VIDYA JYOTHI
INSTITUTE OF TECHNOLOGY
AN AUTONOMOUS INSTITUTION

COURSE STRUCTURE & SYLLABI

R-25

For

B. Tech. (Information Technology)

II Year B.Tech. IT - I Semester

Sl. No.	Category	Course Title	L	T	P	Credits
1	BS	Probability and Statistics	3	0	0	3
2	PC	Computer Organization and Microprocessor	3	0	0	3
3	PC	Object Oriented Programming Through Java	3	0	0	3
4	PC	Operating Systems	3	0	0	3
5	PC	Introduction to IoT	3	0	0	3
6	PC	Computational Mathematics Lab	0	0	2	1
7	PC	Object Oriented Programming Through Java Lab	0	0	2	1
8	PC	Operating Systems Lab	0	0	2	1
9	PC	Internet of Things Lab	0	0	2	1
10	SD	Data Visualization-Power BI/Tableau	0	0	2	1
Total Credits			**15**	**0**	**10**	**20**

II Year B.Tech. - II Semester

S.No	Category	Course Title	L	T	P	C
1	PC	Discrete Mathematics	3	0	0	3
2	PC	Data Communications and Computer Networks	3	0	0	3
3	PC	Formal Languages and Automata Theory	3	0	0	3
4	PC	Database Management Systems	3	0	0	3
5	PC	Full Stack Development	3	0	0	3
6	HS	Innovation and Entrepreneurship	2	0	0	2
7	PC	Computer Networks and Simulation Lab	0	0	2	1
8	PC	Database Management Systems Lab	0	0	2	1
9	PC	Full Stack Development Lab	0	0	2	1
10	PC	UI Design- Flutter	0	0	2	1
11	HS	Indian Knowledge System	1	0	0	1
Total			**18**	**0**	**08**	**22**

II Year B.Tech. IT - I Semester

Sl. No.	Category	Course Title	L	T	P	Credits
1	BS	Probability and Statistics	3	0	0	3
2	PC	Computer Organization and Microprocessor	3	0	0	3
3	PC	Object Oriented Programming Through Java	3	0	0	3
4	PC	Operating Systems	3	0	0	3
5	PC	Introduction to IoT	3	0	0	3
6	PC	Computational Mathematics Lab	0	0	2	1
7	PC	Object Oriented Programming Through Java Lab	0	0	2	1
8	PC	Operating Systems Lab	0	0	2	1
9	PC	Internet of Things Lab	0	0	2	1
10	SD	Data Visualization-Power BI/Tableau	0	0	2	1
Total Credits			**15**	**0**	**10**	**20**

COMPUTER ORGANIZATION AND MICROPROCESSOR

Department : IT					II B.Tech I Semester		
Course Code	Hours/Week			Credits	Marks		
	L	T	P	C	CIE	SEE	Total
	3	0	0	3	40	60	100

Course Outcomes:

1. Able to understand the basic components and the design of CPU, ALU and Control Unit.
2. Understand memory hierarchy and its impact on computer cost/performance.
3. Able to understand the advantage of instruction level parallelism and pipelining for high performance Processor design.
4. Understand the instruction set, instruction formats and addressing modes of8086.
5. Able to write assembly language programs to solve problems.

UNIT - I

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

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

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

UNIT - II

Central Processing Unit: The 8086 Processor Architecture, Register organization, Physical memory organization, General Bus Operation, I/O Addressing Capability, Special Processor Activities, Minimum and Maximum mode system and timings.

8086 Instruction Set and Assembler Directives-Machine language instruction formats, Addressing modes, Instruction set of 8086, Assembler directives and operators.

UNIT - III

Assembly Language Programming with 8086- Machine level programs, Machine coding the programs, Programming with an assembler, Assembly Language example programs. Stack structure of 8086, Interrupts and Interrupt service routines, Interrupt cycle of 8086, Interrupt programming, Passing parameters to procedures, Macros, Timings and Delays.

UNIT - IV

Computer Arithmetic: Introduction, Addition and Subtraction, Multiplication Algorithms, Division Algorithms, Floating - point Arithmetic operations. Input-Output Organization: Peripheral Devices, Input- Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupt, Direct memory Access, Input –Output Processor (IOP), Intel 8089 IOP.

UNIT - V

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

Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processors.

TEXT BOOKS:

1. Computer System Architecture, M. Morris Mano, Third Edition, Pearson. (UNITS – I, IV, V)
2. Advanced Microprocessors and Peripherals, K M Bhurchandi, A. K Ray ,3rd edition, McGraw Hill India Education Private Ltd. (UNITS - II, III).

REFERENCES BOOKS:

1. Microprocessors and Interfacing, D V Hall, SSSP Rao, 3 rd edition, McGraw Hill India Education Private Ltd.
2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky: Computer Organization, 5th Edition, Tata McGraw Hill.
3. Computer Organization and Architecture, William Stallings, 9th Edition, Pearson.
4. David A. Patterson, John L. Hennessy: Computer Organization and Design – The Hardware / Software Interface ARM Edition, 4th Edition, Elsevier.

