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: 2026-2027

Sr. No.	Course Code	Course	Teaching Scheme				Semester End Examination (SEE) - A						Continuous Assessment (CA) - B						Aggregate (A+B)	Credits Earned	
			Theory (Hrs)	Practical (Hrs)	Tutorial (Hrs)	Credits	Duration (Hrs)	Theory	Oral	Pract	Oral & Pract	SEE Total (A)	Term Test 1 (TT1)	Term Test 2 (TT2)	Term Test 3 (TT3)	Term Test Total (TT1 + TT2 + TT3)	Term Work	CA Total (B)			
1	DJS23CPC301	Machine Learning	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	4
	DJS23CPC301L	Machine Learning Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	
2	DJS23CPC302	Automata Theory and Compiler Design	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	4
	DJS23CPC302L	Automata Theory and Compiler Design Laborato	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	
3	DJS23CPC303	Business Intelligence and Analytics	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	4
	DJS23CPC303L	Business Intelligence and Analytics Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	
4	DJS23CPC304L	Web Programming Laboratory	--	2	--	1	2	--	--	--	25	25	--	--	--	--	25	25	50	1	1
#5	DJS23CPE301	Advanced Algorithms	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	4
	DJS23CPE301L	Advanced Algorithms Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	
	DJS23CPE302	Statistical Modeling	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	4
	DJS23CPE302L	Statistical Modeling Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	
	DJS23CPE303	Enterprise Data Systems	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	4
	DJS23CPE303L	Enterprise Data Systems Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	
	DJS23CPE304	Computer Graphics	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	4
	DJS23CPE304L	Computer Graphics Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	
6	DJS23CMD301	Information and Cyber Security	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	4
	DJS23CMD301L	Information and Cyber Security Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	
7	DJS23X5C301P	Innovative Product Development III	--	2	--	1	2	--	--	--	25	25	--	--	--	--	25	25	50	1	1
8	DJS23XHS311	Constitution of India	1	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
		Total	16	14	0	22	24	300	125	0	25	475	75	75	50	200	175	375	850	22	

# indicates only one subject

Sr. No.	Course Code	Course	Teaching Scheme				Semester End Examination (SEE) - A						Continuous Assessment (CA) - B						Aggregate (A+B)	Credits Earned	
			Theory (Hrs)	Practical (Hrs)	Tutorial (Hrs)	Credits	Duration (Hrs)	Theory	Oral	Pract	Oral & Pract	SEE Total (A)	Term Test 1 (TT1)	Term Test 2 (TT2)	Term Test 3 (TT3)	Term Test Total (TT1 + TT2 + TT3)	Term Work	CA Total (B)			
1	DJS23CPC351	Deep Learning	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	4
	DJS23CPC351L	Deep Learning Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	
2	DJS23CPC352	Blockchain and Crypto Technology	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	4
	DJS23CPC352L	Blockchain and Crypto Technology Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	
3	DJS23CPC353L	Software Engineering and DevOPs Laboratory	--	2	--	1	2	--	--	--	25	25	--	--	--	--	25	25	50	1	1
#4	DJS23CPE351	Big Data Analytics	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	4
	DJS23CPE351L	Big Data Analytics Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	
	DJS23CPE352	UI/UX Design	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	4
	DJS23CPE352L	UI/UX Design Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	
	DJS23CPE353	Virtual Reality	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	4
	DJS23CPE353L	Virtual Reality Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	
#5	DJS23CPE354	Computer Vision	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	4
	DJS23CPE354L	Computer Vision Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	
	DJS23CPE355	Development Frameworks	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	4
	DJS23CPE355L	Development Frameworks Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	
	DJS23CPE356	High Performance Computing	3	--	--	3	2	60	--	--	--	60	15	15	10	40	--	40	100	3	4
	DJS23CPE356L	High Performance Computing Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	
6	DJS23CMD351	Ethical Hacking and Digital Forensics	2	--	--	2	2	60	--	--	--	60	15	15	10	40	--	40	100	2	3
	DJS23CMD351L	Ethical Hacking and Digital Forensics Laboratory	--	2	--	1	2	--	25	--	--	25	--	--	--	--	25	25	50	1	
7	DJS23XSC351P	Innovative Product Development IV	--	2	--	1	2	--	--	--	25	25	--	--	--	--	25	25	50	1	1
8	DJS23XHS361T	Environmental Studies	--	--	1	1	--	--	--	--	--	--	--	--	--	--	25	25	25	1	1
		Total	14	14	1	22	24	300	125	0	50	475	125	125	50	200	200	400	875	22	

# indicates any one subject

Continuous Assessment (A):

Course	Assessment Tools	Marks	Time (mins)
Theory	a. Term Test 1 (based on 40 % syllabus)	15	45
	b. Term Test 2 (on next 40 % syllabus)	15	45
	c. Assignment / course project / group discussion / presentation / quiz/ any other.	10	--
	Total marks (a + b + c)	40	--
Audit course	Performance in the assignments / quiz / power point presentation / poster presentation / group project / any other tool.	--	As applicable
Laboratory	Performance in the laboratory and documentation.	25	
Tutorial	Performance in each tutorial & / assignment.	25	
Laboratory &Tutorial	Performance in the laboratory and tutorial.	50	

The final certification and acceptance of term work will be subject to satisfactory performance upon fulfilling minimum passing criteria in the term work / completion of audit course.

Semester End Assessment (B):

Course	Assessment Tools	Marks	Time (hrs.)
Theory / * Computer based	Written paper based on the entire syllabus.	60	2
	* Computer based assessment in the college premises.		
Oral	Questions based on the entire syllabus.	25	As applicable
Practical	Performance of the practical assigned during the examination and the output / results obtained.	25	2
Oral & Practical	Project based courses - Performance of the practical assigned during the examination and the output / results obtained. Based on the practical performed during the examination and on the entire syllabus.	As per the scheme	2

Prepared by Checked by Head of the Department Principal

Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)

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

Course: Machine Learning (DJS23CPC301)

Course: Machine Learning Laboratory (DJS23CPC301L)

Pre-requisite: Data Structures, Basic Probability and Statistics, Algorithms, Data Mining

Objectives:

The objective of the course is

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. Apply dimensionality reduction techniques.
3. Apply various supervised and unsupervised machine learning algorithms.
4. Design application using machine learning techniques

Machine Learning (DJS23CPC301)		
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	**07**
2	**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)	**07**
3	**Dimensionality Reduction** Dimensionality Reduction Techniques: Principal components analysis (Eigen values, Eigen vectors, orthogonality), LDA, Singular value decomposition,	**07**
4	**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, KNN algorithm	**08**
5	**Clustering** Basics of clustering, Hard *vs.* Soft Clustering, Density-based Clustering:	**07**

	DBSCAN, Expectation maximization (EM) algorithm for soft clustering, Semi-supervised learning with EM using labelled and unlabeled data, Radial Basis functions.	
6	**Applications of Machine Learning** Recommendation Systems, Machine Learning for Image Recognition, Sentiment Analysis, Machine Learning for Video Surveillance, Autonomous Vehicles, Healthcare Diagnostics, Agriculture	**06**
	Total	**42**

Machine Learning Laboratory (DJS23CPC301L)	
Exp.	**Suggested experiments (Any 8)**
1	To implement Linear Regression
2	To implement Logistic Regression
3	Implementing CART decision tree algorithm
4	To implement Support Vector Machine
5	To implement Bayesian Classification
6	To implement PCA
7	To implement K-Nearest Neighbor
8	Implementing Radial basis functions
9	Mini project based on any machine learning application

Any other experiment based on syllabus may be included, which would help the learner to understand topic/concept.
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:

Textbooks:

1. Ethem Alpaydın, Introduction to Machine Learning, 4th Edition, The MIT Press 2020
2. Peter Harrington, "Machine Learning in Action", 1st 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", 1st Edition, O'reilly 2016
5. Kevin P. Murphy, "Probabilistic Machine Learning: An Introduction" MIT Press, 2022
6. Wenmin Wang, "Principles of Machine Learning: The Three Perspectives" Springer 2024

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. H. Dunham, "Data Mining: Introductory and Advanced Topics", 1st Edition, Pearson Education, 2006.

Online Resources:

1. NPTEL - Introduction to Machine Learning (Link: https://nptel.ac.in/courses/106106139)

Prepared by | Checked by | Head of the Department | Principal

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

Course: Automata Theory and Compiler Design (DJS23CPC302)

Course: Automata Theory and Compiler Design Laboratory (DJS23CPC302L)

Pre-requisite: Algebraic Number Theory

Objectives:

The objective of the course is

1. To introduce the fundamental concepts of formal languages, grammar, and automata theory.
2. Introduce the major concepts of language translation and compiler design

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. Design relevant automata for different classes of formal languages.
3. Analyze problems and understand the difference between decidable and undecidable problems.
4. Acquire fundamental understanding of the structure of a Compiler.

Automata Theory and Compiler Design (DJS23CPC302)		
Unit	**Description**	**Duration**
1	Fundamentals: Formal Languages, Strings, Alphabets, Languages, Chomsky Hierarchy of languages Finite Automata: Introduction to Finite State machine, Acceptance of strings and languages, Deterministic finite automaton (DFA) and Non-deterministic finite automaton (NFA), Equivalence of NFA and DFA – Equivalence of NFAs with and without ϵ-moves, Minimization of finite automata, Equivalence between two DFA's, Finite automata with output – Moore and Mealy machines, conversion of Moore to Mealy and Mealy to Moore	**10**
2	Regular Languages: Regular expressions, Identity rules, Conversion of a given regular expression into a finite automaton, Conversion of finite automata into a regular expression, Pumping lemma for regular sets, Closure properties of regular sets (proofs not required)	**05**
3	Context Free Grammars: Context free grammars and languages, Derivation trees, Leftmost and rightmost derivation of strings and Sentential forms, Ambiguity, left recursion and left factoring in context free grammars, Simplification of context free grammars, Normal forms for context free grammars, Chomsky normal form, Greibach normal form.	**07**
4	Pushdown Automata: Introduction to Pushdown automata, Acceptance of context free languages, Acceptance by final state and acceptance by empty state and its equivalence, Equivalence of context free grammars and pushdown automata	**06**
5	Turing Machine: Introduction to Turing Machine, Design of Turing Machines, Types of Turing Machines.	**08**

	Undecidability: Recursive and recursively enumerable languages, halting problem, Post correspondence problem	
6	**Introduction to Compilers:** Phases of a Compiler. Lexical Analysis: The Role of Lexical Analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens. Syntax Analysis: The role of the Parser, First and Follow, Predictive Parsing, LR Parsers- Intermediate Code Generation: Intermediate Languages- Graphical Representations, three address code, Implementations.	06
	Total	**42**

Automata Theory and Compiler Design Laboratory (DJS23CPC302L)	
Exp.	**Suggested experiments (Any 8)**
1	Design and implement a program to simulate a Deterministic Finite Automaton (DFA) that accepts all strings over the alphabet Σ={a,b}.
2	Given a Non-deterministic Finite Automaton (NFA) with ϵ -transitions, write a program to convert it into an equivalent NFA without ϵ-transitions, then to a DFA, and finally minimize the resulting DFA to its smallest possible number of states.
3	Program to generate Regular Expression for a given finite automata.
4	Program should take a CFG as input and output the equivalent, simplified grammar in Chomsky Normal Form, showing the steps of the conversion process.
5	Design a DFA that recognizes valid identifiers in the C
6	Write a program that generates the parse tree (or derivation) for a given valid arithmetic expression string, demonstrating the structure defined by your grammar. The program should also be able to identify and reject invalid expressions.
7	To create and simulate automata using JFLAP.
8	Case Study: LLVM for machine learning inference with Google's TensorFlow

Any other experiment based on syllabus may be included, which would help the learner to understand topic/concept.
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:

Textbooks:

1. John E Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, "Introduction to Automata Theory, Languages and Computation", Third Edition, Pearson, 2006.
2. Alfred V.Aho, Monica S.Lam,Ravi Sethi, Jeffrey D. Ullman, "Compilers Principles,

Techniques and Tools", 2nd Edition Pearson, 2014.
3. M. Sipser, "Introduction to the Theory of Computation," Cengage Learning, 3/e, 2012.

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", 3 rd Edition, PHI, 2012.
3. Peter Linz, "An introduction to Formal Languages and Automata ", 6th Edition, Narosa Publishers, 2017.

Online Resources:

1. NPTEL: Introduction to Automata, Languages and Computation (Link: https://nptel.ac.in/courses/106105196)

Prepared by | Checked by | Head of the Department | Principal

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

Course: Business Intelligence and Analytics (DJS23CPC303)

Course: Business Intelligence and Analytics Laboratory (DJS23CPC303L)

Pre-requisite: DBMS, Data Structures, Algorithms, Statistics and Probability

Objectives:

The objective of the course is

1. To introduce the concepts, architecture, and components of Business Intelligence systems.
2. To develop understanding of data warehousing, OLAP, and data mining techniques.
3. To explore business analytics and data visualization tools for decision-making
4. To examine advanced BI technologies and their applications in real-world domains.

