture:** **Threat Detection with Machine Learning and AI**	04
5	**IOT Network Authentication:** **IoT Overview and Architecture:** IoT ecosystem and architecture components, Unique security challenges in IoT environments **Symmetric Key-Based Authentication in IoT:** Lightweight encryption methods for Wireless Sensor Networks (WSNs) **Asymmetric Key-Based Authentication Protocols (ECC in IoT):** ECC for secure key exchange in resource-constrained environments, Importance of ECC for energy efficiency in IoT devices **Quantum-Resistant Solutions for IoT:** Lattice-based cryptographic protocols for quantum resistance, Application of lattice-based algorithms in IoT for future-proof security	04
6	**IOT User and Device Level Authentication:** **Mutual Authentication Protocols:** Ensuring two-way authentication in IoT devices, Prevention of unauthorized access through mutual verification **Biometric-Based Robust Access Control Models:** Use of biometrics (fingerprints, facial recognition) for secure access, Challenges and advantages of biometric authentication in IoT **Gadget-Free Authentication:** Context-aware and password-less authentication methods, Use of geolocation, user behavior, and ambient data for seamless access **Physically Unclonable Function (PUF)-Based Authentication:** PUF for tamper-resistant security, Applications of PUF in IoT for unique **Hardware-Based Encryption and Secure Elements:** Secure hardware modules (TPM, HSM) for encrypted data storage, Key management and tamper-proof elements for enhanced device security	05
	Total	**26**

Security in IoT Ecosystem Laboratory (DJS23CLMD601)	
Exp.	**Suggested experiments**
1	**Modular Arithmetic Operations:** a) Create a program to perform modular exponentiation and demonstrate its application in Fermat's Little Theorem. b) Solve a set of problems using the Chinese Remainder Theorem and demonstrate how to recover the original number from modular equations.

2	**Implementation of Classic Ciphers:** a) Write a program to implement and encrypt/decrypt messages using the Caesar Cipher, Railfence Cipher, and Vigenère Cipher. b) Analyze the strengths and weaknesses of each cipher by attempting frequency analysis on sample ciphertexts.
3	**Symmetric Encryption Implementation:** a) Implement DES and AES encryption algorithms in a programming language of choice (e.g., Python). Test the performance and security of both algorithms using various input sizes. b) Explore the block cipher modes of operation (ECB, CBC, CFB) by implementing them and analyzing their impact on the security of the encrypted data.
4	**Asymmetric Encryption Experiment:** a) Write a program to implement the RSA algorithm and conduct key generation, encryption, and decryption. Evaluate its security based on key sizes. b) Implement Elliptic Curve Cryptography (ECC) for key exchange and demonstrate its use in a simple messaging application.
5	**Hash Function Analysis:** Implement SHA-256 and create a program to generate hash values for various input data. Analyze collision resistance by testing for any two different inputs that produce the same hash.
6	**Digital Signature Generation and Verification:** Implement the Digital Signature Algorithm (DSA) and create a workflow for signing a message and verifying the signature. Discuss potential vulnerabilities and how to mitigate them.
7	**Secure Network Protocols Testing:** a) Set up a secure communication channel using SSL/TLS and demonstrate a secure client-server application. Analyze the security features provided by the protocol. **b)** Configure and test a VPN solution, documenting the steps for secure remote access and discussing its advantages and limitations.
8	**Intrusion Detection System Setup:** a) Install and configure an Intrusion Detection System (IDS) such as Snort. Simulate different attack scenarios and observe how the IDS detects and responds to them. b) Create a simple firewall configuration to control incoming and outgoing traffic, testing its effectiveness against common network attacks.
9	**IoT Device Security Simulation:** a) Simulate a lightweight encryption method for securing data transmission in a Wireless Sensor Network (WSN) and evaluate its performance in terms of energy consumption and security. b) Implement an asymmetric key-based authentication protocol using ECC in an IoT simulation environment. Analyze the efficiency of key exchange in resource-constrained devices.
10	**Quantum-Resistant Protocol Evaluation:**

	Research and implement a basic lattice-based cryptographic protocol. Simulate its application in an IoT scenario and evaluate its effectiveness against potential quantum attacks.
11	**Mutual Authentication Protocol Implementation:** a) Design and implement a mutual authentication protocol between two IoT devices, documenting the process and evaluating its security effectiveness. b) Test the protocol against various attack vectors (e.g., replay attacks, man-in-the-middle attacks) to evaluate its robustness.
12	**Biometric Authentication System Development:** Create a simple biometric authentication system using facial recognition or fingerprint scanning. Test its effectiveness and discuss the challenges faced during implementation.