OBJECT ORIENTED PROGRAMMING THROUGH JAVA

Department : IT					II B.Tech I Semester		
Course Code	Hours/Week			Credits	Marks		
	L	T	P	C	CIE	SEE	Total
	3	0	0	3	40	60	100

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, String Buffer, StringBuilder classes and StringTokenizer.

Java.io package: File, Byte Streams: Input Stream, Output Streams, File Input Stream, File Output Stream, 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

Department : IT					II B.Tech. I Semester		
Course Code	Hours/Week			Credits	Marks		
	L	T	P	C	CIE	SEE	Total
	3	0	0	3	40	60	100

Course Outcomes:

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

1. Understand the fundamental concepts, structures of operating systems, and process management techniques.
2. Analyze CPU scheduling, process management, and synchronization mechanisms.
3. Explain memory management techniques and virtual memory concepts with page replacement algorithms.
4. Explain and apply interprocess communication mechanisms and file system interfaces using appropriate system calls
5. Analyze deadlock situations and apply suitable prevention, avoidance, detection, and recovery techniques

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, Process Control block (PCB), 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

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

UNIT-III

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-IV

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

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.

UNIT-V

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

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.

INTRODUCTION TO IOT

Department : IT					II B.Tech I Semester		
Course Code	Hours/Week			Credits	Marks		
	L	T	P	C	CIE	SEE	Total
	3	0	0	3	40	60	100

Course Outcomes: Upon completing this course, the student will be able to:

1. Explain the architecture, layers, protocols, and sensor networks of IoT and their roles in real-world applications.
2. Differentiate IoT and M2M communication models and implement interoperable systems using Arduino programming.
3. Apply Python programming on Raspberry Pi to interface sensors and develop IoT use-case applications.
4. Design and implement SDN-based IoT solutions using Raspberry Pi for efficient network management.
5. Analyze and select appropriate sensor technologies and cloud storage models for specific IoT applications.

UNIT- I

Introduction to IoT and Sensor Networks

Introduction to Internet of Things (IoT), Characteristics and Applications of IoT, IoT Architecture and Reference Models (IETF, ITU-T)

Physical Design of IoT: Devices, Gateways, and Data Centers

Functional Blocks of IoT: Sensing, Actuation, Communication, Enabling Technologies: RFID, Wireless Sensor Networks, Nanotechnology

Basics of Networking and Communication Protocols: MQTT, CoAP, ZigBee, HTTP Sensor Networks: Types, Topologies, and Protocols

Introduction to IoT Security and Privacy Fundamentals.

UNIT- II

Machine to Machine (M2M) and Embedded Programming for IoT

Machine-to-Machine Communications Overview, Difference between IoT and M2M, Interoperability in IoT: Standards and Protocols.

Introduction to Arduino Programming for IoT, Integration of Sensors and Actuators with Arduino. Hands-on Exercises: Sensor Data Acquisition and Actuator Control, Basic Communication Protocols Implementation on Arduino, IoT Device Interoperability Challenges and Solutions.

UNIT- III

Raspberry Pi with Python Programming for IoT

Introduction to Python Programming Basics

Overview of Raspberry Pi and Its Role in IoT, Interfacing Raspberry Pi with Sensors and Actuators (UART, SPI, I2C).

Data Acquisition and Local Processing on Raspberry Pi, Sending Data to Cloud Platforms. Implementation of IoT Projects Using Raspberry Pi

Case Studies: Smart Home Automation, Healthcare Monitoring, Environmental Sensing

UNIT- IV

Network Management and Software Defined Networking (SDN) for IoT

Introduction to Software Defined Networking (SDN), SDN Architecture: Controller, Data Plane, Control Plane, Benefits of SDN in IoT: Flexibility, Centralized Control, Security, Traffic Management, SDN Models: Open SDN, API SDN, Overlay, Hybrid Data Handling, Analytics, and Security Considerations in SDN for IoT

Network Management Challenges in IoT, Integration of SDN with IoT Networks

UNIT- V

Cloud and Edge Computing Models with IoT Use Cases

Introduction to Cloud Computing and Cloud Storage Models, Edge and Fog Computing Concepts for IoT, Web Servers and Cloud Platforms for IoT (AWS IoT, Azure IoT, etc.), Sensor-Cloud Architecture Data Analytics and Visualization Techniques for IoT Data

IoT Use Cases: Smart Cities, Smart Homes, Connected Vehicles, Smart Grid, Industrial IoT Case Studies: Agriculture, Healthcare, Activity Monitoring.

