VIDYA JYOTHI INSTITUTE OF TECHNOLOGY

(An Autonomous Institution)

Aziz Nagar Gate, C.B. Post, Hyderabad - 500 075, Telangana.

COURSE SYLLABI

B.Tech II Year

R-25

For

B. Tech Computer Science and Engineering

(Artificial Intelligence & Machine Learning)

B. TECH SECOND YEAR COURSE STRUCTURE

B. Tech IIYear I Semester

S.No.	Course Category	Course Code	Course Title	L	T	P	Credits
1.	BS	A253001	Probability & Statistics	3	0	0	3
2.	PC	A253501	Computer Organization and Architecture	3	0	0	3
3.	PC	A253502	Object Oriented Programming through Java	3	0	0	3
4.	PC	A253503	Operating Systems	3	0	0	3
5.	PC	A253504	Database Management System	3	0	0	3
6.	BS	A253081	Computational Mathematics Lab	0	0	2	1
7.	PC LAB	A253582	Object Oriented Programming through Java Lab	0	0	2	1
8.	PC LAB	A253583	Operating Systems Lab	0	0	2	1
9.	PC LAB	A253584	Database Management Systems Lab	0	0	2	1
10.	SD LAB	A253585	Django	0	0	2	1
			Total Credits	15	0	10	20

B. Tech II Year II Semester

S.No.	Course Category	Course Code	Course Title	L	T	P	Credits
1.	PC	A254004	Discrete Mathematics	3	0	0	3
2.	PC	A254501	Full StackDevelopment	3	0	0	3
3.	PC	A254502	Algorithms Design and Analysis	3	0	0	3
4.	PC	A254503	Computer Networks	3	0	0	3
5.	PC	A254504	Machine Learning	3	0	0	3
6.	HS	A254005	Innovation and Entrepreneurship	2	0	0	2
7.	PC LAB	A254582	Full Stack Development Lab	0	0	2	1
8.	PC LAB	A254583	Computer Networks Lab	0	0	2	1
9.	PC LAB	A254584	Machine Learning Lab	0	0	2	1
10.	SC LAB	A254585	Data Visualization – Power BI	0	0	2	1
11.	HS	A254001	Indian Knowledge System	1	0	0	1
			Total Credits	18	0	08	22

COMPUTER ORGANIZATION AND ARCHITECTURE

B.Tech II Year I Semester
Course Code:

L	T	P	C
3	0	0	3

Course Outcomes:

At the end of the course, student will be able to:

1. Understand the basics of instruction sets and their impact on processor design.
2. Demonstrate an understanding of the design of the functional units of a digital computer system.
3. Evaluate cost performance and design trade-offs in designing and constructing a computer processor including memory.
4. Design a pipeline for consistent execution of instructions with minimum hazards.
5. Recognize and manipulate representations of numbers stored in digital computers

UNIT-I

Boolean Algebra and Logic Gates: Binary codes, Binary Storage and Registers, Binary logic, Digital logic gates.

Data Representation: Data types, Complements, Fixed Point Representation, Floating Point Representation

Digital Computers: Introduction, Block diagram of Digital Computer, Definition of Computer Organization, Computer Design and Computer Architecture.

UNIT-II

Combinational Logic: Combinational Circuits, Analysis procedure Design procedure, Binary Adder Subtractor Decimal Adder, Binary multiplier, magnitude comparator, Decoders, Encoders, Multiplexers, HDL for combinational circuits.

Sequential Logic: Sequential circuits, latches, Flip-Flops Analysis of clocked sequential circuits, state Reduction and Assignment, Design Procedure. Registers, shift Registers, Ripple counters, synchronous counters, other counters.

UNIT-III

Register Transfer Language and Micro operations: Register Transfer language, Register Transfer, Bus and memory transfers, Arithmetic Micro operations, logic micro operations, shift micro operations, Arithmetic logic shift unit.

Basic Computer Organization and Design: Instruction codes, Computer Registers Computer instructions, Timing and Control, Instruction cycle, Memory Reference Instructions, Input – Output and Interrupt.

UNIT-IV

Microprogrammed Control: Control memory, Address sequencing, micro program example, design of control unit.

Central Processing Unit: General Register Organization, Instruction Formats, Addressing modes, Data Transfer and Manipulation, Program Control.

Computer Arithmetic: Addition and subtraction, multiplication Algorithms, Division Algorithms, Floating – point Arithmetic operations. Decimal Arithmetic unit, Decimal Arithmetic operations.

UNIT-V

Input-Output Organization: Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupt Direct memory Access.

Memory Organization: Memory Hierarchy, Main Memory, Auxiliary memory, Associate Memory, Cache Memory.

TEXT BOOKS:

1. Digital Design – M. Morris Mano, Third Edition, Pearson/PHI.
2. Computer System Architecture – M. Morris Mano, Third Edition, Pearson/PHI.

REFERENCE BOOKS:

1. Switching and Finite Automata Theory, ZVI. Kohavi, Tata Mc Graw Hill.
2. Computer Organization – Carl Hamacher, ZvonksVranesic, SafeaZaky, 5th Edition, McGraw Hill.
3. Computer Organization and Architecture – William Stallings Sixth Edition, Pearson/PHI.
4. Structured Computer Organization – Andrew S. Tanenbaum, 4th Edition, PHI/Pearson.