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

1. Understand the architecture, components, and applications of Business Intelligence systems.
2. Design data warehouse using dimensional modelling and apply OLAP operations for any application.
3. Apply data mining and analytical techniques for business decision-making.
4. Develop dashboards and visual analytics solutions using modern BI tools.

Business Intelligence and Analytics (DJS23CPC303)		
Unit	**Description**	**Duration**
1	**Introduction to Business Intelligence** Definition, need, and scope of Business Intelligence, BI vs Traditional Decision Support Systems, Framework for BI: BI Architecture and Components, Role of BI in strategic decision-making, Business analytics overview, Introduction to Big Data analytics	**05**
2	**Data Warehousing and OLAP** Data Warehousing Concepts and Architecture, Schema Design: Star, Snowflake, Fact Constellation, ETL Process: Extraction, Transformation, and Loading, OLAP: Different Types of OLAP OLAP Operations: Roll-up, Drill-down, Slice, Dice, Pivot.	**06**
3	**Data Mining for BI** Introduction to Data Mining and Knowledge Discovery Process, Data Preprocessing: Cleaning, Transformation, Normalization Association Rule Mining: Apriori, FP-Growth Classification: Decision Tree, Naïve Bayes Clustering: K-Means, Hierarchical	**10**

	Model Evaluation and Validation: Confusion Matrix, Accuracy, Precision, Recall, F1-score, ROC Curve	
4	**Business Analytics** Concept and Types of Analytics: Descriptive, Predictive, Prescriptive Predictive Modeling and Forecasting Techniques: Regression, Time Series Analysis, Handling Time Series, Regression Based Forecasting: A model with Seasonality, A model with Trend and Seasonality, ARIMA models Prescriptive Analytics: What-if analysis, Decision Analysis with Decision Tables and Decision Trees.	08
5	**Visualization** Data Visualization: Importance, Principles, Different types of Charts and Graphs, The emergence of data visualization and Visual Analytics Performance Dashboards: Dashboard Design, Best practices, KPIs, Business Performance Management, Balanced Scorecards, Six Sigma as a Performance Measurement System: Balanced Scorecard Versus Six Sigma	07
6	**BI and emerging trends** Business Intelligence Applications in marketing, sales, finance, operations, Emerging Trends: Recommendation Engines, Cloud Computing and BI, Text Analytics, Web Analytics, Social Analytics	06
	Total	**42**

Business Intelligence and Analytics Laboratory (DJS23CPC303L)	
Exp.	**Suggested experiments (Any 8)**
1	Study of BI architecture and components using case studies.
2	Design a Star and Snowflake Schema for a business scenario.
3	ETL implementation using Pentaho or Talend on sample datasets.
4	Perform OLAP operations (roll-up, drill-down, slice, dice) using SQL.
5	Apply association rule mining using Apriori Algorithm
6	Implement classification algorithms (Decision Tree/Naïve Bayes). Evaluate the classifier.
7	Perform clustering using K-Means and visualize clusters.
8	Design an interactive dashboard using Power BI or Tableau.
9	Analyze web/social media data for trends or sentiment using Python.
10	Mini Project: Develop an end-to-end BI solution using open datasets.

Any other experiment based on syllabus may be included, which would help the learner to understand topic/concept.

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:

Textbooks:

1. Ramesh Sharda, Dursun Delen, Efraim Turban – Business Intelligence, Analytics, and Data Science: A Managerial Perspective on Analytics, 5th Edition., Pearson, 2023, ISBN-13 9780137931361.
2. Paulraj Ponniah, "Data Warehousing: Fundamentals for IT Professionals", 2nd Edition, Wiley India, 2013.
3. Jiawei Han, Micheline Kamber, Jian Pei – Data Mining: Concepts and Techniques, 3rd Edition., Morgan Kaufmann (Elsevier), 2011, ISBN-13 9780123814791.
4. Galit Shmueli, Peter C. Bruce, Peter Gedeck, Nitin R. Patel, "Data Mining for Business Analytics: Concepts, Techniques and Applications in Python", Wiley India, ISBN: 9781119549840, 1st Edition, 2019

Reference Books:

1. Cindi Howson – Successful Business Intelligence: Unlock the Value of BI and Big Data, 2nd Edition., McGraw-Hill, 2013.
2. Thomas H. Davenport & Jeanne Harris – Competing on Analytics: Updated with New Examples, Harvard Business Review Press, 2017, ISBN-13 9780071809184.
3. Foster Provost & Tom Fawcett – Data Science for Business, 1st Edition., O'Reilly Media, 2013, ISBN-13 9781449361327.
4. Anil Maheshwari – Data Analytics Made Accessible, 2nd Edition., 2019, Amazon Digital Services, 2019, ISBN-13 9789355324559
5. R.N Prasad, Seema Acharya: Fundamentals of Business Analytics, Wiley 2nd Edition,2019.
6. U. Dinesh Kumar: Business Analytics: The Science of Data-Driven Decision Making, Wiley, 2nd Edition, 2021. Forecasting techniques
7. Regi Mathew: Business Analytics for Decision Making, Pearson 1st Edition, 2020
8. Ralph Kimball, Margy Ross – The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, 3rd Edition., Wiley, 2013, ISBN-13 9781118530801.

Online Resources:

1. Regression Modeling Fundamentals
   https://www.coursera.org/learn/regression-modeling-sas
2. "Predictive Modeling with Logistic Regression using SAS"
   https://www.coursera.org/learn/sas-predictive-modeling-using-logistic-regression

Prepared by | Checked by | Head of the Department | Principal

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

Course: Web Programming Laboratory (DJS23CPC304L)

Pre-requisite: Object Oriented Programming using Java, HTML

Objectives:
The objective of the course is
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. Design interactive web page(s) using CSS3 and JavaScript.
2. Design Single Page Application using React.js and Node.js Framework.
3. Develop web based Node.js applications using Express.
4. Apply MongoDB for frontend and backend connectivity using REST API.

Web Programming Laboratory (DJS23CPC304L)		
Unit	**Description**	**Duration**
1	**HTML5, CSS3, Tailwind:** **HTML5:** Overview, Study of AI tools assisting creation of HTML5 web pages. **CSS3:** Introducing CSS3, CSS Preprocessors (SASS/SCSS), Selectors, Box Model, Border, Margin & Padding, Background Images & Colors and Other Decorative (Texts, Fonts, Links, Lists, Tables), Positioning, Combinators, Pseudo-class and Pseudo-element, 2D and 3D Transformations, Transitions and Animations, Flexbox, Media Queries. **Tailwind:** Introduction, setting up Tailwind, Folder structure and build setup, utility-classes, Layout, Flex and Grid, Responsive design, customization, components and plugins.	02
2	**JavaScript:** Introduction to JavaScript, JavaScript DOM Model, var, let, const, Operators, primitive data types & strings, conditional, loop, 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.	04
3	**React Fundamentals:** 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.	06

	Refs, Use effects, useActionState, useFormStatus, and useOptimistic, Hooks, Flux.	
4	**Node.js:** Node.js, Setup Development Environment: Installation of Node.js, Creating simple Node Server, Request and Response, Routing responses, MVC architecture in Node.js Event Loop and Emitters, File System Interaction, Modules, Native Node drivers.	04
5	**Express.js:** Introduction, Installation, Express router, REST API (CRUD Operations), Generator, Authentication, sessions, Integrating with React.	06
6	**Database Connectivity:** MongoDB Installation, connecting to MongoDB, Frontend Integration with React, User Authentication (JWT), Role-based Access Control, connecting MongoDB to Node-RED, Hosting of web application.	06
	Total	**28**

Web Programming Laboratory (DJS23CPC304L)	
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 a 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	Build an interactive JavaScript application implementing CRUD logic. Use DOM manipulation, array methods (map, filter, reduce), and localStorage for data persistence.
5	Create an application to demonstrate JSX, Components, Props, and State in React.
6	Create an application to demonstrate Forms, Events, Routers, Refs, Keys in React.
7	Create an application to demonstrate the 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 the 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 using RESTful API. Deploy website on live webserver and access through URL.

Books Recommended:

Textbooks:

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.

Online Resources:

1. The Complete JavaScript Course 2025: From Zero to Expert! (Udemy Course).
2. React - The Complete Guide 2025 (incl. Next.js, Redux) (Udemy Course).
3. https://www.coursera.org/professional-certificates/ibm-full-stack-cloud-developer

Prepared by | Checked by | Head of the Department | Principal

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

Course: Advanced Algorithms (DJS23CPE301)

Course: Advanced Algorithms Laboratory (DJS23CPE301L)

Pre-requisite:

1. Data structures
2. Analysis of Algorithms

Objectives:

The objective of the course is to provide conceptual and practical knowledge of Advanced Algorithms

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

1. Analyze 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 (DJS23CPE301)

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	06
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	10
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.	08
5	**Computational Geometry**: Line Segment Properties, Convex Hull Graham's scan algorithm. Online Algorithms: Competitive Ratio, K-Server Special topic: Reward-guided Tree search (e.g, reasoning with LLM)	06

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	**08**
	Total	**42**

Advanced Algorithms Laboratory (DJS23CPE301L)	
Exp.	**Suggested experiments (Any 8)**
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 (KD Tree Operations)
6	To implement Graph Based Algorithms (Ford Fulkerson Method)
7	To implement Graph Based Algorithms (Push Relable Method)
8	To implement Computational Geometry (Graham Scan Algorithm)
9	To implement Online Algorithms (K-Server algorithm)
10	To implement Approximation Algorithm (Approximate TSP implementation)

Any other experiment based on syllabus may be included, which would help the learner to understand topic/concept.
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:

Textbooks:

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.

Online Resources:

1. Beyond worst-case analysis | Communications of the ACM
2. Introduction to Approximation Algorithms
3. [2411.11694] Enhancing LLM Reasoning with Reward-guided Tree Search

Prepared by | Checked by | Head of the Department | Principal

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

Course: Statistical Modeling (DJS23CPE302)

Course: Statistical Modeling Laboratory (DJS23CPE302L)

Pre-requisite: Probability

Objectives:
The objective of the course is

1. To understand fundamental concepts of statistics, probability distributions, and hypothesis testing for analyzing engineering data.
2. To develop skills in building and validating regression models for prediction and inference in computational applications.
3. To master time series analysis techniques including ARIMA modeling, forecasting, and advanced methods for sequential data analysis.
4. To apply statistical modeling techniques using Python/R to solve real-world problems in computer engineering domains.

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

1. Apply descriptive and inferential statistical techniques to interpret engineering datasets.
2. Evaluate linear and multiple regression models for prediction tasks in software and computational systems.
3. Design time series models (ARIMA, SARIMA) to forecast sequential data in diverse computing applications.
4. Create end-to-end statistical modeling solutions using Python/R for real-world data analysis, system monitoring, and predictive analytics.