TEXT BOOKS

1. Pethuru Raj and Anupama C. Raman "The Internet of Things: Enabling Technologies, Platforms, and Use Cases", CRC Press.
2. Vijay Madisetti, Arshdeep Bahga, “Internet of Things: A Hands-On Approach”

REFERENCE BOOKS

1. Terokarvinen, kemo, karvinen and villeyvaltokari, “Make sensors”: 1st edition,2014, Maker media.
2. Waltenegus Dargie, Christian Poellabauer, Fundamentals of Wireless Sensor Networks: Theory and Practice.
3. Charles Bell, Beginning Sensor networks with Arduino and Raspberry Pi, 2013, Apress.
4. Fei Hu, Security and Privacy in Internet of Things (IoTs), CRC Press, Taylor & Francis Group.
5. S. Sahoo, S. Sahoo, S. Mishra, Software-Defined Networking for Future Internet Technology: Concepts and Applications, Routledge.
6. Honbo Zhou, The Internet of Things in the Cloud: A Middleware Perspective. 2012, CRC Press (Taylor & Francis).

COMPUTATIONAL MATHEMATICS LAB
(Using Python/MATLAB software)

Department : IT					II B.Tech I Semester		
Course Code	Hours/Week			Credits	Marks		
	L	T	P	C	CIE	SEE	Total
	0	0	2	1	40	60	100

Course outcomes:
After learning the contents of this paper, the student must be able to

1. **Apply numerical and computational techniques** to develop Python/MATLAB programs for computing eigenvalues and eigenvectors of matrices.
2. **Formulate and solve algebraic and transcendental equations** using suitable numerical methods, and **implement solutions** for systems of linear equations using Python/MATLAB.
3. **Develop computational models** to solve first-order ordinary differential equations using appropriate analytical and numerical methods in Python/MATLAB.
4. **Analyze and solve higher-order linear differential equations with constant coefficients** using Python/MATLAB for engineering and scientific applications.

*** Visualize all solutions Graphically through programmes**

UNIT - I: Eigen values and Eigenvectors: **6P**

Programs:

- Finding real and complex Eigen values.
- Finding Eigen vectors.

UNIT-II: Solution of Algebraic and Transcendental Equations **6P**

Bisection method, Newton Raphson Method

Programs:

- Root of a given equation using Bisection method.
- Root of a given equation Newton Raphson Method.

UNIT-III: Linear system of equations: **6P**

Jacobi‟s iteration method and Gauss-Seidal iteration method

Programs:

- Solution of given system of linear equations using Jacobi‟s method
- Solution of given system of linear equations using Gauss-Seidal method

UNIT-IV: First-Order ODEs **8P**

Exact and non-exact equations, Applications: exponential growth/decay, Newton's law of cooling.

Programs:

- Solving exact and non-exact equations
- Solving exponential growth/decay and Newton's law of cooling problems

UNIT-V: Higher order linear differential equations with constant coefficients **6P**

Programs:

- Solving homogeneous ODEs
- Solving non-homogeneous ODEs

TEXT BOOKS:

1. MATLAB and its Applications in Engineering, Rajkumar Basal, Ashok Kumar Geo, Manoj Kumar Sharma, Pearson publication.
2. Kenneth A. Lambert, The fundamentals of Python: First Programs, 2011, Cengage Learnings.
3. Think Python First Edition, by Allen B. Downey, Orielly publishing.
4. Introduction to Python Programming, William Mitchell, Povel Solin, Martin Novak et al., NCLab Public Computing, 2012.
5. Introduction to Python Programming, ©Jacob Fredslund, 2007.

REFERENCE BOOKS:

1. An Introduction to Python, John C. Lusth, The University of Alabama, 2011.
2. Introduction to Python, ©Dave Kuhlman, 2008.

OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB

Department : IT					II B.Tech I Semester		
Course Code	Hours/Week			Credits	Marks		
	L	T	P	C	CIE	SEE	Total
	0	0	2	1	40	60	100

Course Outcomes:

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

1. **Apply object-oriented programming principles** to design and develop Java programs for real-world problem solving.
2. **Implement advanced OOP features** such as abstract classes, interfaces, and inheritance in Java applications.
3. **Develop multithreaded Java programs** and **use the Java Collections Framework** for efficient data handling and concurrent execution.
4. **Design and implement basic web applications** using Java Servlets and JDBC for database connectivity.

List of Experiments:

1. Write a Java Program to accept Student Name and Roll Number as command Line Arguments and Display the output.
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 Method Overriding.
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

Department : IT					II B.Tech I Semester		
Course Code	Hours/Week			Credits	Marks		
	L	T	P	C	CIE	SEE	Total
	0	0	2	1	40	60	100

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

1. **Simulate and analyze CPU scheduling algorithms** and **evaluate their performance** using standard scheduling criteria.
2. **Apply UNIX/Linux system calls** to implement file handling, process creation, and inter-process communication.
3. **Implement synchronization and deadlock handling solutions**, including Producer–Consumer using semaphores and Banker's Algorithm for deadlock avoidance.
4. **Simulate and analyze memory management techniques**, including paging and page replacement policies, to assess efficient memory utilization.