OBJECT ORIENTED PROGRAMMING THROUGH JAVA

B.Tech II Year I Semester

Course Code:

L	T	P	C
3	0	0	3

Course Outcomes:

At the end of the course, student will be able to:

1. Demonstrate the behavior of programs involving the basic programming constructs like control structures, constructors, string handling and garbage collection.
2. Demonstrate the implementation of inheritance (multilevel, hierarchical and multiple) by using extend and implement keywords
3. Use multithreading concepts to develop inter process communication.
4. Understand the different types of file I/O,Exploring String,usage of Container classes
5. Develop web applications using Servlets and JDBC.

UNIT-I

Object oriented thinking and Java Basics- OOP concepts, History of Java, Java buzzwords, data types, variables, operators, expressions, control statements, arrays, type casting, simple java program, concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, parameter passing, object Class and wrapper classes.

UNIT-II

Inner classes in Java: Types of inner classes, Creating static / non-static inner classes, Local and anonymous inner classes.

Inheritance, Interfaces and Packages– Types of Inheritance, benefits of inheritance, super, final keyword with inheritance, method overriding, abstract classes. Defining and implementing interfaces.

Packages: Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages.

UNIT-III

Exception handling and Multithreading-- Concepts of exception handling, benefits of exception handling, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, User Defined Exceptions. Differences between multithreading and multitasking, thread life cycle, creating threads, thread priorities, synchronizing threads, inter thread communication, thread groups, daemon threads.

UNIT-IV

Exploring String, StringBuffer, StringBuilder classes and StringTokenizer.

Java.io package: File, Byte Streams: InputStream, OutputStreams, FileInputStream, FileOutputStream, Character Streams: Reader, Writer, FileReader, FileWriter, BufferedReader, BufferedWriter, InputStreamReader.

java.util package- Collection Interfaces: List, Set, Map. Collection classes: LinkedList, TreeSet, HashMap, Enumerator, Iterator, ListIterator, Calendar, Random, Scanner, Comparator, Comparable.

UNIT-V

Java Servlets: Overview of Java Servlet, Servlet Life cycle, Request and Response methods, Servlet Configuration, Servlet Context, Approaches to Session tracking, Servlet Collaboration.

Database Connections: Introduction to JDBC, JDBC Drivers, Connecting to a Database using JDBC.

TEXT BOOKS:

1. Java the complete reference, 13th edition, Herbert schildt, Dr. Denny Coward, Mc Graw Hill.
2. Understanding OOP with Java, updated edition, T. Budd, Pearson education.

REFERENCE BOOKS:

1. An Introduction to programming and OO design using Java, J.Nino and F.A. Hosch, John Wiley & sons.
2. An Introduction to OOP, third edition, T. Budd, Pearson education.
3. Introduction to Java programming, Y. Daniel Liang, Pearson education.
4. An introduction to Java programming and object-oriented application development, R.A. Johnson- Thomson.
5. Core Java 2, Vol 1, Fundamentals, Cay.S. Horstmann and Gary Cornell, eighth Edition, Pearson Education.

OPERATING SYSTEMS

B.Tech II Year I Semester
Course Code:

L	T	P	C
3	0	0	3

Course Outcomes:

At the end of the course, student will be able to:

1. Will be able to control access to a computer and the files that may be shared
2. Demonstrate the knowledge of the components of computers and their respective roles in computing.
3. Ability to recognize and resolve user problems with standard operating environments.
4. Gain practical knowledge of how programming languages, operating systems, and architectures interact and how to use each effectively.

UNIT-I

Operating System: Introduction, Structures - Simple Batch, Multiprogrammed, Time-shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, System components, Operating System services, System Calls

Process: Process concepts and scheduling, Operations on processes, Cooperating Processes, Threads

UNIT-II

CPU Scheduling: Scheduling Criteria, Scheduling Algorithms, Multiple -Processor Scheduling. System call interface for process management-fork, exit, wait, waitpid, exec

Deadlocks: System Model, Deadlocks Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock

UNIT-III

Process Management and Synchronization: The Critical Section Problem, Synchronization Hardware, Semaphores, and Classical Problems of Synchronization, Critical Regions, Monitors

Interprocess Communication Mechanisms: IPC between processes on a single computer system, IPC between processes on different systems, using pipes, FIFOs, message queues, shared memory.

UNIT-IV

Memory Management and Virtual Memory: Logical versus Physical Address Space, Swapping, Contiguous Allocation, Paging, Segmentation, Segmentation with Paging, Demand Paging, Page Replacement, Page Replacement Algorithms.

UNIT-V

File System Interface and Operations: Access methods, Directory Structure, Protection, File System Structure, Allocation methods, Free-space Management. Usage of open, create, read, write, close, lseek, stat, ioctl system calls.

TEXT BOOKS:

1. Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7th Edition, John Wiley
2. Advanced programming in the UNIX environment, W.R. Stevens, Pearson education.

REFERENCE BOOKS:

1. Operating Systems- Internals and Design Principles, William Stallings, Fifth Edition–2005, Pearson Education/PHI
2. Operating System A Design Approach- Crowley, TMH.
3. Modern Operating Systems, Andrew S. Tanenbaum 2nd edition, Pearson/PHI
4. UNIX programming environment, Kernighan and Pike, PHI/ Pearson Education
5. UNIX Internals -The New Frontiers, U. Vahalia, Pearson Education.