Statistical Modeling (DJS23CPE302)		
Unit	**Description**	**Duration**
1	**Fundamentals of Statistics** Introduction to Statistical Analysis: Role of statistics in computer engineering and data science, Types of data: categorical, numerical, time-ordered, Population vs. Sample Descriptive Statistics: Measures of central tendency: mean, median, mode; Measures of dispersion: variance, standard deviation, IQR Statistical Inference: Confidence intervals, Hypothesis testing: t-tests (one-sample, two-sample), p-values and significance levels, Chi-square test for independence	**07**
2	**Correlation and Regression Models** Correlation Analysis: Pearson correlation coefficient, Spearman rank correlation Simple Linear Regression: Least squares estimation, Regression equation and interpretation, Coefficient of determination (R^2), Residual plots: normality, homoscedasticity, independence Multiple Linear Regression: Multiple regression model formulation, Matrix notation and parameter estimation, Adjusted R^2 and F-statistic	**07**

	Interpretation of regression coefficients, Multicollinearity: Variance Inflation Factor (VIF) and detection, Model selection criteria: Akaike Information Criterion (AIC), the Bayesian Information Criterion (BIC), and Mallows's Cp Advanced Regression Methods: Polynomial regression, Ridge (L2 norm) and LASSO (L1 norm) regression (regularization), Logistic regression for binary classification (overview)	
3	**Introduction to Time Series and Decomposition** Time Series Fundamentals: Definition and characteristics of time series, Examples in computer engineering, Time series vs. cross-sectional data, Time series components: trend, seasonal, cyclical, irregular Classical Decomposition: Additive vs. multiplicative models, Moving average decomposition, Seasonal decomposition methods, STL decomposition Smoothing Techniques: Simple moving average (SMA), Weighted moving average (WMA)	**07**
4	**ARIMA Modeling and Forecasting** Concept of stationarity: weak (covariance) stationarity, Importance in time series modeling, Testing for stationarity: Augmented Dickey-Fuller (ADF) test, Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test Transformations: Logarithmic and Box-Cox transformation, Differencing: first-order, second-order, Seasonal differencing Autocorrelation Analysis: Autocorrelation Function (ACF), Partial Autocorrelation Function (PACF), Interpretation of ACF and PACF plots ARMA and ARIMA Models: Autoregressive (AR) models: AR(p), properties and stationarity, Parameter estimation, Model order selection using PACF Moving Average (MA) models: MA(q), Properties and invertibility, Parameter estimation, Model order selection using ACF ARMA(p,q) models: properties and identification, ARIMA(p,d,q) models for non-stationary series Box-Jenkins Methodology: Model identification, Parameter estimation: Maximum Likelihood Estimation, Diagnostic checking: Ljung-Box test, residual analysis, Forecasting and prediction intervals Model selection: AIC, BIC, AICc criteria	**09**
5	**Seasonal Time Series and Model Evaluation** Seasonal ARIMA (SARIMA): Seasonal patterns and periodicity detection, SARIMA(p,d,q)s notation, Seasonal differencing, Model identification for seasonal data, Parameter estimation Model Evaluation: Forecast accuracy measures: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), Mean Absolute Scaled Error (MASE), Training vs. test set evaluation, Cross-validation for time series	**06**
6	**Advanced Topics and Real-World Applications of Time Series Data** Intervention analysis and outlier detection, Multivariate time series:	**06**

	Vector Autoregression (VAR), Volatility modeling: ARCH/GARCH models Machine learning for time series: LSTM networks (overview), Facebook Prophet (overview), Automated ARIMA (auto.arima, pmdarima) Real-World Applications: Network traffic prediction and capacity planning, System performance monitoring and anomaly detection, Resource utilization forecasting (CPU, memory, bandwidth), Server load prediction and balancing, IoT sensor data analysis and predictive maintenance, Cloud computing: workload prediction and auto-scaling, Cybersecurity: intrusion detection patterns	
	Total	**42**

Statistical Modeling Laboratory (DJS23CPE302L)	
Exp.	**Suggested experiments (Any 8)**
1	To compute statistical measures and create visualizations for data exploration. • Load dataset and compute mean, median, mode, variance, standard deviation • Create histograms, box plots, and scatter plots • Identify outliers using box plots and statistical methods • Generate correlation matrix and heatmap
2	To perform statistical tests and understand inference. • Compute confidence intervals for mean • Perform one-sample t-test • Perform two-sample t-test (independent samples) • Conduct chi-square test for independence • Interpret p-values and draw conclusions
3	To analyze relationships between variables and build simple regression models. • Calculate Pearson and Spearman correlation coefficients • Fit simple linear regression model • Compute R^2 and interpret results • Create scatter plot with regression line • Analyze residuals
4	To build and validate multiple regression models. • Build multiple regression model with multiple predictors • Interpret regression coefficients • Compute adjusted R^2 and F-statistic • Perform residual analysis • Split data into training and testing sets • Evaluate model performance on test data
5	To check regression assumptions and detect multicollinearity. • Check normality of residuals (Q-Q plot, Shapiro-Wilk test) • Test for homoscedasticity (residual vs. fitted plot) • Calculate VIF to detect multicollinearity • Identify influential observations (Cook's distance)

	• Apply remedial measures if needed
6	To implement advanced regression techniques. • Fit polynomial regression for non-linear relationships • Implement Ridge regression • Implement LASSO regression • Compare models using cross-validation • Visualize results and compare performance
7	To decompose time series and apply smoothing techniques. • Load and visualize time series data • Perform seasonal decomposition (additive and multiplicative) • Plot trend, seasonal, and residual components • Implement Simple Exponential Smoothing (SES) • Implement Holt's double exponential smoothing • Implement Holt-Winters triple exponential smoothing • Generate short-term forecasts and evaluate
8	To test for stationarity and analyze autocorrelation structure. • Plot time series and check for trends/seasonality • Perform Augmented Dickey-Fuller (ADF) test • Perform KPSS test • Apply transformations (log, Box-Cox) • Apply differencing (first-order, seasonal) • Plot and interpret ACF and PACF
9	To build ARIMA models using Box-Jenkins methodology. • Identify ARIMA model order using ACF and PACF • Fit ARIMA model using statsmodels/pmdarima • Perform diagnostic checking: • Residual analysis • Ljung-Box test • ACF of residuals • Compare multiple ARIMA models using AIC/BIC • Generate forecasts with 95% confidence intervals • Calculate forecast accuracy metrics (MAE, RMSE, MAPE)
10	Complete time series analysis on real-world dataset. • Select dataset with seasonal patterns (e.g., network traffic, energy consumption, sales) • Perform complete exploratory data analysis (EDA) • Test for stationarity and apply necessary transformations • Build SARIMA model • Compare SARIMA with other models (ARIMA, exponential smoothing) • Generate multi-step forecasts • Visualize actual vs. predicted values • Prepare technical report with: i. Introduction and problem statement ii. Data description and EDA iii. Methodology

	iv.	Results and model comparison
	v.	Forecasts and visualizations
	vi.	Conclusions and future work

Any other experiment based on syllabus may be included, which would help the learner to understand topic/concept.
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:

Textbooks:

1. James, G., Witten, D., Hastie, T., & Tibshirani, R. "An Introduction to Statistical Learning with Applications in R", 2nd Edition, Springer, 2021.
2. Hyndman, R. J., & Athanasopoulos, G.. "Forecasting: Principles and Practice". 3rd Edition, OTexts, 2021.
3. VanderPlas, J., "Python Data Science Handbook", 2nd Edition, O'Reilly Media, 2023.

Reference Books:

1. Shumway, R. H., & Stoffer, D. S., "Time Series Analysis and Its Applications: With R Examples", 4th Edition, Springer, 2019.
2. Cryer, J. D., & Chan, K. S., "Time Series Analysis: With Applications in R", 3rd Edition, Springer, 2023.
3. McKinney, W., "Python for Data Analysis", 3rd Edition, O'Reilly Media, 2023.
4. Tsay, R. S., "An Introduction to Analysis of Financial Data with R", 2nd Edition, Wiley, 2024.

Online Resources:

1. https://www.coursera.org/learn/practical-time-series-analysis
2. https://nptel.ac.in/courses/111105091
3. https://onlinecourses.swayam2.ac.in/ini24_cs02/preview

Prepared by | Checked by | Head of the Department | Principal

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

Course: Enterprise Data Systems (DJS23CPE303) Advanced Database Management Systems

Course: Enterprise Data Systems Laboratory (DJS23CPE303L)

Pre-requisite: Basic knowledge of Database Management System

Objectives:

The objective of the course is

1. To provide an overview of enterprise data systems, platforms, and challenges.
2. To understand the usage of advanced data models and architectures for real life applications
3. To impart knowledge of enterprise data security and optimization.

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

1. Describe enterprise data systems, platforms, and identify challenges.
2. Explain enterprise data models, architectures and specialized databases
3. Optimize and secure enterprise data systems.

Enterprise Data Systems (DJS23CPE303)

Unit	Description	Duration
1	**Introduction to Enterprise Data System** Definition, components, features, factors for choosing enterprise data platforms, enterprise data platforms: NoSQL, Multimedia database, Native XML databases (NXD), Object Oriented Database, Federated Databases, Mobile databases.	06
2	**Enterprise Data Models and Architectures** Temporal data models: Aspects of valid time, transaction time, and bi-temporal time with examples of each. Spatial model: Types of spatial data models - Raster, Vector and Image Graph Database: Introduction, Graph database architecture, Types of graph database, Graph vs Relational database, Data modeling with graph, Neo4j. Distributed Database: Introduction, Fragmentation and its types.	08
3	**Document Oriented Database** Need of Document Oriented database, difference between Document Oriented Database and Traditional database, Types of encoding XML, JSON, BSON, Representation XML, JSON Objects. Case study on document oriented based such as MongoDB.	06
4	**Optimization** Query Optimization: Overview, Measures of Query cost, Selection operation, Sorting and Join Operations, Evaluation of expression query, Translations of SQL Queries into relational algebra, Heuristic approach and cost-based optimization	08
5	**Data Security** Introduction to Database Security Issues; Authentication and	08

	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	
6	**Challenges in Enterprise Data System** Data fragmentation, data quality issues, complex data integration, governance and compliance, scaling with data growth, change management and user adoption, Resource constraints.	06
	Total	**42**

Enterprise Data Systems Laboratory (DJS23CPE303L)	
Exp.	**Suggested experiments (Any 8)**
1	Perform a case study on enterprise database system: Summary and Comparison – Consider any existing enterprise database system. – Identify the strengths and weaknesses. – Recommend changes and/or enhancements.
2	Query execution on the XML database. Use any XML editor (eg XMLSPY) to show query execution in XML.
3	Implement an enterprise data system based on spatial and temporal models.
4	Data handling using JSON. Using JSON, exhibit how data can be stored and queried.
5	Implementation of simple graph database using Neo4j. Create nodes, properties, and relationships.
6	Perform Fragmentation in DDBS design. Design a distributed database for a case study of your choice. Apply fragmentation. (Range, List, Hash and Key)
7	Implementation of simple document-oriented database using MongoDB. Store data using MongoDB for any appropriate case study which uses a document-oriented database.
8	Implementation of Query monitor (QEP- Query Execution Plan, Query Statistics) In MySQL, use the Query monitor to understand the query execution plan and monitor the query statistics.
9	Simulate Query optimization by applying an SQL Query on any database. Apply the optimization techniques for query processing to select the query with the least processing time.
10	Optimization using B/B+ Tree. Write a program in C++/Java/python to analyze the impact of B/B+ tree in searching records.
11	Case study on challenges in enterprise data systems. Study a research paper / patent / product. Identify gaps Suggest Improvement/s Draft & apply for journal / conference / book / book-chapter / magazine / patent / copyright.

Any other experiment based on the syllabus may be included, which would help the learner to

understand topic/concept.
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:

Textbooks:

1. Ole Olesen-Bagneur, "Enterprise Data Catalog", 1st edition, O'Reilly Publication, 2023.
2. Abraham Silberschatz, Henry F. Korth, Sudarshan, "Database System Concepts", 7th Edition, Mc Graw Hill, 2021.
3. Shannon Bradshaw, Eoin Brazil, "MongoDB: The Definitive Guide - Powerful and Scalable Data Storage", 3rd 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

Online Resources:

1. Neo4j documentation https://neo4j.com/docs/

Prepared by | Checked by | Head of the Department | Principal

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

Course: Computer Graphics (DJS23CPE304)

Course: Computer Graphics Laboratory (DJS23CPE304L)

Pre-requisite: C Programming

Objectives:

The objective of the course is

1. The objective of the course is to equip students with the fundamental knowledge of computer graphics and provide an understanding of how to scan convert the basic geometrical primitives, how to transform the shapes to fit them as per the picture definition.
2. Provide an understanding of mapping from a world coordinate to device coordinates, clipping, solid modeling, rendering, and projections.

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

1. Understand the fundamental knowledge of computer graphics applications of computer Graphics.
2. Apply and compare the algorithms for drawing 2D images and explain aliasing, anti-aliasing and half toning techniques.
3. Analyze and apply clipping algorithms and transformation on 2D images.
4. Understand the basic of shading, shadows, curves and surfaces and also solve the problems of curves.
5. Understand and apply fundamental OpenGL concepts—including its features, abstractions, and 3D viewing pipeline—to develop basic interactive graphics programs.