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 implement the Producer – Consumer problem using semaphores using UNIX/LINUX system calls.
4. Write C programs to simulate the following memory management techniques
   a) Paging b) Segmentation
5. Write C programs to simulate Page replacement policies
   a) FCFS b) LRU c) Optimal
6. Write C programs to illustrate the following IPC mechanisms
   a) Pipes b) FIFOs c) Message Queues d) Shared Memory
7. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance.

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.

INTERNET OF THINGS LAB

Department : IT					II B.Tech I Semester		
Course Code	Hours/Week			Credits	Marks		
	L	T	P	C	CIE	SEE	Total
	0	0	2	1	40	60	100

Course Outcomes:

1. Illustrate various IoT devices, sensors, and actuators, including Arduino, Raspberry Pi, and microcontrollers.
2. Identify sensors technologies used in IoT applications and their working principles.
3. Apply programming languages like Python, C/C++ in IoT applications.
4. Develop skills in collecting data from sensors and processing it using microcontrollers or edge computing devices.

List of Programs

1. Installing OS on Raspberry Pi
   a. Installation using Pi Imager
   b. Installation using image file
      - Downloading an Image
      - Writing the image to an SD card using Linux using Windows
      - Booting up Follow the instructions given in the URL https://www.raspberrypi.com/documentation/computers/gettingstarted.html
2. Accessing GPIO pins using Python
   a. Installing GPIO Zero library.
      First, update your repositories list:
      sudo apt update
      Then install the package for Python 3:
      sudo apt install python3-gpiozero
   b. Blinking an LED connected to one of the GPIO pin
   c. Adjusting the brightness of an LED Adjust the brightness of an LED (0 to 100, where 100 means maximum brightness) using the in-built PWM wavelength.
3. Using Raspberry Pi
   a. Calculate the distance using a distance sensor.
   b. Basic LED functionality using switch.
4. Using Arduino
   a. Calculate the distance using a distance sensor.
   b. Basic LED functionality using switch.
   c. Calculate temperature using a temperature sensor
5. Using Node MCU
   a. Including required libraries of Node MCU
   b. Calculate the distance using a distance sensor.
   c. Basic LED functionality using switch.

d. Calculate temperature using a temperature sensor

6. Using ESP32
   a. Including required libraries of ESP32.
   b. Calculate the distance using a distance sensor.
   c. Basic LED functionality using switch.
   d. Calculate temperature using a temperature sensor.

7. Collecting Sensor Data
   a. DHT Sensor interface
   - Connect the terminals of DHT GPIO pins of Raspberry Pi.
   - Import the DHT library using import Adafruit, DHT.
   - Read sensor data and display it on screen.
   - Read sensor data and display it on the Cloud platform using Thing Speak (API).
   - Read sensor data and display it through the Blynk App.

8. Capstone Project: End-to-End IoT Solution Integrating Sensors, Network, and Cloud.

TEXT BOOKS:

1. Arshdeep Bahga and Vijay Madisetti, Internet of Things - A Hands-on Approach, 2015, Universities Press.
2. Matt Richardson & Shawn Wallace, Getting Started with Raspberry Pi, 2014, O"Reilly.

REFERENCE BOOKS:

1. Bernd Scholz-Reiter, Florian Michahelles, "Architecting the Internet of Things", ISBN 978-3- 642-19156-5 e-ISBN 978-3-642-19157-2, Springer, 2016.
2. N. Ida, Sensors, Actuators and Their Interfaces, Scitech Publishers, 2014.

DATA VISUALIZATION - POWER BI/ TABLEAU

Department : IT					II B.Tech I Semester		
Course Code	Hours/Week			Credits	Marks		
	L	T	P	C	CIE	SEE	Total
	0	0	2	1	40	60	100

Course Outcomes:

1. Understand how to import data into Tableau.
2. Understand concepts of Dimensions and Measures.
3. Develop Programs and understand how to map Visual Layouts and Graphical Properties
4. Create Dashboard, custom charts, and, publish to tableau online for any real time dataset

List of Programs

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.
11. Visualize various data patterns taking any dataset from Kaggle.
12. Visualize data patterns using Google Charts by creating interactive Line, Bar, and Pie charts along with Dashboards using HTML and JavaScript.

REFERENCE BOOKS:

1. Brett Powell, Microsoft Power BI cookbook, 2nd edition.
2. Roger D. Peng, R Programming for Data Science
3. Norman Matloff Cengage Learning India, The Art of R Programming.

WEB RESOURCE:

1. https://developers.google.com/chart/interactive/docs.