DATABASE MANAGEMENT SYSTEMS

B.Tech II Year I Semester
Course Code:

L	T	P	C
3	0	0	3

Course Outcomes:

At the end of the course, student will be able to:

1. Gain knowledge of fundamentals of DBMS, database design and normal forms
2. Master the basics of SQL for retrieval and management of data.
3. Be acquainted with the basics of transaction processing and concurrency control.
4. Familiarity with database storage structures and access techniques

UNIT-I

Database System Applications: A Historical Perspective, File Systems versus a DBMS, the Data Model, Levels of Abstraction in a DBMS, Data Independence, Structure of a DBMS

Introduction to Database Design: Database Design and ER Diagrams, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design With the ER Model

UNIT-II

Introduction to the Relational Model: Integrity constraint over relations, enforcing integrity constraints, querying relational data, logical database design, introduction to views, destroying/altering tables and views.

Relational Algebra, Tuple relational Calculus, Domain relational calculus.

UNIT-III

SQL: QUERIES, CONSTRAINTS, TRIGGERS: form of basic SQL query, UNION, INTERSECT, and EXCEPT, Nested Queries, aggregation operators, NULL values, complex integrity constraints in SQL, triggers and active databases.

Schema Refinement: Problems caused by redundancy, decompositions, problems related to decomposition, reasoning about functional dependencies, FIRST, SECOND, THIRD normal forms, BCNF, lossless join decomposition, multivalued dependencies, FOURTH normal form, FIFTH normal form.

UNIT-IV

Transaction Concept, Transaction State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for serializability, Lock Based Protocols, Timestamp Based Protocols, Validation- Based Protocols.

UNIT-V

Hashing: Hash Based Indexing, Tree based Indexing, Comparison of File Organizations, Indexes- Intuitions for tree Indexes, Indexed Sequential Access Methods (ISAM)

B+ Trees: A Dynamic Index Structure.

TEXT BOOKS:

1. Database System Concepts, Silberschatz, Korth, McGraw hill, V edition.3rd Edition
2. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Tata Mc Graw Hill

REFERENCE BOOKS:

1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition.
2. Fundamentals of Database Systems, ElmasriNavrate, Pearson Education
3. Introduction to Database Systems, C. J. Date, Pearson Education
4. Oracle for Professionals, The X Team, S.Shah and V. Shah, SPD.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL, Shah, PHI.
6. Fundamentals of Database Management Systems, M. L. Gillenson, Wiley Student Edition.

OBJECT ORIENTED PROGRAMMING THROUGHJAVA LAB

B.Tech II Year I Semester
Course Code:A253582

L	T	P	C
0	0	2	1

Course Outcomes:

At the end of the course, student will be able to:

1. Able to write programs for solving real world problems
2. Able to write programs using abstract classes.
3. Able to write multithreaded programs.
4. Able to write the programs using the java collection framework.
5. Able to create Web applications using servlets and JDBC.

List of Experiments:

1. Write a Java program that prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$. Read in a, b, c and use the quadratic formula. If the discriminate b2 -4ac is negative, display a message stating that there are no real solutions.
2. Write a Java program that prompts the user for an integer and then prints out all prime numbers up to that integer.
3. Write a Java program that uses both recursive and non recursive functions to print the nth value in the Fibonacci sequence.
4. Write a Java program to create an Employee class to read and display the data (emp_id,emp_name,Department,salary,experience) using constructor and method.
5. Write a Java program to illustrate method and constructor overloading.
6. Write a Java program to multiply two given matrices by passing objects as parameters.
7. Write a Java program to illustrate different types of inheritances.
8. Write a java program to illustrate Methodoverriding.
9. Write a Java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.
10. Write a Java program to illustrate the use of creating and importing packages.
11. Write a Java program to handle multiple exceptions.
12. Write a Java program to create User defined exceptions.
13. Write a Java program that implements a multi-thread application that has three threads. The first thread generates a random integer every 1 second and if the value is even, the second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of the cube of the number.

14. Write a Java program that implements the producer – consumer problem using the concept of inter thread communication.
15. Write a Java program to list all the files in a directory including the files present in all its subdirectories.
16. Write a Java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record is separated by a tab (\t). It takes a name or phone number as input and prints the corresponding other value from the hash table (hint: use hash tables).
17. Write a Java Program that reads a line of integers, and then displays each integer, and the sum of all the integers
18. Write a Java program to make frequency count of words in a given text
19. Write a java program to establish Database connection and execute queries using JDBC.
20. Installation and configuration of Tomcat and deploy a simple "Hello World" servlet.

TEXTBOOKS:

1. Java for Programmers, P. J. Deitel and H. M. Deitel, 10th Edition Pearson education.
2. Thinking in Java, Bruce Eckel, Pearson Education.

REFERENCE BOOKS:

1. Java Programming, D. S. Malik and P. S. Nair, Cengage Learning.
2. Core Java, Volume 1, 9th edition, Cay S. Horstmann and G Cornell, Pearson.