Computer Graphics (DJS23CPE304)		
Unit	**Description**	**Duration**
1	**Introduction to Computer Graphics:** Overview of Computer Graphics, Computer Graphics Application and Software, Description of some graphics devices, Input Devices for Operator Interaction, Active and Passive Graphics Devices, Display Technologies, Storage Tube Graphics Displays, Calligraphic Refresh Graphics Displays, Raster Refresh (Raster-Scan) Graphics Displays, Cathode Ray Tube Basics, Color CRT Raster Scan Basics, Video Basics, The Video Controller, Random Scan Display Processor, LCD displays.	06
2	**Scan conversion: lines, circles and Ellipses and Filling polygons** Scan Converting Lines, Mid-point criteria, Problems of Aliasing, Scan Converting Circles, Scan Converting Ellipses, Filling Polygons, edge data structure	08
3	**Two-Dimensional and Three Dimensional Transformations:** Transformations and Matrices, Transformation Conventions, 2D Transformations, Homogeneous Coordinates and Matrix Representation of 2D Transformations, Translations and Homogeneous Coordinates, Rotation, Reflection, Scaling, Combined Transformation, Transformation of Points, Transformation of The Unit Square, Solid Body	07

	Transformations, Rotation About an Arbitrary Point, Reflection through an Arbitrary Line, A Geometric Interpretation of Homogeneous Coordinates. **Three-Dimensional Transformations:** Scaling, Shearing, Rotation, Reflection, Translation, Multiple Transformation, Rotation about an Arbitrary Axis in Space, Reflection through an Arbitrary Plane, Matrix Representation of 3D Transformations, Composition of 3D Transformations, Affine and Perspective Geometry, Perspective Transformations, Techniques for Generating Perspective Views, Vanishing Points, the Perspective Geometry and camera models, Orthographic Projections, Axonometric Projections, Oblique Projections	
4	**Two-Dimensional Viewing:** Introduction, Viewing Pipeline View Coordinate reference frame, Window to viewport transformation point clipping, Text Clipping, Line Clipping: Cohen Sutherland Algorithm, Liang Barsky algorithms, Polygon clipping: Sutherland Hodgeman polygon clipping and Weiler Atherton. Fractal Geometry: Fractal Dimension, Koch Curve. Piano Curve, Hilbert Curve.	**08**
5	**Visible-Surface Determination** Techniques for efficient Visible-Surface Algorithms, Categories of algorithms, Back face removal, The z-Buffer Algorithm, Scan-line method, Painter's algorithms (depth sorting), Area sub-division method, BSP trees, Visible-Surface Ray Tracing, comparison of the methods. **Illumination and Shading** Illumination and Shading Models for Polygons, Reflectance properties of surfaces, Ambient, Specular and Diffuse reflections, Atmospheric attenutation, Phong's model, Gouraud shading, some examples	**08**
6	**Graphics Programming using OPENGL** Why OpenGL, Features in OpenGL, OpenGL operations, Abstractions in OpenGL – GL, GLU & GLUT, 3D viewing pipeline, viewing matrix specifications, a few examples and demos of OpenGL programs	**05**
	Total	**42**

Computer Graphics Laboratory (DJS23CPE304L)	
Exp.	**Suggested experiments (Any 8)**
1	Implementation of Line Drawing algorithms: DDA, Bresenham's and using them generating line with different styles like dotted, dashed, centered and thick line.
2	Implementation of Circle generation algorithm: Midpoint and using it generating concentric circles.
3	Implementation of Area Filling Algorithm: Boundary Fill, Flood Fill and Scan line, Polygon Fill.
4	Curve Generation: Bezier for n control points, B Spline (Uniform), Fractal Generation (Koch Curve)

5	Program for performing Two Dimensional Transformations: Translation, Scaling, Rotation, Reflection, Shear by using a homogeneous Matrix representation, use of a function for matrix multiplication is desirable, to perform composite transformation
6	Implementation of Line Clipping Algorithm: Cohen Sutherland, Liang Barsky.
7	Implementation of Polygon Clipping Algorithm: Sutherland Hodgman.
8	Program to represent a 3D object using polygon surfaces and then perform 3D transformation.
9	Program to perform projection of a 3D object on Projection Plane: Parallel and Perspective.
10	Implement Illumination and shading apply on sphere using two light sources in OpenGL

Any other experiment based on syllabus may be included, which would help the learner to understand topic/concept.
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.

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:

Textbooks:
1.A. P. Godse, Dr. D. A. Godse, Computer Graphics and Multimedia, Concepts, Algorithms and Implementation using C, KDP Print US ,2020.
2. Donald Hearn and M. Pauline Baker, "Computer Graphics", Pearson Education,2011.

Reference Books:
1Computer Graphics and Multimedia: Concepts, Algorithms and Implementation using C, Technical publications,2020.
2. William M. Newman, "Principles of Interactive Computer Graphics", 2001.
B.M. Havaldar ,"C Graphics and Projects", 2006.

Online Resources:
1. https://onlinecourses.nptel.ac.in/noc20_cs90/preview
2. https://www.edx.org/learn/computer-graphics/the-university-of-california-san-diego-computer-graphics

Prepared by | Checked by | Head of the Department | Principal

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

Course: Information and Cyber Security (DJS23CMD301)

Course: Information and Cyber Security Laboratory (DJS23CMD301L)

Pre-requisite: Computer Networks, Database Management System, Algebraic Number Theory

Objectives:
The objective of the course is to:

1. Understand the fundamental concepts of cybersecurity, including security principles, classical cryptography, and modern symmetric and asymmetric encryption techniques.
2. Explain authentication, integrity, and secure communication mechanisms such as digital signatures, PKI, hashing, SSL/TLS, and network defense tools.
3. Identify and analyze system, network, and application-level vulnerabilities and attacks, along with corresponding defense techniques and secure software practices.
4. Gain awareness of modern cybersecurity domains, including malware analysis, system hardening, security operations (SOC), incident response, and AI-enabled security mechanisms.

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

1. Apply cryptographic techniques to achieve data confidentiality, integrity, and authentication.
2. Identify and assess systems and networks against security threats and attacks using appropriate defense mechanisms.
3. Analyze software and web applications to identify vulnerabilities and recommend appropriate secure development practices.
4. Understand security governance, system hardening, incident response, SOC operations, and legal and regulatory compliance.

Information and Cyber Security (DJS23CMD301)		
Unit	**Description**	**Duration**
1	**Cyber Security Essentials and Classical Cryptography:** CIA Triad, security goals, threats, vulnerabilities, attacks, Security services & mechanisms, OSI security architecture, network security model, Classical encryption: substitution (Caesar, Vigenère) and transposition ciphers, Security models: Bell-LaPadula, Biba, Zero Trust Model, Risk management & threat intelligence, Malware types: Viruses, Worms, Trojans, Ransomware, Logic Bombs, Rootkits, Bots, Social Engineering Attacks: Phishing, Spear Phishing, Baiting, Tailgating.	06
2	**Symmetric Cryptography & Secure Communication:** Symmetric encryption basics, block vs stream ciphers, DES, Double DES, Triple DES, AES, Simplified AES, Block cipher modes: ECB, CBC, CFB, OFB, CTR, Applications of symmetric cryptography in SSL/TLS,	08

	VPNs, disk encryption	
3	**Asymmetric Cryptography, Digital Signatures & PKI**: Symmetric vs asymmetric encryption, RSA algorithm (conceptual without number theory), Diffie–Hellman key exchange, Elliptic Curve Cryptography (ECC – concept only), ECC-based DH, Digital signatures: RSA, ElGamal, DSA, Public Key Infrastructure (PKI), X.509 digital certificates, HTTPS, SSL/TLS handshake overview, Introduction to quantum-safe cryptography	**08**
4	**Integrity, Authentication & Network Security Basics:** Cryptographic hash functions: MD5, SHA-1, SHA-256, SHA-3, Message Authentication Codes: MAC, HMAC, CMAC, User & entity authentication: passwords, biometrics, multi-factor authentication, Authentication protocols: Needham–Schroeder, Kerberos, Network attacks: ARP spoofing, IP spoofing, packet sniffing, DNS spoofing, TCP SYN flood, Firewalls, IDS, IPS, VPN basics, Wi-Fi security: WEP, WPA2, WPA3	**07**
5	**Application, Web & Software Security:** Secure OS design, access controls: DAC, MAC, RBAC, Software & web vulnerabilities: Buffer overflow, SQL injection, XSS, CSRF, Format string attacks, OWASP Top 10 (latest list), Secure Software Development Life Cycle (SSDLC), Introduction to DevSecOps and security in CI/CD pipelines, Container security basics (Docker, Kubernetes)	**06**
6	**Security Operations, Compliance, and Quantum Cryptography:** Basic System Hardening Techniques: Patch management, secure configuration, password policies, Security Policies and Compliance (GDPR, DPDPA 2023 - basic awareness), Security Operations Center (SOC) – Introduction, Incident Response: Preparation, Detection, Containment, Recovery (NIST model), Basics of SIEM (Security Information and Event Management), Introduction to AI in security (SOC automation, threat detection), Introduction to Quantum Cryptography: Basics of quantum computing threats, Quantum Key Distribution (QKD), post-quantum cryptography fundamentals.	**07**
	Total	**42**

Information and Cyber Security Laboratory (DJS23CMD301L)	
Exp.	**Suggested experiments (Any 8)**
1	Implement Caesar Cipher (encryption & decryption)
2	Implement Vigenère Cipher & Transposition Cipher
3	Implement DES and AES Encryption (using any programming language or tool like Cryptool/OpenSSL)
4	Implement RSA Algorithm (key generation, encryption, decryption – conceptual coding without number theory)

5	Implement Diffie–Hellman Key Exchange (simulate shared key generation)
6	Implement Digital Signature (using RSA or DSA) and verify integrity
7	Demonstrate Hashing Algorithms (MD5, SHA-1, SHA-256) and HMAC verification
8	Packet sniffing and protocol analysis using Wireshark (capture HTTP/HTTPS, ARP, DNS traffic)
9	Simulate ARP Spoofing or MITM attack using tools like Ettercap/Scapy and analyze using Wireshark
10	Demonstrate Firewall configuration / Simple IDS using Snort (detect port scan or ping sweep)
11	Demonstrate SQL Injection attack and its prevention
12	Demonstrate XSS (Cross-Site Scripting) or CSRF attack on a test web page
13	Buffer Overflow attack simulation in C/Python and mitigation using safe functions / DEP
14	Study of Malware & Incident Response: Analyze a malware sample (safe simulation), observe SOC tools/SIEM logs, prepare an incident response report (NIST phases)

Any other experiment based on syllabus may be included, which would help the learner to understand topic/concept.
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:

Textbooks:

1. Stallings, W., *Network Security Essentials: Applications and Standards*, 6th ed. Boston, MA: Pearson, 2023.
2. B. A. Forouzan and D. Mukhopadhyay, *Cryptography and Network Security*, 3rd ed. McGraw Hill, 2015.
3. Whitman, M. E.; Mattord, H. J., *Principles of Information Security*, 7th ed. Boston, MA: Cengage, 2024

Reference Books:

1. C. Kaufman, R. Perlman, and M. Speciner, *Network Security: Private Communication in a Public World*, 2nd ed. Pearson, 2015.
2. Chapple, M.; Stewart, J. M.; Gibson, D., *(ISC)² CISSP Certified Information Systems Security Professional Official Study Guide*, 9th ed. Indianapolis, IN: Sybex, 2024.
3. R. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, 3rd ed. Wiley, 2020.
4. M. Sikorski and A. Honig, *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*, No Starch Press, 2012.

Online Resources:

1. "National Institute of Standards and Technology (NIST) – Computer Security Resource Center." Available: https://csrc.nist.gov.
2. "OWASP Foundation – Open Web Application Security Project." Available: https://owasp.org.
3. "MIT OpenCourseWare – Computer and Network Security." Massachusetts Institute of Technology, Available: https://ocw.mit.edu.
4. "CyberChef – The Cyber Swiss Army Knife," GCHQ, Available: https://gchq.github.io/CyberChef/.
5. "Wireshark Network Protocol Analyzer – Official Documentation," Available: https://www.wireshark.org.
6. "Snort – Network Intrusion Detection System," Cisco, Available: https://www.snort.org.

Prepared by | Checked by | Head of the Department | Principal

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

Course: Innovative Product Development III (DJS23XSC301P)

Course:

Pre-requisite:

Objectives:

The objective of the course is

1. To acquaint the students with the process of identifying the need (considering a societal requirement) and ensuring that a solution is found 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 conceptualize and create a successful product.

Outcomes: On completion of the course, the 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 conceptualizing 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 analyze 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: o Marks awarded by the supervisor based on log-book: 10 o Marks awarded by review committee: 10 o 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 finalization of the product selected.
  - Second shall be on finalization 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 organizations 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.

Prepared by | Checked by | Head of the Department | Principal

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

Course: Constitution of India (DJS23XHS331)

Pre-requisite:

Objectives:

The objective of the course is

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.

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 (DJS23XHS331)		
Unit	**Description**	**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.	03
6	**Scope & Aims of Engineering Ethics:**	02

	Responsibility of Engineers and Impediments to Responsibility. Risks, Safety and liability of Engineers. Honesty, Integrity & Reliability in Engineering.	
	Total	**14**

Books Recommended:

Textbooks:

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

Online Resources:

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

Prepared by | Checked by | Head of the Department | Principal

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

Course: Deep Learning (DJS23CPC351)

Course: Deep Learning Laboratory (DJS23CPC351L)

Pre-requisite: Artificial Intelligence, Machine Learning

Objectives:

The objective of the course is

1. To understand the Hyper Parameter Tuning.
2. To explore Deep Learning Techniques with different learning strategies.
3. To design Deep Learning Models for real time applications.

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

1. Understand the fundamentals and tools of Deep Learning.
2. Apply Hyper-parameter Tuning techniques.
3. Design and analyze Convolutional and Sequential Models.
4. Develop and evaluate Generative and Adversarial Networks.
5. Investigate suitable Deep Learning models for real-world applications.