II Year B.Tech. - II Semester

S.No	Category	Course Title	L	T	P	C
1	PC	Discrete Mathematics	3	0	0	3
2	PC	Data Communications and Computer Networks	3	0	0	3
3	PC	Formal Languages and Automata Theory	3	0	0	3
4	PC	Database Management Systems	3	0	0	3
5	PC	Full Stack Development	3	0	0	3
6	HS	Innovation and Entrepreneurship	2	0	0	2
7	PC	Computer Networks and Simulation Lab	0	0	2	1
8	PC	Database Management Systems Lab	0	0	2	1
9	PC	Full Stack Development Lab	0	0	2	1
10	PC	UI Design- Flutter	0	0	2	1
11	HS	Indian Knowledge System	1	0	0	1
Total			**18**	**0**	**08**	**22**

DISCRETE MATHEMATICS

Department : IT					II B.Tech II Semester		
Course Code	Hours/Week			Credits	Marks		
	L	T	P	C	CIE	SEE	Total
	3	0	0	3	40	60	100

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 & Truth Tables, Normal Forms (DNF, CNF, PDNF, PCNF).

UNIT-II

Theory of Inference for the Statement Calculus, The Predicate Calculus, Inference Theory of the Predicate Calculus.

Set theory: Introduction, Basic Concepts of Set Theory,

Relations: Special Properties of Binary Relations, Partially ordered relations, Hasse diagrams, Lattices.

UNIT-III

Functions: Composition of functions Inverse Functions

Algebraic Structures: Algebraic Systems-Examples and General Properties, Semi groups and Monoids, groups, sub groups, homomorphism, Isomorphism.

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.

DATA COMMUNICATIONS AND COMPUTER NETWORKS

Department : IT					II B.Tech II Semester		
Course Code	Hours/Week			Credits	Marks		
	L	T	P	C	CIE	SEE	Total
	3	0	0	3	40	60	100

Course Outcomes:

1. **Understand the fundamentals of data communication and computer networks**, including network types, topologies, and transmission media.
2. **Describe the functions and services of the Data Link Layer**, including framing, error detection/correction, and flow control mechanisms.
3. **Analyze Network Layer concepts** such as routing, addressing, and congestion control in packet-switched networks.
4. **Understand Transport Layer mechanisms** including port addressing, flow control, error control, and congestion management.
5. **Illustrate the working principles of Application Layer protocols** such as HTTP, FTP, SMTP, and DNS in real-world networking scenarios.

UNIT – I

Data Communications: Components – Direction of Data flow – Networks – Components and Categories – Types of Connections – Topologies –Protocols and Standards – ISO / OSI model, Example Networks such as ATM, Frame Relay, ISDN

Physical layer: Transmission modes, Multiplexing, Transmission Media, Switching, Circuit Switched Networks, Datagram Networks, Virtual Circuit Networks.

UNIT – II

Data link layer: Introduction, Framing, and Error – Detection and Correction – Parity – LRC – CRC Hamming code, Flow and Error Control, Noiseless Channels, Noisy Channels, HDLC, Point to Point Protocols. 111 Medium Access sub layer: ALOHA, CSMA/CD, LAN – Ethernet IEEE 802.3, IEEE 802.5 – IEEE 802.11, Random access, Controlled access, Channelization.

UNIT – III

Network layer: Logical Addressing, Internetworking, Tunneling, Address mapping, ICMP, IGMP, Forwarding, Uni-Cast Routing Protocols, Multicast Routing Protocols.

UNIT – IV

Transport Layer: Process to Process Delivery, UDP and TCP protocols, Data Traffic, Congestion, Congestion Control, QoS, Integrated Services, Differentiated Services, QoS in Switched Networks.

UNIT – V

Application Layer: Domain name space, DNS in Internet, Electronic Mail, SMTP, FTP, WWW, HTTP, SNMP.

TEXT BOOKS:

1. Data Communications and Networking, Behrouz A. Forouzan, Fourth Edition TMH.

REFERENCE BOOKS:

1. Computer Networks, Andrew S Tanenbaum, 6th Edition. Pearson Education.
2. Computer Networking: A Top-Down Approach Featuring the Internet. James F. Kurose & Keith W. Ross, 3 rd Edition, Pearson Education
3. Data communications and Computer Networks, P.C Gupta, PHI.
4. An Engineering Approach to Computer Networks, S. Keshav, 2nd Edition, Pearson Education.

FORMAL LANGUAGES AND AUTOMATA THEORY

Department : IT					II B.Tech II Semester		
Course Code	Hours/Week			Credits	Marks		
	L	T	P	C	CIE	SEE	Total
	3	0	0	3	40	60	100

Course Outcomes

1. **Explain the concept of abstract machines** and their computational power in recognizing formal languages.
2. **Apply finite state machines** (DFA, NFA) and regular expressions to model and solve computing problems.
3. **Analyze and design context-free grammars** and pushdown automata for specifying formal languages.
4. **Differentiate between decidable and undecidable problems** using formal proofs and examples.
5. **Interpret the limits of computation** through Turing machines and undecidability concepts in language recognition.

UNIT - I

Introduction to Finite Automata: Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory – Alphabets, Strings, Languages, Problems.

Nondeterministic Finite Automata: Formal Definition, an application, Text Search, Finite Automata with Epsilon-Transitions.