OPERATING SYSTEMS LAB

B.Tech II Year I Semester
Course Code: A253583

L	T	P	C
0	0	2	1

Course Outcomes:

At the end of the course, student will be able to:

1. Simulate and implement operating system concepts such as scheduling, deadlock management, file management and memory management.
2. Able to implement C programs using Unix system calls.

LIST OF EXPERIMENTS:

1. Write C programs to simulate the following CPU Scheduling algorithms
   a) FCFS b) SJF c) Round Robin d) priority
2. Write programs using the I/O system calls of UNIX/LINUX operating system (open, read, write, close, lseek, stat, fork, exit)
3. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance.
4. Write a C program to implement the Producer – Consumer problem using semaphores using UNIX/LINUX system calls.
5. Write C programs to illustrate the following IPC mechanisms
   a) Pipes b) FIFOs c) Message Queues d) Shared Memory
6. Write C programs to simulate the following memory management techniques
   a) Paging b) Segmentation
7. Write C programs to simulate Page replacement policies
   a) FCFS b) LRU c) Optimal

TEXT BOOKS:

1. Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7th Edition, John Wiley.
2. Advanced programming in the Unix environment, W. R. Stevens, Pearson education.

REFERENCE BOOKS:

1. Operating Systems – Internals and Design Principles, William Stallings, Fifth Edition–2005, Pearson Education/PHI.
2. Operating System - A Design Approach-Crowley, TMH.
3. Modern Operating Systems, Andrew S Tanenbaum, 2nd edition, Pearson/PHI.
4. UNIX Programming Environment, Kernighan and Pike, PHI/Pearson Education.
5. UNIX Internals: The New Frontiers, U. Vahalia, Pearson Education.

DATABASE MANAGEMENT SYSTEMS LAB

B.Tech II Year I Semester
Course Code: A253584

L	T	P	C
0	0	2	1

Course Outcomes:

At the end of the course, student will be able to:

1. Design database schema for a given application and apply normalization
2. Acquire skills in using SQL commands for data definition and data manipulation
3. Develop solutions for database applications using procedures, cursors and triggers

List of Experiments:

1. Concept design with E-R Model
2. Relational Model
3. Normalization
4. Practicing DDL commands
5. Practicing DML commands
6. A) Querying (using ANY, ALL, UNION, INTERSECT, JOIN, Constraints etc.)
   B) Nested, Correlated subqueries
7. Queries using Aggregate functions, GROUP BY, HAVING and Creation and dropping of Views.
8. Triggers (Creation of insert trigger, delete trigger, update trigger)
9. Procedures
10. Usage of Cursors

TEXTBOOKS:

1. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Tata Mc Graw Hill,3rd Edition
2. Database System Concepts, Silberschatz, Korth, McGraw Hill, V edition.

REFERENCE BOOKS:

1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition.
2. Fundamentals of Database Systems, ElmasriNavrate, Pearson Education
3. Introduction to Database Systems, C.J. Date, Pearson Education
4. Oracle for Professionals, The X Team, S. Shah and V. Shah, SPD.
5. Database Systems Using Oracle: A Simplified guide to SQL and PL/SQL, Shah, PHI.
6. Fundamentals of Database Management Systems, M. L. Gillenson, Wiley Student Edition.

DJANGO LAB

L	T	P	C
0	0	2	1

B.Tech II Year I Semester

Course Code: A253585

Course Outcomes:

At the end of the course, student will be able to:

1. Design and Develop a responsive Web Application using HTML & CSS.
2. Build a responsive Web Application using Java Script, CSS, Flex & Grid.
3. Demonstrate understanding of the Django Framework
4. Implement Mailing, Cache and Session Management in Django Framework
5. Analyze, Design, Develop and Implement complete Web Applications using Django Framework

LIST OF EXPERIMENTS:

1. Build a Responsive Web Application for Registration Form, which contains User Name, Password, Date of Birth, Gender, Mail-id, Contact Number, Address and Submit Button.
2. Write a JavaScript Program to Validate Registration Page using Regular Expression.
3. Build a Responsive Web Application for Shopping Cart with Registration, Login, Catalog and Cart Pages using CSS3 features, Flex and Grid.
4. Basics of Django Framework & Installation of required Software's.
5. Create a shopping cart web application and add a customized admin page for product catalog management for Experiment 3
6. Demonstrate Cache, Session Management for Student Login Application.
7. Create a Quiz Application using Django Framework along with SQLite3.
8. Create a Django application using generic views and forms
9. Create an Application for Working with Mails using Django Framework

10. Create an application that fetches the weather information from openweathermap.org and the display the current and historical weather information using graphical representation using chart.js implement using Django
11. **Case Study:** Django Application for VJIT Student Portal and deploy it into GitHub.