Deep Learning (DJS23CPC351)		
Unit	**Description**	**Duration**
1	**Introduction to Deep Learning:** Overview of Neural Network, Deep learning and human brain, Factors Contributing to the Rise of Deep Learning, Deep Learning applications Overview of Tools: Torch, TensorFlow, Keras	04
2	**Convolutional Neural Network:** Introduction to CNNs, Kernel filter, Principles behind CNNs, Multiple Filters, CNN applications **ConvNet Architectures** Discussions on famous convnet architectures: AlexNet, VGG, GoogLeNet, ResNet	09
3	**Hyper-parameter Tuning, Batch Normalization** Tuning Process, Using an Appropriate Scale to pick Hyper-parameters, Hyper-parameters Tuning in Practice: Pandas vs. Caviar, Normalizing Activations in a Network, Fitting Batch Norm into a Neural Network, why does Batch Norm work, Batch Norm at Test Time	06
4	**Sequential models:** Introduction to Sequence Models and RNNs, Recurrent Neural Network Model, Backpropagation Through Time, Different Types of RNNs: Unfolded RNNs, Seq2Seq RNNs, Long Short-Term Memory (LSTM), Bidirectional RNN, Vanishing Gradients with RNNs, Gated Recurrent Unit (GRU), RNN applications, Sequence models and attention mechanism, Attention over Images, Hierarchical Attention	09

5	**Adversarial Networks** Introduction to adversarial Networks, Auto encoders (standard, de-noising, contractive, etc.), Variational Auto encoders, Generative Adversarial Networks, Diffusion Models: Basics of Diffusion models, Advantages of over GANs, Popular Diffusion tools: Stable Diffusion, DALL-E, Mid-journey, Applications of Adversarial Networks	09
6	**Deep Learning Case Studies:** Image Processing, Natural Language Processing, Speech Recognition, Video Analytics	05
	Total	**42**

Deep Learning Laboratory (DJS23CPC351L)	
Exp.	**Suggested experiments**
1	To implement backpropagation algorithm.
2	Understanding ANN using Pytorch/Tensor Flow.
3	Visualizing Convolutional Neural Network using Pytorch/Tensor Flow.
4	Object detection using RNN using Pytorch/Tensor Flow.
5	Handwritten Digit Generation using GANs
6	Students are supposed to complete any one mini project not limited to following list of projects. 1. Sequence Prediction 2. Object Detection 3. Traffic Sign Classification 4. Automatic Music Generation 5. Music Genre Classification 6. Text Summarizer 7. Gender and Age Detection Using Voice 8. Chatbot Using Deep Learning 9. Neural Style Transfer 10. Face Aging 11. Driver Drowsiness Detection 12. Language Translator 13. Image Reconstruction

Any other experiment based on syllabus may be included, which would help the learner to understand topic/concept.
Batchwise laboratory work of minimum 5 experiments from the above suggested list along with a mini project based on syllabus will be included, which would help the learner to apply the concept learnt.

Books Recommended:
Textbooks:

1. Aggarwal, C., Neural Networks and Deep Learning: A Textbook, 2nd Edition,

Springer, 2023
2. Goodfellow, I., Bengio,Y., and Courville, A., Deep Learning, MIT Press, 2016.
3. Umberto Michelucci, Advanced Applied Deep Learning: Convolutional Neural Networks and Object Detection, 2019.
4. Michael Nielsen (Goodreads Author), "Neural Networks and Deep Learning",2015

Reference Books:

1. Chollet, F., and Watson, M., Deep Learning with Python, 3rd Edition, Manning Publications, 2025.
2. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.
3. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.
4. David Foster, Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play O"Relly, 2nd Edition, 2019.
5. Maxim Lapan, Deep Reinforcement Learning HandsOn: Apply modern RL methods, with deep Q- networks, value iteration, policy gradients, TRPO, AlphaGo Zero and more,O'Reilly, 3rd Edition, 2020
6. TensorFlow 1.x Deep Learning Cookbook, Gulli and Kapoor, Packt, 2017.

Online Resources:

1. NPTEL:

   Deep Learning, By Prof. Prabir Kumar Biswas, IIT Kharagpur

   https://onlinecourses.nptel.ac.in/noc22_cs22/preview

2. Coursera:

   Deep Learning Specialization, By

   DeepLearning.AI

   https://www.coursera.org/specializations/deep-learning#courses

Prepared by | Checked by | Head of the Department | Principal

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

Course: Blockchain and Crypto Technology (DJS23CPC352)

Course: Blockchain and Crypto Technology Laboratory (DJS23CPC352I

Pre-requisite: Computer Networks, Data Structures, Information and Cyber Security

Objectives:

The objective of the course is

1. To introduce Blockchain architecture, distributed ledgers, cryptographic foundations, and consensus mechanisms.
2. To expose learners to public Blockchain systems such as Bitcoin and Ethereum, including smart contracts and Web3 concepts.
3. To develop practical skills for designing and deploying smart contracts, tokens, and decentralized applications.
4. To provide knowledge of crypto systems including token standards, stablecoins, NFTs, DAOs, DeFi protocols, and enterprise Blockchain platforms.
5. To create awareness of security threats and best practices in Blockchain, smart contract development, and wallet security.

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

1. Explore Blockchain architecture, cryptographic foundations, and consensus mechanisms.
2. Analyze public Blockchain systems (Bitcoin, Ethereum), smart contracts, and Web3 applications.
3. Develop basic smart contracts, crypto tokens, and decentralized applications.
4. Evaluate security vulnerabilities in Blockchain systems and compare public and permissioned Blockchain platforms.

Blockchain and Crypto Technology (DJS23CPC352)		
Unit	**Description**	**Duration**
1	**Blockchain & Crypto Fundamentals** Evolution of Web1 → Web2 → Web3. Introduction to Distributed Ledger Technology (DLT), Blockchain architecture: blocks, hash pointers, Merkle trees, Peer-to-peer network and nodes, Cryptographic foundations: Hashing (SHA-256, Keccak-256), Public-key cryptography, ECDSA, Digital signatures, Consensus concepts (overview): PoW, PoS, Crypto basics: Coins vs Tokens, UTXO vs Account-based model, Tokenomics basics: supply models, inflation & deflation	**08**
2	**Public Blockchain Systems & Smart Contracts** **Bitcoin Blockchain:** Motivation for Bitcoin, UTXO model and transaction structure, Mempool, confirmations, and fee mechanism, Mining and difficulty adjustment, Block structure and propagation **Ethereum Blockchain:** Transition from PoW to PoS, Ethereum architecture: EVM, accounts, state, Transactions, gas mechanism, Merkle	**12**

	Patricia Tree – conceptual overview **Smart Contracts & Web3:** Smart contracts: concepts & characteristics, Solidity basics: syntax, functions, events, Contract deployment using Remix, Oracles: types and role. Web3 workflow: Wallet → Contract → Blockchain, Basic DApp architecture and interaction **Layer-2 Scaling (Overview):** Rollups: Optimistic & ZK, Sidechains vs Layer-2	**08**
3	**Crypto Systems & Web3 Applications** **Crypto Ecosystem:** Token standards: ERC-20, ERC-721, ERC-1155, Stablecoins (USDT, USDC, DAI), DAOs: governance models and examples, Decentralized identity (DID) – basics, Real-World Assets (RWAs) **Decentralized Finance (DeFi):** Introduction and motivation, AMMs: Uniswap model (concept), Liquidity pools & impermanent loss (high-level), Lending and borrowing (Aave model – overview), Real-World Assets in DeFi **Web3 Applications:** NFTs: architecture, storage, use cases, Tokenized digital and physical assets, Web3 in gaming, metaverse, and creator economy, Wallets and DApp integration	**07**
4	**Enterprise Blockchain & Interoperability** **Private & Consortium Blockchains:** Permissioned vs Permissionless, Enterprise use cases, Platforms: Hyperledger, Corda, Ripple (overview) **Hyperledger Fabric (Core Concepts):** Components: MSP, peers, orderers, endorsement policies, Channels and privacy, Chaincode (overview), World State and ledger, Transaction flow (conceptual) **Interoperability:** Cross-chain bridges, Polkadot overview (relay chain), Cosmos overview (IBC)	**07**
5	**Blockchain, Crypto Security & Regulation** **Blockchain Security:** Majority attack (51% attack), Sybil attack, Replay and double-spending attacks, MEV (concept only), Wallet security: hot/cold wallets, seed phrase management, Bridge vulnerabilities (high-level) **Smart Contract Security:** Vulnerabilities: Reentrancy, Integer overflow/underflow, Access control issues, Secure coding practices, Overview of security tools: Slither, MythX **Zero Knowledge Proofs (Concept):** Motivation, zk-SNARKs & zk-STARKs (no mathematical depth) **Regulations & Compliance:** Global crypto regulations (overview), Indian regulatory landscape (high-level), Ethical and legal considerations	**07**
	Total	**42**

Blockchain and Crypto Technology Laboratory (DJS23CPC352L)	
Exp.	**Suggested experiments (Any 8)**
1	Implement SHA-256 hashing and develop a simple linked-block simulation.

2	Construct a Merkle tree and verify proof of inclusion.
3	Create Bitcoin/Ethereum testnet wallets; perform and trace transactions.
4	Simulate basic Proof-of-Work mining with difficulty and nonce calculation.
5	Write and deploy a simple smart contract on Remix IDE.
6	Develop and deploy an ERC-20 or ERC-721 token on a testnet.
7	Build a basic Web3 DApp integrated with MetaMask.
8	Analyze vulnerabilities in sample smart contracts using tools.
9	Interact with DeFi testnet protocols (AMM swap simulation).
10	Mini Project integrating blockchain + crypto concepts.

Any other experiment based on syllabus may be included, which would help the learner to understand topic/concept.
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:

Textbooks:

1. S Chandramouli, Asha A George, Abhilash K A, and Meena Kartikeyan, "Blockchain Technology", University Press, 2022.
2. Lorne Lantz, Daniel Cawrey, "Mastering Blockchain", O'Reilly, 2020
3. Andreas M. Antonopoulos, Gavin Wood, "Mastering Ethereum", O'Reilly, 2018
4. Kumar Saurabh, Ashutosh Saxena, "Blockchain Technology: Concepts and Applications", Wiley, 2020

Reference Books:

1. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder, "Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction", Princeton University Press, 2016
2. Dr. Ramchandra Mangrulkar, Dr. Pallavi Mahajan, "Blockchain Essentials: Core Concepts and Implementations", Apress, 2024.
3. Sudeep Tanwar, "Blockchain Technology: from Theory to Practice", Springer, 2022.

Online Resources:

1. NPTEL: Blockchain and its Applications, Prof. Sandip Chakraborty, Prof. Shamik Sural, IIT Kharagpur
2. Coursera: Introduction to Blockchain, James Won-Ki HONG

Prepared by Checked by Head of the Department Principal

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

Course: Software Engineering and DevOPs Laboratory (DJS23CPC355L)

Pre-requisite:
1. Concepts of Object Oriented Programming & Methodology
2. Knowledge of developing applications with front end & back end connectivity.
3. Knowledge of Linux Operating system, installation and configuration of services and command line basics.

Objectives:
The objective of the course is
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.
2. To understand the fundamentals of DevOps engineering.
3. 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. Understand basic concepts of Software Engineering process and models.
2. Identify requirements, analyze and design for real world software projects.
3. Apply various principles, phases and activities of DevOps to meet software development process requirements.
4. Understand and implement DevOps principles for CI/CD.
5. Apply deployment and management tools for agile software projects using DevOps.

Software Engineering and DevOPs Laboratory (DJS23CPC353L)		
Unit	**Description**	**Duration**
1	**Introduction to Software Engineering and Process Models:** Nature of Software, Software Engineering, Software Process, Prescriptive Process Models: The Waterfall Model, V Model; Evolutionary Process Models: The Spiral Model **Agile Methodology:** Agility Principles **Agile Process Models:** Extreme Programming (XP), Scrum, Agile Modeling (AM), Kanban Model, LEAN models **Requirement Analysis and Models:** Software Requirement Specification (SRS), Scenario Based Models, Class Based Models, Behavioral Models, and Flow Models. **AI based software development process tools**: Claude Code, Gradio, OpenUI, etc.	07
2	**Design Engineering and Analysis:** Design Principles, Design Concepts, Effective Modular Design-Cohesion and Coupling **Introduction to DevOps:** Minimum Viable Product (MVP) & Cross-functional Teams, Lean, ITIL,	04

	DevOps as a prominent culture to achieve agility in the software development process, DevOps Stakeholders, Goals, DevOps and Agile, DevOps Tools	
3	**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	04
4	**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	04
5	**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
6	**Continuous Management:** The Parts of an Infrastructure System, Infrastructure Platforms, Infrastructure Resources, Compute Resources, Storage Resources, Network Resources. 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	05
	Total	**28**

Software Engineering and DevOPs Laboratory (DJS23CPC353L)	
Exp.	**Suggested experiments (Any 8)**
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	Perform various GIT operations on local and Remote repositories using GIT Cheat-Sheet.
4	To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job.
5	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.
6	To understand Jenkins Master-Slave Architecture and scale your Jenkins standalone implementation by implementing slave nodes
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 learn Software Configuration Management and provisioning using Puppet Blocks (Manifest, Modules, Classes, Function)

Any other experiment based on syllabus may be included, which would help the learner to understand topic/concept.
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:

Textbooks:

1. Roger Pressman, "Software Engineering: A Practitioner's Approach", McGraw-Hill Publications 8 th Edition, 2015
2. Ali Behfrooz and Fredeick J. Hudson, "Software Engineering Fundamentals", Oxford University Press.
3. Karl Matthias & Sean P. Kane, "Docker: Up and Running", 3rd Edition, O'Reilly Publication, 2022.
4. Craig Berg, "DevOps For Beginners: A Complete Guide To DevOps Best Practices" 2020.
5. Mikael Krief, "Learning DevOps: A comprehensive guide to accelerating DevOps culture adoption with Terraform, Azure DevOps, Kubernetes, and Jenkins", Packt Publication, 2nd Edition, 2022.

