B. Tech – III Year I Semester

S.No.	Category	Title	L	T	P	C
1	Professional Core	Deep Learning	3	0	0	3
2	Professional Core	Computer Networks	3	0	0	3
3	Professional Core	Operating Systems	3	0	0	3
4	Professional Elective-I	1. Object Oriented Analysis and Design 2. Automata Theory & Compiler Design 3.Soft computing 4.Internet of Things	3	0	0	3
5	Open Elective- I	OR Entrepreneurship Development & Venture Creation	3	0	0	3
6	Professional Core	Deep learning Lab	0	0	3	1.5
7	Professional Core	Computer Networks Lab	0	0	3	1.5
8	Skill Enhancement course	Full Stack development-2 / SWAYAM Plus – Data Engineer / AI Engineer /	0	1	2	2
9	ES	Tinkering Lab (*User Interface Design using Flutter*) / SWAYAM Plus - Android Application Development (with Flutter)	0	0	2	1
10	Evaluation of Community Service Project Internship		-	-	-	2
		Total	**15**	**1**	**10**	**23**

Open Electives, offered to other department students:

Open Elective I: Operating Systems / Computer Organization and Architecture

Open Elective II: Database Management Systems

Open Elective III: Object Oriented Programming Through Java

Open Elective IV: Computer Networks / Software Engineering / IOT Based Smart Systems

B. Tech III Year – I Semester

III Year I Semester	DEEP LEARNING	L	T	P	C
		3	0	0	3

Course Objectives:

The objective of this course is to cover the fundamentals of neural networks as well as some advanced topics such as recurrent neural networks, long short-term memory cells and convolution neural networks.

Course Outcomes:

After completion of course, students would be able to:

- Explore feed forward networks and Deep Neural networks
- Mathematically understand the deep learning approaches and paradigms
- Apply the deep learning techniques for various applications

UNIT-I:

Basics- Biological Neuron, Idea of computational units, McCulloch–Pitts unit and Thresholding logic, Linear Perceptron, Perceptron Learning Algorithm, Linear separability, Convergence theorem for Perceptron Learning Algorithm.

UNIT-II:

Feed forward Networks- Multilayer Perceptron, Gradient Descent, Back propagation, Empirical Risk Minimization, regularization, auto encoders. Deep Neural Networks: Difficulty of training deep neural networks, Greedylayer wise training.

UNIT-III:

Better Training of Neural Networks –Newer optimization methods for neural networks (Adagrad, adadelta, rmsprop, adam, NAG), second order methods for training, Saddle point problem in neural networks, Regularization methods (dropout, drop connect, batch normalization).

UNITIV:

Recurrent Neural Networks- Back propagation through time, Long Short-Term Memory, Gated Recurrent Units, Bidirectional LSTMs, Bidirectional RNNs.

Convolutional Neural Networks: LeNet, AlexNet. Generative models: Restrictive Boltzmann Machines (RBMs), Introduction to MCMC and Gibbs Sampling, gradient computations in RBMs, Deep Boltzmann Machines.

UNITV:

Recent trends-Variational Autoencoders,Transformers,GPT Applications: Vision, NLP, Speech

TextBooks:

1. DeepLearning, Ian Good fellow and Yoshua Bengio and Aaron Courville, MIT Press, 2016

Reference Books:

1. NeuralNetworks:ASystematicIntroduction,RaúlRojas,1996
2. PatternRecognitionandMachineLearning,ChristopherBishop,2007
3. DeepLearningwithPython,FrançoisChollet,ManningPublications,2017

III Year I Semester	COMPUTER NETWORKS	L	T	P	C
		3	0	0	3

Course Objectives:

The course is designed to

- To understand the different types of networks
- To develop an understanding the principles of computer networks.
- To familiarize with Reference model OSI and TCP/IP
- To understand various layers of Reference models functions
- To explore network protocols

Course Outcomes (CO):

After completion of the course, students will be able to

- Understand the reference models and network protocols
- Describe data transmission media and data link layer.
- Understand the network layer design issues and Network layer Protocols.
- Evaluate transport layer services and its protocols.
- Understand application layer protocols and their uses

UNIT I: Introduction

Types of Computer Networks, Reference Models- The OSI Reference Model, The TCP/IP Reference Model, A Critique of the OSI Model and Protocols, A Critique of the TCP/IP Reference Model. History of Internet.

UNITII: The Data Link Layer

Transmission Media, Guided and Un-guided media, Data Link Layer Design Issues, Services Provided to the Network Layer, Error detecting and Error Correcting codes, Elementary Data Link Protocols, Sliding Window Protocols, HDLC, PPP. Multiple Access Protocols Wired Lans: Ethernet, Fast Ethernet, Gigabit Ethernet

UNITIII: The Network Layer

Network LayerDesign Issues, Routing Algorithms, Congestion, Congestion control algorithms. The Network Layer in the Internet, The IP Version 4 Protocol, IP Addresses- Classful, CIDR, NAT, IP Version 6 Protocol, Transition from IPV4 to IPV6

UNITIV: The Transport Layer

The Transport Layer Services, Transport Layer Protocols: UDP, TCP and SCTP

UNITV: The Application Layer

The World Wide Web, HTTP, Domain Name Space, Remote Loging, Electronic Mail and File Transfer

Textbooks:

1. “Computer Networks”, Andrew S Tanenbaum, David J Wetherall, 5th Edition, Pearson
2. “Data Communications and Networking”, Behrouz A Forouzan, 4th Edition, Tata McGraw Hill Education

Reference Books:

1. “Data and Computer Communication”, William Stallings, Pearson
2. “TCP/IP Protocol Suite”, Behrouz Forouzan, McGraw Hill.