REFERENCE BOOKS:

1. "Web Technologies: HTML, CSS, JavaScript" *Uttam K. Roy*
2. "Django for Beginners: Build Websites with Python and Django" *William S. Vincent*
3. "Python Web Development with Django" *Jeff Forcier, Paul Bissex, Wesley Chun*

Web URLs:

1. W3Schools Web Development Tutorials
   https://www.w3schools.com
2. JavaScript Regular Expressions Guide – MDN
   https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions
3. Python Official Documentation
   https://www.python.org/doc/
4. Python Tutorial (W3Schools)
   https://www.w3schools.com/python/
5. Django Official Documentation
   https://docs.djangoproject.com
6. Django Tutorials
   https://djangoproject.com/start/

DISCRETE MATHEMATICS

B.Tech II Year II Semester
Course Code:

L	T	P	C
3	0	0	3

Course Outcomes:

At the end of the course, student will be able to:

1. Understand and construct precise mathematical proofs.
2. Apply logic and set theory to formulate precise statements.
3. Analyze and solve counting problems on finite and discrete structures.
4. Describe and manipulate sequences.
5. Apply graph theory in solving computing problems

UNIT-I

Mathematical logic: Introduction, Statements and Notation, Connectives, Normal Forms, Theory of Inference for the Statement Calculus, The Predicate Calculus, Inference Theory of the Predicate Calculus.

UNIT-II

Set theory: Introduction, Basic Concepts of Set Theory, Representation of Discrete Structures, Relations and Ordering, Functions.

UNIT-III

Algebraic Structures: Introduction, Algebraic Systems, Semi groups and Monoids, Lattices as Partially Ordered Sets, Boolean Algebra.

UNIT-IV

Elementary Combinatorics: Basics of Counting, Combinations and Permutations, Enumeration of Combinations and Permutations, Enumerating Combinations and Permutations with Repetitions, Enumerating Permutation with Constrained Repetitions, Binomial Coefficient, The Binomial and Multinomial Theorems, The Principle of Exclusion.

UNIT-V

Graph Theory: Basic Concepts, Isomorphism and Subgraphs, Trees and their Properties, Spanning Trees, Directed Trees, Binary Trees, Planar Graphs, Euler's Formula, Multi-graphs and Euler Circuits, Hamiltonian Graphs, Chromatic Numbers, The Four-Color Problem.

TEXT BOOKS:

1. Discrete Mathematical Structures with Applications to Computer Science: J.P. Tremblay, R. Manohar, McGraw-Hill, 1st ed.
2. Discrete Mathematics for Computer Scientists & Mathematicians: Joe l. Mott, Abraham Kandel, Teodore P. Baker, Prentis Hall of India, 2nd ed.

REFERENCE BOOKS:

1. Discrete and Combinatorial Mathematics - an applied introduction: Ralph. P. Grimald, Pearson education, 5th edition.
2. Discrete Mathematical Structures: Thomas Kosy, Tata McGraw Hill Publishing co.

FULL STACK DEVELOPMENT

B.Tech II Year II Semester
Course Code:

L	T	P	C
3	0	0	3

Course Outcomes:

At the end of the course, student will be able to:

1. Design aesthetically appealing and responsive web pages using HTML and CSS.
2. Develop real-world React web applications and utilize related development tools.
3. Apply agile methodologies for rapid and efficient project development.
4. Build complete end-to-end applications from scratch using Node.js.
5. Understand and implement logical relationships between documents using MongoDB.

UNIT-I

Introduction to Full Stack: MVC pattern, Web Fundamentals. **HTML 5.0:** Basic tags, HTML DOM, Images, Tables, Lists, Forms, Layout, Graphics, span and div tags. **Cascading Style Sheets (CSS):** Types of CSS, CSS Selectors, CSS BOX Model, Text and Font, Color, CSS Positioning and CSS floating, CSS Grid layout Module, CSS Media Queries.

UNIT-II

JavaScript: Data Types & Type Conversion, JSON, Events, String and Date Functions, Local Storage, Object Oriented Programming (OOP) in JS, Arrow Functions, Async/Await, JavaScript Regular Expressions. **Bootstrap:** Introduction of Bootstrap, Container and Container-fluid, Bootstrap Carousel. **Bootstrap Components:** Button, Grid, Table, Form, Alert, Image, Tabs/Pill, Navbar, Modals.

UNIT-III

React JS: Introduction to React, React with JSX, Actual DOM vs React VDOM, Components, Lifecycle, State, Props, Fragments, Events, Router, Forms, Pagination, Tables, Portals, Hook, Signals, React 18 New Features.

Integration of Google MAP API and GPS Location Tracking: Incorporating Google MAP API and GPS Location Tracking for location-based services.

UNIT-IV

Node JS: Modules, Node Package Manager (npm), Creating Web Server, Sending Requests and Handling HTTP requests, Handling User Authentication with NodeJS, File System, Writing a file asynchronously and Other I/O Operations. **Events:** Event Emitter class, Inheriting Events and Returning event emitter. **Express JS:** Introduction to the Express framework – Server-side rendering with Templating Engines, Routing, Middleware, Custom Middleware, JWT Authentication.

UNIT-V

Mongo DB: Introduction, Importance of NoSQL databases, JSON Vs BSON, Data types and examples. CRUD Operations, Data Modelling & Schema Design, Indexing and Aggregation, MongoDB Replication and Sharding.

TEXT BOOKS:

1. Vasan Subramanian, "Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node", Second Edition, Apress Publications, 2019.
2. David Hows, Peter Membrey, EelcoPlugge – "MongoDB Basics", Apress, 2014.