Deterministic Finite Automata: Definition of DFA, How A DFA Process Strings, The language of DFA, Conversion of NFA with €-transitions to NFA without €-transitions. Conversion of NFA to DFA, Moore and Melay machines

UNIT - II

Regular Expressions: Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Conversion of Finite Automata to Regular Expressions.

Pumping Lemma for Regular Languages, Statement of the pumping lemma, Applications of the Pumping Lemma.

Closure Properties of Regular Languages: Closure properties of Regular languages, Decision Properties of Regular Languages, Equivalence and Minimization of Automata.

UNIT - III

Context-Free Grammars: Definition of Context-Free Grammars, Derivations Using a Grammar, Leftmost and Rightmost Derivations, the Language of a Grammar, Sentential Forms, Parse Tress, Applications of Context-Free Grammars, Ambiguity in Grammars and Languages.

Push Down Automata: Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA's and CFG's, Acceptance by final state, Acceptance by empty stack, Deterministic Pushdown Automata. From CFG to PDA, From PDA to CFG.

UNIT - IV

Normal Forms for Context- Free Grammars: Eliminating useless symbols, Eliminating €-Productions. Chomsky Normal form Greibach Normal form.

Pumping Lemma for Context-Free Languages: Statement of pumping lemma, Applications **Closure Properties of Context-Free Languages:** Closure properties of CFL"s, Decision Properties of CFL's Turing Machines: Introduction to Turing Machine, Formal Description, Instantaneous description, The language of a Turing machine

UNIT - V

Types of Turing machine: Turing machines and halting

Undecidability: Undecidability, A Language that is Not Recursively Enumerable, An Undecidable Problem that is RE, Undecidable Problems about Turing Machines, Recursive languages, Properties of recursive languages, Post's Correspondence Problem, Modified Post Correspondence problem, Other Undecidable Problems, Counter machines.

TEXT BOOKS:

1. Introduction to Automata Theory, Languages, and Computation, 3nd Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2. Theory of Computer Science – Automata languages and computation, Mishra and Chandrasekaran, 2nd edition, PHI.

REFERENCE BOOKS:

1. Introduction to Languages and The Theory of Computation, John C Martin, TMH.
2. Introduction to Computer Theory, Daniel I.A. Cohen, John Wiley.
3. A Text book on Automata Theory, P. K. Srimani, Nasir S. F. B, Cambridge University Press.
4. Introduction to the Theory of Computation, Michael Sipser, 3rd edition, Cengage Learning.
5. Introduction to Formal languages Automata Theory and Computation Kamala Krithivasan, Rama R, Pearson.

DATABASE MANAGEMENT SYSTEMS

Department : IT					II B.Tech II Semester		
Course Code	Hours/Week			Credits	Marks		
	L	T	P	C	CIE	SEE	Total
	3	0	0	3	40	60	100

Course Outcomes:

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

1. Gain knowledge of fundamentals of DBMS, database design.
2. Apply basic SQL concepts for effective retrieval and management of data.
3. Analyze various Normal Forms to carry out the concepts of Schema Refinement.
4. Understand the basics of transaction processing and concurrency control.
5. Understand the 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, Elmasri Navrate, 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.

FULL STACK DEVELOPMENT

Department : IT					II B.Tech II Semester		
Course Code	Hours/Week			Credits	Marks		
	L	T	P	C	CIE	SEE	Total
	3	0	0	3	40	60	100

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, Eelco Plugge – “MongoDB Basics”, Apress, 2014.

REFERECNE BOOK:

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/

COMPUTER NETWORKS LAB AND SIMULATION LAB

Department : IT					II B.Tech II Semester		
Course Code	Hours/Week			Credits	Marks		
	L	T	P	C	CIE	SEE	Total
	0	0	2	1	40	60	100

Course Outcomes:

1. **Implement and analyze Data Link Layer framing techniques** along with error detection and correction methods.
2. **Analyze and apply routing algorithms and congestion control techniques** in network design.
3. **Implement encoding and decoding techniques** used at the Presentation Layer for data representation and security.
4. **Use standard network tools** to monitor, analyze, and troubleshoot network performance.

List of Experiments

1. Implement the data link layer framing methods such as character, character-stuffing and bit stuffing.
2. Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP
3. 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.
4. Implement Dijsktra‟s algorithm to compute the shortest path through a network
5. Take an example subnet of hosts and obtain a broadcast tree for the subnet.
6. Implement distance vector routing algorithm for obtaining routing tables at each node.
7. Implement data encryption and data decryption
8. Write a program for congestion control using Leaky bucket algorithm.
9. Write a program for frame sorting techniques used in buffers.
10. **Wireshark**
    i. Packet Capture Using Wire shark
    ii. Starting Wire shark
    iii. Viewing Captured Traffic
    iv. Analysis and Statistics & Filters.