Reference Books:

1. Ugrasen Suman, "Software Engineering-Concepts and Practices", Cengage Learning, 2022
2. Pankaj Jalote, "Software Engineering: A Precise Approach", Wiley Publications, 2010
3. Rajib Mall, "Fundamentals of Software Engineering", Prentice Hall India, 4th edition, 2014
4. Mark S. Merkow, "Practical Security for Agile and DevOps", CRC Press Taylor & Francis, 2022.
5. Emily Freeman, "DevOps for Dummies", 3rd Edition, Wiley Publication, 2019.

Online Resources:

1. NPTEL: Software Engineering, By Prof. Rajib Mall, IIT Kharagpur, https://onlinecourses.nptel.ac.in/noc25_cs108/preview
2. Advanced Certification in DevOps & Cloud Computing by Dr. Noor Mohammad and Dr. Pilli Emmanuel Shubhakar, IIT Madras.

Prepared by | Checked by | Head of the Department | Principal

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

Course: Big Data Analytics (DJS23CPE351)

Course: Big Data Analytics Laboratory (DJS23CPE351L)

Pre-requisite: Database Management Systems, Business Intelligence and Analytics

Objectives:

The objective of the course is

1. Provide a comprehensive overview of Big Data analytics, its challenges, and the limitations of traditional data mining approaches.
2. Introduce key Big Data technologies and frameworks—such as Hadoop, NoSQL, and MapReduce—along with scalable and streaming-based analytical techniques.
3. Develop the ability to analyze diverse Big Data types (social media, web graphs, data streams) and apply analytical skills to solve complex real-world decision-support problems.

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

1. Identify the motivation for Big Data systems and major real-world sources of Big Data.
2. Utilize Big Data frameworks such as Hadoop, NoSQL, and MapReduce to store, retrieve, and process large-scale datasets.
3. Apply advanced analytical techniques—including clustering, classification, association mining, and stream/web graph/social media analytics—to derive insights from Big Data.
4. Develop data-driven solutions such as recommendation engines for real-world applications.

Big Data Analytics (DJS23CPE351)		
Unit	**Description**	**Duration**
1	**Introduction to Big Data and Big Data Analytics** Big Data: Definition, Characteristics, Drivers, 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	**08**

	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	**08**
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**: Handling Larger Datasets in Main Memory Basic Algorithm of Park, Chen, and Yu., 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 Neighbour	**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. **Mining Social- Network Graphs**: Social Networks as Graphs, Types , Clustering of Social Network Graphs, Direct Discovery of Communities, **Recommendation Engines**: A Model for Recommendation Systems, Content-Based Recommendations, Collaborative Filtering	**06**
	Total	**42**

Big Data Analytics Laboratory (DJS23CPE351L)	
Exp.	**Suggested experiments (Any 8)**
1	Installation of Hadoop on a single node cluster and execute HDFS commands.

2	Implementation of Map Reduce program to count words in a text file, matrix multiplication
3	Execute HIVE commands to load, insert, retrieve, update, or delete data in the tables.
4	Execute HBASE commands to perform basic CRUD operations and joins.
5	Implement matrix-vector multiply using MapReduce and evaluate correctness and scaling.
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.

Any other experiment based on syllabus may be included, which would help the learner to understand topic/concept.
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:

Textbooks:

1. Seema Acharya, Subhashini Chellappan, "Big Data and Analytics", 2nd Edition, Wiley, 2022.
2. Raj Kamal, Preeti Saxena, "BIG DATA ANALYTICS - Introduction to Hadoop, Spark, and Machine-Learning", McGraw Hill Education, 2019.
3. Alex Holmes, "Hadoop in Practice", 2nd Edition, Manning Publications Co., 2014.

Reference Books:

1. Anand Rajaraman, Jure Leskovec, and Jeffrey D. Ullman, "Mining of Massive Datasets", 3rd Edition, Cambridge University Press, 2014.
2. C.S.R. Prabhu, Aneesh Sreevallabh Chivukula, Aditya Mogadala, Rohit Ghosh, L.M. Jenila Livingston, "Big Data Analytics: Systems, Algorithms, Applications", Springer, 2019.

Online Resources:

1. https://nptel.ac.in/courses/106104189
2. https://www.coursera.org/specializations/big-data#courses

Prepared by | Checked by | Head of the Department | Principal

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

Course: UI/UX Design (DJS23CPE352)

Course: UI/UX Design Laboratory (DJS23CPE352L)

Pre-requisite: Design thinking

Objectives:

The objective of the course is

1. Explore essential visual design & interaction principles applicable to user interfaces.
2. Conduct user research and apply user-centered methodologies for design.
3. Create wireframes, interactive prototypes, and evaluate usability through testing methods.

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

1. Analyze the principles of user interface and user experience.
2. Apply the UI design elements in building user-friendly applications.
3. Analyze user behavior and business requirements.
4. Develop user-centric design solutions using wireframes and prototypes.

UI/UX Design (DJS23CPE352)		
Unit	**Description**	**Duration**
1	**The Importance of UI and Characteristics of Graphical and Web User Interfaces:** Defining the User Interface; The Importance of Good Design; Interaction Styles, Graphical User Interface, Web User Interface; Merging Graphical Business Systems and the Web; Principles of User Interface Design.	06
2	**User Interface Design Process:** Obstacles and Pitfalls in the Development Path; Understanding your User or Client: Understanding How People Interact with Computers, Important Human Characteristics in Design, Human Considerations in the Design of Business Systems, Human Interaction Speeds. Understand the Business Function: Business Definition and Requirements Analysis, Determining Basic Business Functions	09
3	**Principles of Good Interface and Screen Design:** Human Considerations in Interface and Screen Design; Technological Considerations in Interface Design; Structures, Functions, Content and Formatting of Menus; Types of Graphical Menus; Window Characteristics; Components of a window; Types of Windows	08
4	**Interaction Devices and Screen-Based Controls:**	

	Input Devices; Output Devices; Operable Controls; Text Entry/Read-Only Controls; Selection Controls; Combination Entry/Selection Controls; Other Operable Controls; Presentation Controls	07
5	**Effective Text, Feedback, Guidance, Internationalization and Accessibility:** Words, Sentences, Messages, and Text; Content and Text for Web Pages; Providing the Proper Feedback; Guidance and Assistance; International Considerations; Accessibility	06
6	**Creating Meaningful Graphics, Images and Organizing Windows and Pages:** Icons: Types, Characteristics, Icon Design Process; Multimedia and Graphics; Choosing the Proper Colors: Color Uses, Possible Problems with Color, Color and Human Vision; Organizing and Laying Out Screens; Iterative Testing	06
	Total	**42**

UI/UX Design Laboratory (DJS23CPE352L)

Exp.	Suggested experiments (Any 8)
1	Identify and analyze the UI and UX of two digital products.
2	Design a Mobile app/ Website that can teach mathematics to children of 4-5 years age in schools in Rural /Urban Sector
3	ATM machine/KIOSK screen design for rural people
4	Design a UI application for Institute event management.
5	Designing user interfaces for the systems using various interaction styles.
6	Design appropriate icons pertaining to a given domain.
7	Redesign of a user interface (Suggest and implement changes in Existing User Interface)
8	Design an interface for home appliances
9	Design UI for Motor paralysis for disabled people.
10	Mini-project on KIOSK design for hospital/school/educational campus/National Institute

Any other experiment based on the syllabus may be included, which would help the learner understand the topic/concept.
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:

Textbooks:

1. Joel Marsh, "UX for Beginners", O'Reilly. 2022
2. Jon Yablonski, "Laws of UX using Psychology to Design Better Product & Services" O'Reilly 2021

Reference Books:

1. Jenifer Tidwell, Charles Brewer, Aynne Valencia, "Designing Interface" 3rd Edition, O'Reilly 2020
2. Wilbert O. Galitz, "The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques" Wiley Publishing, Inc. 2007

Online Resources:

Google UX Design Professional Certificate
Google UX Design Professional Certificate | Coursera

Prepared by Checked by Head of the Department Principal

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

Course: Virtual Reality (DJS23CPE353)

Course: Virtual Reality Laboratory (DJS23CPE353L)

Pre-requisite: Computer Graphics

Objectives:

The objective of the course is

1. Introduce the principles and technology of Virtual Reality (VR) systems.
2. Familiarize with current VR hardware, software frameworks, and interaction techniques.
3. Develop practical skills in creating immersive VR applications.
4. Explore real-world applications of VR in domains such as gaming, education, healthcare and industrial training.

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

1. Understand the key concepts, architecture, and components of contemporary VR systems.
2. Demonstrate knowledge of modern VR hardware, display/input devices, and software frameworks.
3. Apply 3D modelling, rendering and graphics concepts to design virtual environments.
4. Develop interactive VR applications using modern engines and SDKs.
5. Evaluate VR applications based on performance, usability, comfort, and human factors.
6. Analyze and propose VR applications across different domains (education, training, entertainment).

Virtual Reality (DJS23CPE353)		
Unit	**Description**	**Duration**
1	**Introduction to Virtual Reality:** Definitions & evolution of VR, What is virtual reality? ,The beginnings of VR , VR paradigms ,Collaboration, Virtual reality systems, Representation ,User interaction immersive systems, presence & immersion, VR vs AR vs MR, major application domains.	06
2	**VR Hardware & Software Frameworks:** Head-mounted displays (HMDs), motion tracking, controllers, sensors, input/output devices, VR system architecture, display latency, software frameworks (XR Interaction Toolkit, OpenXR, SteamVR, etc.).	07
3	**3D Graphics & Modelling for VR:** Coordinate systems, transformations, 3D modelling tools, lighting & shading, textures/materials, rendering pipeline, scene optimization for VR	07
4	**Interaction & User Interface in VR:** Navigation techniques (teleportation, locomotion),	07

	selection/manipulation, gesture & hand-tracking, spatial audio & haptics, designing UI for VR, motion sickness and human comfort.	
5	**VR Application Development:** Introduction to development platform (e.g., Unity with XR toolkit), setting up VR projects, camera & rig setup, controller integration, object interactions, scripting (C#), deploying VR applications (PC, standalone).	08
6	**Applying Virtual Reality:** Virtual reality: the medium, Form and genre, What makes an application a good candidate for VR, Promising application fields, Demonstrated benefits of virtual reality , More recent trends in virtual reality application development, A framework for VR application development.	07
	Total	**42**

Virtual Reality Laboratory (DJS23CPE353L)	
Exp.	**Suggested experiments (Any 8)**
1	Study of VR Hardware and Setup – identify components, set up a VR headset and controllers.
2	Configuring a VR Project in Unity + XR Toolkit – create base VR scene, set up VR camera rig.
3	Building & Navigating a Simple 3D Virtual Environment – design scene, load assets, implement navigation.
4	Object Interaction in VR – implement grabbing, throwing, UI interactions using controllers/hand-tracking.
5	Spatial Audio & Lighting Effects – add 3D audio, lighting, optimize for immersion
6	Developing a VR Mini-Game – design and implement a small interactive game in VR (e.g., object hunt, gallery tour).
7	Performance Optimization & Comfort Design – analyze frame-rate, latency, adjust lighting/meshes, implement teleportation vs smooth locomotion, evaluate motion-sickness.
8	Mini Project: Domain-Specific VR Application – students pick a domain (education, training, simulation) and create a short VR experience; show demo and report.