SUGGESTED READING:

1. Ethan Brown, "Web Development with Node and Express", O'Reilly Publishers, First Edition, 2014.

WEB RESOURCES:

1. https://web.stanford.edu/class/cs142/index.html
2. https://nodejs.org/en/docs/
3. https://www.mongodb.com/
4. https://reactjs.org/
5. https://getbootstrap.com/docs/5.0/utilities/api/
6. https://edu.anarcho-copy.org/Programming%20Languages/Node/

ALGORITHMS DESIGN AND ANALYSIS

B.Tech II Year II Semester
Course Code:

L	T	P	C
3	0	0	3

Course Outcomes:

At the end of the course, student will be able to:

1. Apply space and time complexity analysis using asymptotic notations.
2. Design divide-and-conquer algorithms and critically assess their runtime and space trade-offs.
3. Device backtracking and dynamic programming solutions.
4. Apply greedy methods and graph traversal algorithms
5. Analyse and Design branch-and-bound algorithms for NP-hard problems

UNIT-I

Introduction: Algorithm, Performance Analysis-Space complexity, Time complexity, Asymptotic Notations- Big oh notation, Omega notation, Theta notation, and Little oh notation.

Divide and conquer: General method, applications-Binary search, Quick sort, Merge sort, Strassen's matrix multiplication.

UNIT-II

Disjoint Sets: Disjoint set operations, union and find algorithms, Priority Queue- Heaps, Heapsort

Backtracking: General method, applications, n-queens problem, sum of subsets problem, graph coloring, Hamiltonian cycles.

UNIT-III

Dynamic Programming: General method, applications- Optimal binary search tree, 0/1 knapsack problem, All pairs shortest path problem, Traveling salesperson problem, Reliability design.

UNIT-IV

Greedy method: General method, applications- Job sequencing with deadlines, knapsack problem, Minimum cost spanning trees, Single source shortest path problem.

Basic Traversal and Search Techniques: Techniques for Binary Trees, Techniques for Graphs, Connected components, Biconnected components.

UNIT-V

Branch and Bound: General method, applications - Travelling salesperson problem, 0/1 knapsack problem - LC Branch and Bound solution, FIFO Branch and Bound solution.

NP-Hard and NP-Complete problems: Basic concepts, non-deterministic algorithms, NP - Hard and NP-Complete classes, Cook's theorem.

TEXT BOOKS:

1. Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni, and Rajasekaran, University Press.

REFERENCE BOOKS:

1. Design and Analysis of algorithms, Aho, Ullman, and Hopcroft, Pearson education.
2. Introduction to Algorithms, second edition, T. H. Cormen, C.E. Leiserson, R. L. Rivest, and C.Stein, PHI Pvt. Ltd./ Pearson Education.
3. Algorithm Design: Foundations, Analysis and Internet Examples, M.T. Goodrich and R. Tamassia, John Wiley and Sons.

COMPUTER NETWORKS

B.Tech II Year II Semester
Course Code:

L	T	P	C
3	0	0	3

Course Outcomes:

At the end of the course, student will be able to:

1. Gain the knowledge of the basic computer network technology.
2. Gain the knowledge of the functions of each layer in the ISO-OSI and TCP/IP reference model.
3. Obtain the skills of subnetting and routing mechanisms.
4. Familiarity with the essential protocols of computer networks, and how they can be applied in network design and implementation.
5. Understanding working of the protocols through traces captured by a packet sniffer

UNIT-I

Introduction: The Internet, Protocol, Network Edge, Access Networks, Network Core, Packet Switching, Circuit Switching, Delay, Loss, and Throughput in Packet-Switched Networks, Protocol reference models: ISO-OSI, TCP/IP, Types of Network attacks, History of Computer Networking and the Internet.

UNIT-II

Application Layer: Principles of Network Applications, Network Application Architectures, Processes Communicating, Transport Services Available to Applications, Transport Services Provided by the Internet, Application-Layer Protocols, The Web and HTTP, File Transfer: FTP, Electronic Mail in the Internet, SMTP, DNS, Peer-to-Peer Applications, Socket Programming: Creating Network Applications.

UNIT-III

Transport Layer: Transport-Layer Services, Multiplexing and Demultiplexing, Connectionless Transport: UDP, Principles of Reliable Data Transfer, Building a Reliable Data Transfer Protocol, Pipelined Reliable Data Transfer Protocols, Go-Back-N (GBN), Selective Repeat (SR), Connection-Oriented Transport: TCP, The TCP Connection, Segment Structure, Round-Trip Time Estimation and Timeout, Reliable Data Transfer, Flow Control, TCP Connection Management, Principles of Congestion Control, TCP Congestion Control, Fairness.

UNIT-IV

Network Layer: Data and Control plane, Forwarding and Routing, Network Service Models, Virtual Circuit and Datagram Networks, Router working, The Internet Protocol (IP): Forwarding and Addressing in the Internet, Datagram Format, IPv4 Addressing, Internet Control Message Protocol (ICMP), IPv6, IP Security, Routing Algorithms- The Link-State (LS) Routing Algorithm, The Distance Vector (DV) Routing Algorithm, Hierarchical

Routing, Routing in the Internet-Intra-AS Routing in the Internet: RIP, Intra-AS Routing in the Internet: OSPF, Inter-AS Routing: BGP, Broadcast and Multicast Routing, Broadcast Routing Algorithms, Multicasting.