1. How to run Nmap scan
2. Operating System Detection using Nmap
3. Do the following using NS2 Simulator
    I. NS2 Simulator-Introduction
    II. Simulate to Find the Number of Packets Dropped
    III. Simulate to Find the Number of Packets Dropped by TCP/UDP
    IV. Simulate to Find the Number of Packets Dropped due to Congestion
    V. Simulate to Compare Data Rate & Throughput.
    VI. Simulate to Plot Congestion for Different Source/Destination
    VII. Simulate to Determine the Performance with respect to Transmission of Packets

TEXT BOOK:

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

REFERENCES:

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

DATABASE MANAGEMENT SYSTEMS LAB

Department : IT					II B.Tech II Semester		
Course Code	Hours/Week			Credits	Marks		
	L	T	P	C	CIE	SEE	Total
	0	0	2	1	40	60	100

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

1. **Apply conceptual database design principles** using the Entity–Relationship (E–R) model to design relational schemas.
2. **Use SQL DDL and DML commands** to create, modify, and manipulate data in relational databases.
3. **Execute advanced SQL queries** involving joins, constraints, set operations, and nested/correlated subqueries.
4. **Develop database applications** using programming components such as triggers, procedures, and cursors.

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, Elmasri Navrate, 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.

FULL STACK DEVELOPMENT LAB

Department : IT					II B.Tech II Semester		
Course Code	Hours/Week			Credits	Marks		
	L	T	P	C	CIE	SEE	Total
	0	0	2	1	40	60	100

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

1. **Design and develop responsive web pages** using HTML5 and CSS.
2. **Apply JavaScript concepts** to build interactive and dynamic client-side web applications.
3. **Develop single-page applications (SPAs)** using the React framework.
4. **Design and implement full-stack web applications** using Node.js and real-time database technologies.

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, Eelco Plugge – "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/

UI DESIGN - FLUTTER

Department : IT					II B.Tech II Semester		
Course Code	Hours/Week			Credits	Marks		
	L	T	P	C	CIE	SEE	Total
	0	0	2	1	40	60	100

Course Outcomes:

1. Design responsive web pages using HTML, CSS, Bootstrap, and JavaScript.
2. Develop backend applications using Java and Node.js.
3. Build single-page applications with React.js.
4. Create Flutter apps with custom widgets, layouts, and forms.

List of Experiments: Students need to implement the following experiments

1 Build a responsive website with registration, login, catalog, and cart pages using HTML, CSS3, Bootstrap, and JavaScript validation.

2 Create a react application for the student management system having registration, login, contact, about pages and implement routing to navigate through these pages

3 Create a service in react that fetches the weather information from openweathermap.org and the display the current and historical weather information using graphical representation using chart.js

4 Develop an express web application that can interact with REST API to perform CRUD operations on student data. (Use Postman)

5 Implement JWT authentication in Node.js to create secure endpoints.

6 Create a TODO application in react with necessary components and deploy it into github.

7 Integrate a React frontend with Node.js backend to perform CRUD operations on a shared dataset (e.g., MongoDB, MySQL.)

8 Build a Django REST API for managing student data, tested using Postman.

9 Install Flutter and Dart SDK; Write Dart programs on data types, control flow, Functions, Class & Objects and collections to understand syntax and features.

10 Create a Flutter app showcasing common widgets (Text, Image, Container, Card, ListView).

11 Design a responsive UI using Row, Column, Stack, media queries, and breakpoints.

12 Implement screen navigation using Navigator and named routes.

13 Use stateful & stateless widgets; manage state using Provider or setState.

14 Design a form in Flutter with input fields, validation, and error handling (e.g., student registration form).

15 Fetch and display data from a REST API (e.g., weather or student info) in Flutter.

16 Add basic animations (fade or slide) to a Flutter app for enhancing UI interactions

TEXT BOOK:

1. Marco L. Napoli, Beginning Flutter: A Hands-on Guide to App Development.

REFERENCE BOOKS:

1. Jon Duckett, Beginning HTML, XHTML, CSS, and JavaScript, Wrox Publications, 2010
2. Bryan Basham, Kathy Sierra and Bert Bates, Head First Servlets and JSP, O"Reilly Media, 2nd Edition, 2008.

3. Vasan Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node, 2nd Edition, A Press.

INDIAN KNOWLEDGE SYSTEM

Department : IT					II B.Tech II Semester		
Course Code	Hours/Week			Credits	Marks		
	L	T	P	C	CIE	SEE	Total
	1	0	0	1	40	60	100

Bharat is considered one of the oldest civilizations of the world. Some of the archaeological evidences proved the existence of Indus Valley Civilization in 7000 B.C. Bhartiya traditions, culture, cultural activities, rituals, sacraments, painting, art of dancing, art of singing etc. is being practised till the modern times without knowing scientific approaches behind that. Eternity of Indian knowledge system proved itself that not only many rituals but also many traditions, many streams of knowledge like astrology, mathematics, physics, chemistry, biology, language studies, yoga and meditation had been following from the starting till now with some changes, in the form of traditions.

This course is for undergraduate students to inculcate Indian values. It will promote advance study and inter disciplinary research on all aspects of the Indian knowledge system.