Any other experiment based on syllabus may be included, which would help the learner to understand topic/concept.
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:

Textbooks:

1. Virtual Reality, Steven M. LaValle, Cambridge University Press, 2016
2. Unity 2020 Virtual Reality Projects: Jonathan Linowes, Packt Publishing, 2020.
3. The VR Book: Human-Centered Design for Virtual Reality: Jason Jerald, ACM Books, 2020
4. Understanding Virtual Reality: Interface, Application and Design, William R Sherman andAlan B Craig, (The Morgan Kaufmann Series in Computer Graphics)I. Morgan Kaufmann Publishers, San Francisco, CA, 2002
5. Developing Virtual Reality Applications: Foundations of Effective Design, Alan B Craig,William R Sherman and Jeffrey D Will, Morgan Kaufmann, 2009.

Reference Books:

1. Unity 2022 by Example: A Project-based Guide to Building 2D & 3D Games, Enhanced for AR, VR & MR Experiences: Packt, 2022.
2. Ultimate Unity for Multiplayer VR Development: Focuses on multiplayer VR with Unity, XR Toolkit and Photon Fusion (recent edition).
3. "Augmented Reality and Virtual Reality: New Trends in Immersive Technology" (Springer Cham, 2021) – M. C. tom Dieck et al.
4. "Virtual Reality in Higher Education: Instruction for the Digital Age" (Bloomsbury, 2021) – D. W. Staat.

Online Resources:

1. Introduction to Virtual Reality (by Goldsmiths, University of London on Coursera https://www.coursera.org/specializations/virtual-reality
2. VR Development With XR Interaction Toolkit (on LearnXR.io)
3. Unity VR Tutorial: Unity XR Interaction Toolkit 3.0 (on Udemy) https://www.udemy.com/course/unity-vr-game-development/?couponCode=25BBPMXNVD35V2
4. Free tutorial in Virtual Reality https://www.udemy.com/course/how-to-create-vr-experiences-simlab-soft-vr-studio/?srsltid=AfmBOopN4jR29Pi7m5jzr8HB-UeF04y36V6ZrOTM355z-ibRe6sUyjOg

Prepared by | Checked by | Head of the Department | Principal

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

Course: Computer Vision (DJS23CPE354)

Course: Computer Vision Laboratory (DJS23CPE354L)

Pre-requisite: Computer Graphics and Image Processing

Objectives:

The objective of the course is

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 (DJS23CPE354)		
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 quantization	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, Thickening , 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-	08

	based methods, Application: Medical image segmentation.	
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.	08
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.	08
	Total	**42**

Computer Vision Laboratory (DJS23CPE354L)	
Exp.	**Suggested experiments (Any 8)**
1	To perform basic and geometric image transformations such as rotation, scaling, translation, and affine transformations using computer vision techniques.
2	To apply various image de-noising and enhancement techniques for improving image quality using spatial and frequency-domain methods.
3	To apply image smoothing and sharpening filters for enhancing image details.
4	To perform segmentation of medical images for isolating regions of interest using thresholding, region-based, or deep learning methods
5	To implement feature detection techniques such as edge detection, corner detection, and image pyramids using OpenCV.
6	To detect and localize objects in an image or video using classical or deep-learning-based object detection methods.
7	To track the movement of objects across video frames using suitable object tracking algorithms.
8	To implement pattern recognition techniques for classifying or identifying patterns using machine learning or image-based features.
9	To implement a face recognition system using feature extraction and classification techniques or pre-trained deep learning models.
10	Mini Project Based Learning /Research Review Article

Any other experiment based on the syllabus may be included, which would help the learner understand the topic/concept.
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:

Textbooks:

1. Richard Szeliski, "Computer Vision: Algorithms and Applications", 2nd edition, Springer, 2022.
2. Alexander Hornberg, "Handbook of Machine and Computer Vision", Wiley-VCH, 2017.

Reference Books:

1. Milan Sonka, Vaclav Hlavac, Roger Boyle, "Image Processing, Analysis, and Machine Vision", 4th Edition, Cengage Learning, 2014.
2. David Forsyth, Jean Ponce, "Computer Vision A Modern Approach", 2nd Edition, Pearson, 2013.

Online Resources:

1. https://onlinecourses.nptel.ac.in/noc25_cs143/preview
2. https://www.coursera.org/learn/introduction-computer-vision-watson-opencv

Prepared by | Checked by | Head of the Department | Principal

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

Course: Development Frameworks (DJS23CPE355)

Course: Development Frameworks Laboratory (DJS23CPE355L)

Pre-requisite: Java Programming, Object Oriented Programming Concept, Database Management System.

Objectives:

The objective of the course is

1. To apply design patterns and SOLID principles for creating flexible and maintainable software.
2. To develop database-driven Java web applications using Collections, Servlets, JSP, and JDBC.
3. To build enterprise applications using ORM, Spring Core, Spring Boot, and MVC architecture.
4. To design secure RESTful services and microservices using Spring Boot, JWT/OAuth2, and service discovery tools.

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

1. Apply Object-Oriented Design Patterns and SOLID principles to build scalable and maintainable software solutions.
2. Develop Java applications using Collections, Servlets, JSP, and JDBC for dynamic, database-driven web solutions.
3. Build enterprise-level applications using ORM tools and Spring Framework (Core, Boot) with dependency injection and MVC architecture.
4. Design and implement secure RESTful services and Microservices using Spring Boot, JWT/OAuth2, API Gateway, Load Balancing, and Service Discovery.

Development Frameworks (DJS23CPE355)		
Unit	**Description**	**Duration**
1	**Design Patterns:** Introduction to design patterns and Implementation: Singleton, Factory, Observer, Strategy. **SOLID Principles:** Understanding and applying SOLID principles for better design, Implementation of SOLID Principles.	**03**
2	**Collections:** List: ArrayList, LinkedList, Set: HashSet, Tree Set, Map: HashMap, LinkedHashMap.	**04**
3	**Servlets and JSP:** Introduction, Web application Architecture, Http Protocol & Http Methods, Web Server & Web Container, Servlet Interface, GenericServlet, HttpServlet, Servlet Life Cycle, Servlet Communication, Introduction, JSP LifeCycle, JSP Implicit Objects & Scopes, JSP Directives, JSP Scripting Elements, JSP Actions, connecting to databases using JDBC, Executing SQL queries and managing results.	**08**

	Introduction to Object-Relational Mapping (ORM): Overview of Hibernate and JPA, Creating a simple application using Hibernate	
4	**Introduction to Spring:** Overview of Spring Framework features, Inversion of Control (IoC) and Dependency Injection (DI). (JDBC Must be covered) **Spring Core:** Understanding Beans, Application Context, and Bean Lifecycle Configuring Spring with XML and Java annotations.	**10**
5	**Introduction to Spring Boot:** Introduction to MVC architecture, understanding its purpose and advantages over traditional Spring. Setting Up Spring Boot Applications: Project structure and configuration, Annotations, Layers, Service Class, Repository Layer, Controllers, Model Object, Thymeleaf. **Building RESTful Web Services:** Create REST APIs using Spring Boot. Spring Data JPA: Introduction to database interactions and repository patterns, JWT, OAuth2.	**10**
6	**Microservices:** Fundamentals of Microservices, Microservices Architecture & Design Principles, Service Discovery, Service Registry, Working with Feign, Load Balancing, API Gateway.	**07**
	Total	**42**

Development Frameworks Laboratory (DJS23CPE355L)	
Exp.	**Suggested experiments (Any 8 + Mini Project)**
1	**Implementation of Design Patterns and SOLID Principles.** (Applying Singleton, Factory, Strategy/Observer along with SOLID-based refactoring)
2	**Develop an application to implement Java Collections Framework.** (ArrayList, LinkedList, HashSet, TreeSet, HashMap, LinkedHashMap)
3	**Developing a Basic Servlet Application** (Handling GET/POST requests and servlet lifecycle)
4	**Servlet-to-JSP Communication with MVC Approach** (Forwarding requests, using JSP scopes, and displaying dynamic data)
5	**JSP Application with JDBC Connectivity** (Performing CRUD operations using JSP, JDBC, and MySQL)
6	**Building an ORM-Based Application using Hibernate/JPA** **(Entity mapping and CRUD operations)**
7	**Spring Core Application** (Bean creation, Dependency Injection using XML and annotations)
8	**Spring Boot MVC Application**

	(Controller, Service, Repository layers with Thymeleaf views)
9	**Creating RESTful Web Services using Spring Boot** (CRUD APIs using Spring Data JPA)
10	**Microservices Implementation** (Service Discovery, Feign Client, and API Gateway)
11	**Mini-Project**

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.
The students are expected to complete 1 Mini-Project based on the syllabus

Books Recommended:

Textbooks:

1. Craig Walls, "Spring in Action", 6th Edition, Manning Publications, 2022
2. Craig Walls, "Spring Boot in Action", Manning Publications, 2016
3. Greg L. Turnquist, "Learning Spring Boot 3.0 — Third Edition", O'Reilly Media, 2023
4. Magnus Larsson, "Microservices with Spring Boot and Spring Cloud", O'Reilly Media, 2023

Reference Books:

1. Christian Bauer & Gavin King, "Java Persistence with Hibernate", 2nd Edition, Manning Publications, 2020
2. Ranga Rao Karanam, "Mastering Spring Boot 3.0", 1st Edition, Packt Publishing, 2023
3. ohn Thompson, "The Complete Beginner's Guide to Spring Boot", Independently Published, 2022
4. Mark Johnson, "Learn All About Spring Boot: Building Modern Java Applications", Packt Publishing, 2023
5. Moisés Macero García, "Learn Microservices with Spring Boot", Apress, 2020
6. Sam Newman, "Building Microservices: Designing Fine-Grained Systems", O'Reilly Media, 2015

Online Resources:

1. Zero to Hero: Master Java SpringBoot & JPA with Projects (Link: https://www.coursera.org/learn/packt-zero-to-hero-java-springboot-and-jpa-mastery-with-real-project-1ee3f)
2. Master Spring Boot 3 & Spring Framework 6 with Java (Link: https://www.udemy.com/course/spring-boot-and-spring-framework-tutorial-for-beginners)

Prepared by | Checked by | Head of the Department | Principal

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

Course: High Performance Computing (DJS23CPE356)

Course: High Performance Computing Laboratory (DJS23CPE356L)

Pre-requisite: Computer Architecture, Operating System

Objectives:

1. To understand parallel computing concepts, architectures, and performance metrics.
2. To design and implement efficient parallel programs using OpenMP, MPI, and CUDA.

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

1. To understand fundamentals of parallel computing, classify parallel architectures, and evaluate performance.
2. To design and implement parallel algorithms using decomposition and load balancing.
3. Develop high-performance programs using OpenMP, MPI and CUDA.

High Performance Computing (DJS23CPE356)		
Unit	**Description**	**Duration**
1	Introduction to Parallel Computing: Parallelism, Levels of parallelism(instruction, transaction, task, thread, memory, function) Classification Models: Architectural Schemes(Flynn's, Shore's, Feng's, Handler's) Memory Access: Distributed Memory, Shared Memory, Hybrid Distributed Shared Memory Parallel Architecture: Pipeline Architecture: Arithmetic pipelines, Floating Point, Array Processor	**06**
2	Parallel Programming Platform and Algorithm Design: Parallel Programming Platform: Physical Organization of Parallel Platforms, Communication Costs in Parallel Machines Algorithm Design: Preliminaries, Decomposition Techniques, Characteristics of Tasks and Interactions, Mapping Techniques for Load Balancing, Methods for Containing Interaction Overheads, Parallel Algorithm Models.	**10**
3	Performance Measures: Performance Measures: Speedup, execution time, efficiency, cost, scalability, Effect of granularity on performance, Scalability of Parallel Systems, Amdahl_s Law, Gustavson_s Law, Performance Bottlenecks, The Karp Flatt Metric	**04**
4	HPC Programming: OpenMP and MPI: HPC Programming: Programming Using the Message-Passing Paradigm: Principles of Message Passing Programming, The Building Blocks: Send and Receive Operations MPI: the Message Passing Interface, Topology and Embedding, Overlapping Communication with Computation, Collective Communication and Computation Operations, Introduction to OpenMP	**07**
5	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	**08**

	block indexing, CUDA Memory Management: Memory Hierarchy and Memory Management, Case Studies: simulation, data analytics and machine learning.	
6	Advanced Topics in GPU: Introduction to GPU accelerated libraries, GPU computing frameworks (TensorFlow, PyTorch) and their integration with GPUs, Introduction to GPU clusters and distributed GPU computing, Cluster Setup & its Advantages. Case studies: Real-world applications for GPU computing.	06
	Total	**42**

High Performance Computing Laboratory (DJS23CPE356L)	
Exp.	**Suggested experiments (Any 8)**
1	Perform Linux based commands on remote machine
2	Implement parallel program using OpenMP library and analyze its performance
3	Implement parallel program using MPIplatform and analyze its performance
4	Set up the CUDA environment, install the CUDA Toolkit, and write a basic CUDA program to understand the CUDA development environment.
5	Implement vector addition using CUDA to introduce students to parallelism, thread management, and memory allocation in GPU programming
6	Develop a CUDA program for matrix multiplication to understand parallelism and optimization techniques in GPU computing.
7	Apply CUDA for image processing tasks, like blurring and edge detection, to learn how to process images efficiently using GPU parallelism
8	Implement parallel reduction operations (e.g., sum, min, max) to grasp the concept of efficient parallel reduction.
9	Explore parallel sorting algorithms using CUDA, comparing their performance with CPU based sorting and optimizing CUDA sorting.
10	Employ CUDA to solve real-world problems to understand the power of GPU parallelism.