UNIT-V

The Link Layer: The Services Provided by the Link Layer, Error-Detection and -Correction Techniques- Parity Checks, Checksum Methods, Cyclic Redundancy Check (CRC), Hamming code, Multiple Access Links and Protocols, Channel Partitioning Protocols, Random Access Protocols, ARP, Ethernet, Link-Layer Switches, Virtual Local Area Networks (VLANs), Wireless network characteristics, Wireless LAN.

TEXT BOOKS:

1. Computer Networking: A Top-Down Approach – James F.Kurose, Keith W. Ross, Pearson
2. Computer Networks -- Andrew S Tanenbaum, David. j. Wetherall, 5th Edition. Pearson/PHI

REFERENCE BOOKS:

1. Data Communications and Networking – Behrouz A. Forouzan. Third Edition TMH.

MACHINE LEARNING

B.Tech II Year II Semester
Course Code:

L	T	P	C
3	0	0	3

Course Outcomes:

At the end of the course, student will be able to:

1. Distinguish between, supervised, unsupervised and semi-supervised learning.
2. Understand algorithms for building classifiers applied on datasets of non-linearly separableclasses
3. Design an ensemble model to increase the classification accuracy
4. Analyze the principles of RL evolutionary computing algorithms

UNIT-I

Introduction to Machine Learning: Types of Human learning, machine learning process, Well-posedlearning problem, Types of machine learning and comparison, applications of machine learning.Model Preparation, Evaluation and feature engineering: Machine learning activities, Types of data inmachine learning, dataset understanding, plotting and exploration, checking data quality, remediation,data pre-processing, selecting a model, predictive and descriptive models, supervised learning modeltraining, cross-validation and boot strapping, lazy vs eager learner, interpreting the model-underfitting, overfitting, bias-variance trade-off.

UNIT-II

Feature Engineering: Feature transformation - feature construction, feature extraction by PCA, SVD,LDA. Feature subset selection – feature relevancy and redundancy measures. Feature selectionprocess and approaches.

Review of Probability concepts: joint probability, conditional probability, bayes rule, Common discrete and continuous distributions, dealing with multiple random variables, Bayes classifier, Multi-class Classification, Bayesian belief network.

UNIT-III

Supervised Learning: Introduction to supervised learning,

Regression: Introduction of regression, Regression algorithms: Simple linear regression, Multiple linear regression, Polynomial regression model, Logistic regression, Maximum likelihood estimation.

Classification: Classification model and learning steps, Classification algorithms: Naïve Bayes classifier, Distance measures, k-Nearest Neighbor (kNN), Decision tree, Support vector machines, Kernel trick, Random Forest.

UNIT-IV

Unsupervised Learning: Introduction to unsupervised learning, Unsupervised vs supervised learning,Application of unsupervised learning, Clustering and its types, Partitioning method: k-Means and KMedoids, Hierarchical clustering, Density-based methods – DBSCAN.

Parameter for evaluating performance of classification, regression, and clustering model. Improving performance of a model.

UNIT-V

Artificial Neural Network: Biological neuron, Artificial neuron, Weights, Bias, Activation functions – linear, sigmoid, tanh, softmax, ReLU, LeakyReLU, Swish, Neural network architecture, Perceptron – Single layer and Multilayer Perceptron, Learning process in ANN- Feedforward Learning Process, Back Propagation algorithm.

TEXT BOOKS:

1. Saikat Dutt, S. Chjandramouli, Das – Machine Learning, First Edition, Pearson
2. M N Murty, Anathanarayana V S – Machine Learning, First Edition, University Press
3. Tom M Mitchell, —Machine Learning, First Edition, McGraw Hill Education, 2013.

REFERENCE BOOKS:

1. Stephen Marsland, —Machine Learning – An Algorithmic Perspective, Second Edition,
2. Chapman and Hall/CRC Machine Learning and Pattern Recognition Series, 2014.
3. Artificial Neural Networks, B. Yegnanarayana, PHI Learning Pvt. Ltd., 2009

FULL STACK DEVELOPMENT LAB

B.Tech II Year II Semester
Course Code: A254582

L	T	P	C
0	0	2	1

Course Outcomes:

At the end of the course, student will be able to:

1. Apply HTML5 and CSS effectively to design and develop dynamic web pages.
2. Explain and utilize JavaScript concepts in real-world applications.
3. Develop single-page applications using the React framework.
4. Implement server-side application development using Node.js.
5. Design and develop real-time database applications based on system requirements.

List of Experiments:

1. Design a Login Page using HTML, CSS (Media Query), and JavaScript.
2. Design a chessboard pattern using HTML and CSS.
3. Design a calculator application using JavaScript.
4. Create a responsive web page of your class timetable using the Bootstrap Grid System.
5. Create a timer component to start, pause, and reset using ReactJS.
6. Create a React component that checks the strength of a password and displays the result to the user.
7. Design authorized endpoints using JWT (JSON Web Token).
8. Develop a backend application with REST API to perform CRUD operations on student data (use Postman tool).
9. Design a replica set of the student database, insert records in the primary node, and display the records in secondary nodes.
10. Implement MongoDB Sharding to distribute and access student records across multiple shards for improved scalability.

TEXTBOOKS:

1. Vasan Subramanian, "Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node", Second Edition, Apress Publications, 2019.
2. David Hows, Peter Membrey, EelcoPlugge – "MongoDB Basics", Apress, 2014.