Course Objectives: This course aims:

1. To provide a tribune of the rich culture and traditions of Indian knowledge system to students of various disciplines.
2. To introduce historical account on the education and scientific literature available in ancient Indian traditions and its connections with ancient Indian Philosophy
3. To give insights about the applications of Bharatiya Jnana Parampara
4. To introduce Indian approach towards health and wellbeing
5. To elaborate vast contribution of ancient Indian researchers, engineers, scientists and architects to the modern world

Course Outcomes: Students will be able to:

1. Understand nature, scope and related fields of Indian knowledge system.
2. Demonstrate the scientific literature available in ancient Indian traditions
3. Understanding the application of Bharatiya Jnana Parampara
4. Understand Indian approach towards Wellbeing
5. Appreciate vast contribution of ancient Indian researchers, engineers, scientists and architects to the modern world

UNIT-1: **Introduction to Indian Knowledge Systems**

Meaning**,** Nature, Scope and Salient Aspects of Bharatiya Jnana Parampara - Introduction to Vedas, Upanishads, Vidya, Kala, Jnana, Shastra - Practices and Continuity of Tradition

UNIT-2: Overview of History of Indian Education and Scientific Literature

Gurukul System - Role of Sanskrit in Natural Language Processing - Scientific Literature - Vedic Literature - Available Scientific Treatises - Interlinkings

UNIT-3: Introduction to Scientific Theories from Pure Sciences from Ancient Indian Knowledge Systems

Overview of theories from available ancient Indian Literature about Physics, Chemistry and Mathematics - Interlinkings and applications

UNIT-4: Introduction to Ancient Indian Wellness Systems

Concept of Wellness – Yoga System - Ayurveda System - Ancient Indian Aesthetics

UNIT-5: Development of Engineering, Science, Technology & Fine Arts in India

Various Industries - Silk, Cotton and Ship Building - Evolution of Indian Fine Arts – Cave and Temple Architecture, Vastu - Vidya, Sculpture, Forts and Stepwells, Observatories and Paintings - Music and Natyakala - Cultural Traditions & Folk Arts

- **Pedagogy for Teachers: Apart from Class Room Instruction, the following Methods are Suggested.**

1. Project based activities and learning.
2. Presentation and case studies.
3. Film screening and book reviews.
4. Visit to historical places, archives centre, research centre or library nearby.

***Note**: Activities mentioned above are only suggestive. Teacher-educators should encourage students to be innovative.*

Suggested Readings:

1. B. Mahadevan, Bhat Vinayak and Nagendra Pavan R.N., (2022) '*Introduction to Indian Knowledge Systems: Concepts and Applications*' PHI learning PVT, New Delhi ISBN [9789391818203]
2. Dharmapal (1971) '*Indian Science and Technology in the Eighteenth Century*'. Other India Press, Goa.
3. Kapil Kapoor, Singh Avdhesh Kumar, (2005) '*Indian Knowledge Systems*' D.K. Printworld (P) Ltd. ISBN 10: 8124603367 / ISBN 13: 9788124603369
4. Chakradeo, Ujwala, Temples of Bharat, Aayu Publications, New Delhi, 2024.
5. D.N. Bose, S.N. Sen and B. V. Subbarayappa, *A Concise History of Science in India*, Indian National Science Academy, New Delhi, 2009.
6. Datta B. and A. N. Singh, *History of Hindu Mathematics: Parts I and II*, Asia Publishing House, Bombay, 1962.
7. Kapoor, K. (2021), *Indian Knowledge System: Nature, Philosophy, Character in Indian Knowledge System*, vol. 1, Pub. Indian Institute of Advanced Studies, Shimla
8. Mahadevan, B., Bhat, V.R., Pavana, N. (2022), Philosophical Systems, in Introduction to Indian Knowledge System, Pub. PHI Learning, New Delhi.
9. Mahadevan, B., Bhat, V.R., Pavana, N. (2022), Knowledge: Framework and Classification, in Introduction to Indian Knowledge System, Pub. PHI Learning, New Delhi.

Video Resources:

1. Introductory lectures by Prof. Gauri Mahulikar
2. Introductory lectures by Prof. Kapil Kapoor

Websites:

- https://iksin dia.org/index.php
- Official Website of IKS- Indian Knowledge System
- https://www.youtube.com/watch?v=uKcf-hSlcUE
- Address by Prof Kapil Kapoor | Indian Institute of Advanced Study (FDP 2021)
- https://www.youtube.com/watch?v=MDJTXNiH2_A
- Mukul Kanitkar on Bharatiya Knowledge System
- https://www.youtube.com/watch?v=uARMhv97pjk
- https://www.youtube.com/watch?v=oTwgf56GbsA
- Scientific History of India | Mukul Kanitkar Lecture in DTU
- https://youtu.be/gNJNmPJqXJc?si=WFBbuUT65mLZzpOW
- Ancient India's Scientific Achievements & Contribution in Mathematics, Astronomy, Science & Medicine