Batchwise laboratory work of minimum 8 experiments from the above suggested list or any other experiment based on syllabus will be included, which would help the learner to apply the concept learnt.

Books Recommended:

Text Books:

1. William Stallings, "Cryptography and Network Security: Principles and Practice", Pearson 2020, 8th Edition.
2. Behrouz A. Forouzan, "Cryptography and Network Security", McGraw-Hill Education, 2017, 4th Edition.
3. Madhusanka Liyanage, An Braeken, Pardeep Kumar, Mika Ylianttila, "IoT Security", Wiley, 2020

Reference Books:

1. Brucer Schneier, "Applied Cryptography: Porotocols, Algorithms and Source Code in C", Wiley, 2007,
2. Charles Pfleeger, Shari Lawrence Pfleeger & Jonathan Margulies, "Security in Computing", 5th Edition, Prentice Hall, 2018.

Prepared by Checked by Head of the Department Principal

Program: Computer Engineering **T.Y B.Tech.** **Semester: VI**

Course: Innovative Product Development IV (DJS23IPSCX04)

Course Objectives:

1. To acquaint the students with the process of identifying the need (considering a societal requirement) and ensuring that a solution is found out to address the same by designing and developing an innovative product.
2. To familiarize the students with the process of designing and developing a product, while they work as part of a team.
3. To acquaint the students with the process of applying basic engineering fundamentals, so as to attempt at the design and development of a successful value added product.
4. To inculcate the basic concepts of entrepreneurship and the process of self-learning and research required to conceptualise and create a successful product.

Outcomes:

Learner will be able to:

1. Identify the requirement for a product based on societal/research needs.
2. Apply knowledge and skills required to solve a societal need by conceptualising a product, especially while working in a team.
3. Use standard norms of engineering concepts/practices in the design and development of an innovative product.
4. Draw proper inferences through theoretical/ experimental/simulations and analyse the impact of the proposed method of design and development of the product.
5. Develop interpersonal skills, while working as a member of the team or as the leader.
6. Demonstrate capabilities of self-learning as part of the team, leading to life-long learning, which could eventually prepare themselves to be successful entrepreneurs.
7. Demonstrate product/project management principles during the design and development work and also excel in written (Technical paper preparation) as well as oral communication.

Guidelines for the proposed product design and development:

- Students shall form a team of 3 to 4 students (max allowed: 5-6 in extraordinary cases, subject to the approval of the department review committee and the Head of the department).
- Students should carry out a survey and identify the need, which shall be converted into conceptualization of a product, in consultation with the faculty supervisor/head of department/internal committee of faculty members.
- Students in the team shall understand the effective need for product development and accordingly select the best possible design in consultation with the faculty supervisor.
- Students shall convert the best design solution into a working model, using various components drawn from their domain as well as related interdisciplinary areas.
- Faculty supervisor may provide inputs to students during the entire span of the activity, spread over 2 semesters, wherein the main focus shall be on self-learning.

- A record in the form of an activity log-book is to be prepared by each team, wherein the team can record weekly progress of work. The guide/supervisor should verify the recorded notes/comments and approve the same on a weekly basis.
- The design solution is to be validated with proper justification and the report is to be compiled in a standard format and submitted to the department. Efforts are to be made by the students to try and publish a technical paper, either in the institute journal, "Techno Focus: Journal for Budding Engineers" or at a suitable publication, approved by the department research committee/ Head of the department.
- The focus should be on self-learning, capability to design and innovate new products as well as on developing the ability to address societal problems. Advancement of entrepreneurial capabilities and quality development of the students through the year long course should ensure that the design and development of a product of appropriate level and quality is carried out, spread over two semesters, i.e. during the semesters V and VI.