SUGGESTED READING:

1. Ethan Brown, "Web Development with Node and Express", O'Reilly Publishers, First Edition, 2014.

WEB RESOURCES:

1. https://web.stanford.edu/class/cs142/index.html
2. https://nodejs.org/en/docs/

3. https://www.mongodb.com/
4. https://reactjs.org/

COMPUTER NETWORKS LAB

B.Tech II Year II Semester
Course Code: A254583

L	T	P	C
0	0	2	1

Course Outcomes:

At the end of the course, student will be able to:

- To be able to work with different network tools
- Implement Encoding and Decoding techniques used in presentation layer
- Implement and analyze routing and congestion issues in network design.
- Analyze error detection and error correction codes.
- Implement data link layer farming methods

LIST OF EXPERIMENTS:

1. Wireshark
   i. Packet Capture Using Wire shark
   ii. Starting Wire shark
   iii. Viewing Captured Traffic
2. Implement Dijsktra's algorithm to compute the shortest path through a network
3. Take an example subnet of hosts and obtain a broadcast tree for the subnet.
4. Implement distance vector routing algorithm for obtaining routing tables at each node
5. Write a program for congestion control using Leaky bucket algorithm.
6. Write a program for frame sorting techniques used in buffers.
7. Implement the data link layer framing methods such as character stuffing and bit stuffing
8. Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCI
9. Write a program tolmplement hamming code algorithm
10. Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism
11. Configure the Local Area Network using Packet Tracer

TEXT BOOKS:

1. Computer Networks, Andrew S Tanenbaum, David. j. Wetherall, 5th Edition. Pearson Education/PHI

REFERENCE BOOKS:

1. An Engineering Approach to Computer Networks, S.Keshav, 2nd Edition, Pearson Education
2. Data Communications and Networking – Behrouz A. Forouzan. 3rd Edition, TMH.

MACHINE LEARNING LAB

B.Tech II Year II Semester
Course Code: A254584

L	T	P	C
0	0	2	1

Course Outcomes:

At the end of the course, student will be able to:

1. Understand modern notions in predictive data analysis
2. Select data, model selection, model complexity and identify the trends
3. Understand a range of machine learning algorithms along with their strengths and weaknesses
4. Build predictive models from data and analyze their performance

LIST OF EXPERIMENTS:

1. Write a python program to compute Central Tendency Measures: Mean, Median, Mode, Measure of Dispersion: Variance, Standard Deviation
2. Study of Python Basic Libraries such as Statistics, Math, Numpy and Scipy
3. Study of Python Libraries for ML application such as Pandas and Matplotlib
4. Write a Python program to implement Simple Linear Regression
5. Implementation of Multiple Linear Regression for House Price Prediction using sklearn
6. Implementation of Decision tree using sklearn and its parameter tuning
7. Implementation of KNN using sklearn
8. Implementation of Logistic Regression using sklearn
9. Implementation of K-Means Clustering
10. Performance analysis of Classification Algorithms on a specific dataset (Mini Project)

TEXT BOOKS:

1. Machine Learning – Tom M. Mitchell, - MGH.

REFERENCE BOOKS:

1. Machine Learning: An Algorithmic Perspective, Stephen Marshland, Taylor & Francis.

DATA VISUALIZATION –POWER BI

B.Tech II Year II Semester
Course Code: A254585

L	T	P	C
0	0	2	1

Course Outcomes:

At the end of the course, student will be able to:

1. Understand How to import data into Tableau.
2. Understand Tableau concepts of Dimensions and Measures.
3. Develop Programs and understand how to map Visual Layouts and Graphical Properties.
4. Create a Dashboard that links multiple visualizations.
5. Use graphical user interfaces to create Frames for providing solutions to real world problems.

LAB PROBLEMS:

1. Understanding Data, What is data, where to find data, Foundations for building Data Visualizations, Creating Your First visualization?
2. Getting started with Tableau Software using Data file formats, connecting your Data to Tableau, creating basic charts(line, bar charts, Tree maps),Using the Show me panel.
3. Tableau Calculations, Overview of SUM, AVR, and Aggregate features, Creating custom calculations and fields.
4. Applying new data calculations to your visualizations, Formatting Visualizations, Formatting Tools and Menus, Formatting specific parts of the view.
5. Editing and Formatting Axes, Manipulating Data in Tableau data, Pivoting Tableau data.
6. Structuring your data, Sorting and filtering Tableau data, Pivoting Tableau data.
7. Advanced Visualization Tools: Using Filters, Using the Detail panel, using the Size panels, customizing filters, Using and Customizing tooltips, Formatting your data with colors.
8. Creating Dashboards & Storytelling, creating your first dashboard and Story, Design for different displays, adding interactivity to your Dashboard, Distributing & Publishing your Visualization.
9. Tableau file types, publishing to Tableau Online, Sharing your visualizations, printing, and Exporting.
10. Creating custom charts, cyclical data and circular area charts, Dual Axis charts

REFERENCE BOOKS:

1. Microsoft Power BI cookbook, Brett Powell, 2nd edition.
2. R Programming for Data Science by Roger D. Peng (References)
3. The Art of R Programming by NormanMatloff Cengage Learning