Any other experiment based on the syllabus may be included, which would help the learner understand the topic/concept.

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:

Textbooks:

1. Ananth Grama et al., Introduction to Parallel Computing, 2nd Edition, Addison-

Wesley, 2023.
2. High Performance Computing for Drug Discovery and Biomedicine, Alexander Heifetz, Springer Nature, ISBN, 1071634496, 9781071634493, 2023.
3. Michael J. Quinn, Parallel Programming in C with MPI and OpenMP, 1st Edition, McGraw-Hill, 2003.
4. Programming in Parallel with CUDA", Richard Ansorge, Cambridge University Press, ISBN-13 978-1108479530, 2022.

Reference Books:

1. Georg Hager and Gerhard Wellein, Introduction to High Performance Computing for Scientists and Engineers, CRC Press, 2019.
2. Duane Storti and Mete Yurtoglu, CUDA for Engineers, 1st Edition, Addison-Wesley, 2016.
3. Kai Hwang and Naresh Jotwani, Advanced Computer Architecture: Parallelism, Scalability, Programmability, 2nd Edition, McGraw-Hill, 2008.

Online Resources:

1. **https://nptel.ac.in/courses/106108055**

Prepared by Checked by Head of the Department Principal

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

Course: Ethical Hacking and Digital Forensics (DJS23CMD351)

Course: Ethical Hacking and Digital Forensics Laboratory (DJS23CMD3

Pre-requisite:

Objectives:
The objective of the course is

1. Introduce ethical hacking concepts, phases of cyber-attacks, and legal and ethical responsibilities of cybersecurity professionals.
2. Develop the ability to perform reconnaissance, vulnerability assessment, exploitation, and penetration testing using standard tools and techniques.
3. Provide knowledge of digital forensics methodologies including evidence collection, preservation, analysis, and reporting as per legal standards.
4. Enable learners to analyze security breaches, perform system/network investigations, and apply forensic tools for cybercrime detection and incident handling.

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

1. Explain the phases of ethical hacking, legal frameworks, and apply professional ethics during security assessments.
2. Perform footprinting, scanning, vulnerability assessment, and simulate system, web, and wireless attacks in a controlled environment.
3. Apply digital forensic procedures for acquiring, preserving, examining, and analyzing digital evidence from storage, memory, mobile, or network sources.
4. Prepare technical forensic or penetration testing reports, follow chain of custody, and recommend preventive and corrective security measures.

Ethical Hacking and Digital Forensics (DJS23CMD351)		
Unit	**Description**	**Duration**
1	**Introduction to Ethical Hacking & Cyber Laws:** Ethical hacking vs. malicious hacking, Hacker types: White hat, black hat, grey hat, hacktivists, Hacking phases: Reconnaissance, Scanning, Gaining Access, Maintaining Access, Covering Tracks, Footprinting – Passive & Active information gathering, Social engineering (Phishing, Pretexting, Vishing, Shoulder Surfing), Cyber laws in India: IT Act 2000 & Amendments, Evidence Act relevance, Legal & ethical responsibilities of an ethical hacker	**04**
2	**Scanning, Enumeration & Vulnerability Assessment** Network scanning: ICMP scanning, TCP connect, SYN scan, NULL, FIN, Xmas scans, Tools: Nmap, Zenmap, Angry IP Scanner, Enumeration: SNMP, LDAP, FTP, SMTP, NetBIOS, Banner grabbing, OS fingerprinting, Vulnerability assessment vs. Penetration testing, Tools: Nessus, OpenVAS, Nikto	**04**

3	**System & Network Hacking Techniques** Password cracking: Brute force, dictionary, rainbow tables, John the Ripper, Hashcat, Privilege escalation: Windows & Linux techniques, Keyloggers, spyware, Trojans (conceptual – no overlap with malware internals studied earlier), Hacking Windows systems: SAM file extraction, Mimikatz basics, Hacking Linux systems: /etc/shadow, SSH attacks, Sniffing & MITM attacks (Ettercap, Bettercap, ARP poisoning – practical focus), Session hijacking fundamentals	**06**
4	**Web Application & Wireless Hacking** Web application penetration testing methodology. Burp Suite basics: Intercepting, scanning, exploiting, Command Injection, Directory Traversal, File Upload attacks, Session hijacking & Cookie stealing, Wireless hacking: WEP cracking, WPA/WPA2 handshake capture & cracking (aircrack-ng), Evil Twin & Rogue Access Point attacks	**05**
5	**Digital Forensics & Incident Handling** Digital forensics process: Identification, Preservation, Collection, Examination, Analysis, Reporting, Forensic readiness & chain of custody, Disk forensics: Imaging (dd, FTK Imager), Hashing (MD5/SHA-1 for integrity), Autopsy/Sleuth Kit usage, Volatile data acquisition: RAM dump (FTK Imager Lite, DumpIt), live response tools, File systems: FAT, NTFS artifacts (deleted files, metadata, MFT), Timestamp analysis, log analysis	**05**
6	**Network, Email, Mobile & Cloud Forensics** Network forensics: Packet reconstruction, PCAP analysis, Wireshark forensic techniques, Email forensics: Header analysis, tracing emails, spoofed email detection, Cloud forensics fundamentals: Data acquisition challenges, legal issues, hypervisor forensics, Report writing & forensic documentation standards	**04**
	Total	**28**

Ethical Hacking and Digital Forensics Laboratory (DJS23CMD351L)	
Exp.	**Suggested experiments (Any 8)**
1	Perform passive OSINT reconnaissance on a mock target and prepare an evidence log suitable for a forensic timeline.
2	Use DNS/WHOIS and subdomain enumeration tools to map a target's public footprint; capture and store outputs with hashes.
3	Run safe vulnerability scans (in an isolated lab) using an industry scanner; validate two findings manually and document false positives.
4	Demonstrate safe, controlled exploitation of a deliberately vulnerable VM (e.g., privilege escalation on Linux or Windows) and record the post-exploit steps (no real targets).
5	Create and verify a forensic image of a disk using a forensic imaging tool; compute hashes and demonstrate evidence integrity verification.

6	Recover deleted files from a disk image and reconstruct a simple timeline of user activity.
7	Capture and analyze a memory image from a test VM; identify running processes, network connections, and suspicious loaded modules.
8	Simulate a basic SQL injection on a lab web application, extract controlled test data, and then apply a mitigation (parameterized queries) to demonstrate fix.
9	Demonstrate a stored/reflective XSS attack on an isolated test page and then implement Content Security Policy (CSP) mitigation.
10	Perform captive-portal/WPA2 test in a controlled wireless lab (assess AP configuration weaknesses) and recommend hardening steps.
11	Perform static malware analysis on a safe sample in an isolated sandbox (examine strings, headers, basic indicators) and summarize findings.
12	Conduct a basic dynamic malware behavior run in a sandbox environment (network, file, registry changes) and capture indicators of compromise.
13	Acquire a logical image from an Android or iOS test device (emulator or spare device), extract call/SMS/app artifacts, and document acquisition steps and limitations.
14	Produce a professional forensic/penetration test report from one of the above labs: include scope, methodology, evidence (hashed files), findings, severity, remediation, and an executive summary suitable for stakeholders.

Any other experiment based on syllabus may be included, which would help the learner to understand topic/concept.

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:
Textbooks:

1. **Matt Walker**, *CEH Certified Ethical Hacker All-in-One Exam Guide*, 6th Edition, McGraw Hill, 2023.
2. **William Stallings**, *Computer Security: Principles and Practice*, 5th Edition, Pearson, 2023.
3. **Nelson, Phillips & Steuart**, *Guide to Computer Forensics and Investigations*, 6th Edition, Cengage, 2023.

Reference Books:

1. **Dafydd Stuttard & Marcus Pinto**, *The Web Application Hacker's Handbook*, 2nd Edition, Wiley, 2023 (latest available).
2. **Jason Luttgens, Matthew Pepe, Kevin Mandia**, *Incident Response & Computer Forensics*, 3rd Edition, McGraw Hill, 2022.
3. **Michael Sikorski & Andrew Honig**, *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*, No Starch Press, Reprint 2023.

Online Resources:

1. **MITRE ATT&CK Framework** – Real-world hacking tactics & techniques
   https://attack.mitre.org
2. **OWASP (Open Web Application Security Project)** – Web & API security, Top 10 vulnerabilities, labs https://owasp.org
3. **DFIR Training & Forensics Tools (SANS | Autopsy | Volatility Foundation)**
   https://digital-forensics.sans.org
   https://www.autopsy.com
   https://www.volatilityfoundation.org

Prepared by | Checked by | Head of the Department | Principal

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

Course: Innovative Product Development IV (DJS23XSC351P)

Pre-requisite:

Objectives:

The objective of the course is

1. To acquaint the students with the process of identifying the need (considering a societal requirement) and ensuring that a solution is found 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 conceptualize and create a successful product.

Outcomes: On completion of the course, the 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 conceptualizing 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 analyze 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: o Marks awarded by the supervisor based on log-book: 10 o Marks awarded by review committee: 10 o 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 finalization of the product selected.
  - Second shall be on finalization 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.

Prepared by | Checked by | Head of the Department | Principal

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

Course: Environmental Studies (DJS23XHS361T)

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

Objectives:

The objective of the course is

1. Familiarize 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, the learner will be able to:

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

Environmental Studies (DJS23XHS361T)		
Unit	**Description**	**Duration**
1	**Air Pollution** Sources of Air pollution. Definition of Air Quality Index and how it is measured.	2
2	**Water Pollution** Sources of water pollution. Ground water pollution and eutrophication.	2
3	**Noise Pollution** Noise pollution and sources. Decibel limits for hospital, library, silence zone.	1
4	**Biodiversity loss** Value of Biodiversity. Endangered species.	1
5	**Deforestation** Product and services provided by forests. Relationship between forests and climate change.	2
6	**Renewable Energy sources** Our energy needs and global energy crisis. Renewable energy sources.	2
7	**Climate change** Greenhouse gases and climate change.	2
8	**Green Technology** Data Center Energy Efficiency, Thin-Client and Energy Efficiency.	2
	Total	**14**

List of Tutorials

Sr. No.	Tutorial List
1	Case study on Smog.
2	Qualitative and Quantitative methods for Air Pollution Monitoring
3	Presentation on Water Pollution (Industrial, Sewage) explaining any specific case.
4	General Techniques in Water Quality Monitoring
5	List effects of noise pollution on human health. Measure decibel level in college library, canteen, classroom
6	Case study on effect of Pollution on Biodiversity loss.
7	Radioactive and Hazardous Pollutants.
8	Debate for and against "To promote Economic growth, Deforestation is required."
9	Presentation on different Renewable Energy Technologies.
10	Green Hydrogen: The Energy of Future
11	Report on major impact of Global warming on Environment giving real examples.
12	Report on advantages and examples of Green Building for Sustainable development,
13	Sustainable Software Design: Reducing energy/carbon footprint (e.g., green coding, optimized AI/ML, efficient cloud usage)
14	Reducing AI's carbon footprint (efficient models, serverless) and applying AI for environmental good, like optimizing energy grids, smart buildings, precision agriculture (water/fertilizer use), waste management, and climate modeling.

Tutorial: (Term work: 25 marks)

- Minimum of 10 Tutorials required.
- The distribution of marks for term work shall be as follows:

i. Performance in Tutorial: 15 Marks
ii. Write-up/ Report: 10 marks
The final certification and acceptance of term work will be subject to satisfactory work performance and fulfilling minimum passing criteria in the term work.

Books Recommended:
Textbooks:

1. R. Rajagopalan, Environmental Studies: From Crisis to Cure, Oxford University Press, 2011.
2. Erach Bharucha, Textbook of Environmental Studies For Undergraduate Courses,

Universities Press (India) Pvt. Limited, 2005.
3. P. Narayanan, Environmental Pollution: Principles, Analysis and Control, CBS Publishers, 2018.
4. Mohammad Dastbaz, Colin Pattinson, Babak Akhgar, Morgan and Kaufman, Green Information Technology A Sustainable Approach, Elsevier, 2015.

Reference Books:

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

Websites:

CITES: www.cites.org
Convention on Biological Diversity: www.biodiv.org
Kalpvriksh: www.kalpvriksh.org
Water pollution: http://en.wikipedia.org/wiki/Water_polution
Ecosan: www.eco-solutions.org

Prepared by | Checked by | Head of the Department | Principal