Guidelines for Assessment of the work:

- The review/ progress monitoring committee shall be constituted by the Head of the Department. The progress of design and development of the product is to be evaluated on a continuous basis, holding a minimum of two reviews in each semester.
- In the continuous assessment, focus shall also be on each individual student's contribution to the team activity, their understanding and involvement as well as responses to the questions being raised at all points in time.
- Distribution of term work marks during the subsequent semester shall be as given below:
    - Marks awarded by the supervisor based on log-book: 10
    - Marks awarded by review committee: 10
    - Quality of the write-up: 05

In the last review of the semester VI, the term work marks will be awarded as follows.

- Marks awarded by the supervisor (Considering technical paper writing): 15
- Marks awarded by the review committee: 10

Review/progress monitoring committee may consider the following points during the assessment.

- In the semester V, the entire design proposal shall be ready, including components/system selection as well as the cost analysis. Two reviews will be conducted based on the presentation given by the student's team.
    - First shall be for finalisation of the product selected.
    - Second shall be on finalisation of the proposed design of the product.

- In the semester VI, the expected work shall be procurement of components/systems, building of the working prototype, testing and validation of the results based on work completed in semester III.
    - First review is based on readiness of building the working prototype.
    - Second review shall be based on a presentation as well as the demonstration of the working model, during the last month of semester IV. This review will also look at the readiness of the proposed technical paper presentation of the team.

The overall work done by the team shall be assessed based on the following criteria;

1. Quality of survey/ need identification of the product.
2. Clarity of Problem definition (design and development) based on need.
3. Innovativeness in the proposed design.
4. Feasibility of the proposed design and selection of the best solution.
5. Cost effectiveness of the product.
6. Societal impact of the product.
7. Functioning of the working model as per stated requirements.
8. Effective use of standard engineering norms.
9. Contribution of each individual as a member or the team leader.
10. Clarity on the write-up and the technical paper prepared.

- The semester reviews (V and VI) may be based on relevant points listed above, as applicable.

Guidelines for Assessment of Semester Reviews:

- The write-up should be prepared as per the guidelines given by the department.
- The design and the development of the product shall be assessed through a presentation and demonstration of the working model by the student team to a panel of Internal and External Examiners, preferably from industry or any research organisations having an experience of more than five years, approved by the Head of the Institution. The presence of the external examiner is desirable only for the 2nd presentation in semester IV. Students are compulsorily required to present the outline of the technical paper prepared by them during the final review in semester VI.

Program: Computer Engineering **T.Y B.Tech.** **Semester: VI**

Course: Environmental Science Tutorial (DJS23ITHSX10)

Course:

Pre-requisite: Interest in Environment and its impact on Human

Objectives:

1.Familiarise students with environment related issues such as depleting resources, pollution, ecological problems and the renewable energy scenario.
2. Give overview of Green Technology options.

Outcomes: On completion of the course, learner will be able to:

1. Understand how human activities affect environment
2. Understand the various technology options that can make a difference

Detailed Syllabus: (Unit wise)		
Unit	**Description**	**Duration**
1	**Air Pollution** i. Air Quality Index ii. Case study on Smog	1
2	**Water Pollution** Presentation on Water Pollution (Industrial, Sewage, etc.) explaining any specific case.	1
3	**Noise Pollution** i. Decibel limits for hospital, library, silence zone ii. List effects of noise pollution on human health iii. Measure decibel level in college library, canteen, classroom	1
4	**Biodiversity loss** Case study on effect of pollution on biodiversity loss.	1
5	**Deforestation** Debate for and against "To promote Economic growth Deforestation is required."	1
6	**Renewable Energy sources** Presentation on different Renewable Energy Technologies	1
7	**Climate change** Report on major Impact of Global warming on Environment giving real examples**.**	1
8	**Green Technology** Advantages and Examples of Green Building for Sustainable development, Sustainable Software Design, Data Center Energy Efficiency, Thin-Client and Energy Efficiency.	1

Books Recommended:

Text books:

1. Environmental Studies From Crisis to Cure, R. Rajagopalan,
2. Textbook of Environmental Studies For Undergraduate Courses, Erach Bharucha
3. Green Information Technology A Sustainable Approach, Mohammad Dastbaz, Colin Pattinson, Babak Akhgar, Morgan and Kaufman, Elsevier.

***Reference Books**:*

1. Information Technologies in Environmental Engineering: New Trends and Challenges, Paulina Golinska, Marek Fortsch, Jorge Marx-Gómez, Springer, 2011.

Prepared by Checked by Head of the Department Principal