III Year I Semester	OPERATING SYSTEMS	L	T	P	C
		3	0	0	3

Course Objectives:

The main objectives of the course is to make student

- Understand the basic concepts and principles of operating systems, including process management, memory management, file systems, and Protection
- Make use of process scheduling algorithms and synchronization techniques to achieve better performance of a computer system.
- Illustrate different conditions for dead lock and their possible solutions.

UNIT-I:

Operating Systems Overview: Introduction, Operating system functions, Operating systems operations, Computing environments, Free and Open-Source Operating Systems System Structures: Operating System Services, User and Operating-System Interface, system calls, Types of System Calls, system programs, Operating system Design and Implementation, Operating system structure, Building and Booting an Operating System, Operating system debugging

UNIT-II:

Processes: Process Concept, Process scheduling, Operations on processes, Inter-process communication. Threads and Concurrency: Multi threading models, Thread libraries, Threading issues. CPU Scheduling: Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple processor scheduling.

UNIT–III:

Synchronization Tools: The Critical Section Problem, Peterson’s Solution, Mutex Locks, semaphores, Monitors, Classic problems of Synchronization. Deadlocks: system Model, Dead lock characterization, Methods for handling Dead locks,Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from Deadlock.

UNIT-IV:

Memory-Management Strategies: Introduction, Contiguous memory allocation, Paging,Structure of the Page Table, Swapping. Virtual Memory Management: Introduction ,Demand paging, Copy-on- write, Page replacement Allocation of frames Thrashing Storage Management: Overview of Mass Storage Structure,HDD Scheduling.

UNIT-V:

File System:File System Interface: File concept, Access methods, Directory Structure; File system Implementation: File-system structure, File-system Operations, Directory implementation, Allocation method, Free space management; File-System Internals: File-System Mounting, Partitions

And Mounting,File Sharing. Protection:Goals of protection, Principles of protection, Protection Rings,Domain of protection, Access matrix

TextBooks:

1. OperatingSystem Concepts,SilberschatzA,GalvinPB,GagneG,10th Edition,Wiley, 2018.
2. ModernOperatingSystems,TanenbaumAS,4th Edition,Pearson ,2016.

Reference Books:

1. Operating Systems -Internals and Design Principles,Stallings W, 9th edition, Pearson, 2018
2. OperatingSystems:A ConceptBasedApproach,D.MDhamdhere,3rd Edition, McGraw-Hill, 2013

OnlineLearningResources:

1. https://nptel.ac.in/courses/106/106/106106144/
2. http://peterindia.net/OperatingSystems.html

III Year I Semester	**OBJECT ORIENTED ANALYSIS AND DESIGN (PROFESSIONAL ELECTIVE-I)**	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

Course Objectives: The main objective is the students to

- Become familiar withal phases of OOAD.
- Master the main features of the UML.
- Master the main concepts of Object Technologies and how to apply them at work and develop the ability to analyze and solve challenging problem in various domains.
- Learn the Object design Principles and understand how to apply them towards Implementation.

UNIT-I:

Introduction: The Structure of Complex systems, The Inherent Complexity of Software, Attributes of Complex System, Organized and Disorganized Complexity, Bringing Order to Chaos, Designing Complex Systems. **Case Study:** System Architecture: Satellite-Based Navigation

UNIT-II:

Introduction to UML: Importance of modeling, principles of modeling, object- oriented modeling, conceptual model of the UML, Architecture, and Software Development Life Cycle. **Basic Structural Modeling:** Classes, Relationships, common Mechanisms, and diagrams. **Case Study:** Control System: Traffic Management.

UNIT-III:

Class &Object Diagrams: Terms, concepts, modeling techniques for Class & Object Diagrams.

Advanced Structural Modeling: Advanced classes, advanced relationships, Interfaces, Types and Roles, Packages. **Case Study:** AI: Cryptanalysis.

UNITIV:

Basic Behavioral Modeling-I: Interactions, Interaction diagrams Use cases, Use case Diagrams, Activity Diagrams. **Case Study:** Web Application: Vacation Tracking System

UNITV:

Advanced Behavioral Modeling: Events and signals, state machines, processes and Threads, time and space, state chart diagrams. **Architectural Modeling:** Component, Deployment, Component diagrams and Deployment diagrams **Case Study:** Weather Forecasting.

Text Books:

1. Grady BOOCH, Robert A. Maksim chuk, Michael W.ENGLE,BobbiJ.Young,Jim Conallen,

Kellia Houston, “Object- Oriented Analysis and Design with Applications”, 3rd edition,2013,PEARSON.

2. Grady Booch,James Rumbaugh,Ivar Jacobson:The Unified Modeling Language User Guide, Pearson Education.

ReferenceBooks:

1. Meilir Page-Jones:Fundamentals of Object-Oriented Designin UML,Pearson Education.
2. Pascal Roques:ModelingS of tware SystemsUsingUML2,WILEY- Dreamtech India Pvt.Ltd.
3. Atul Kahate:Object Oriented Analysis &Design, TheMcGraw-Hill Companies. Appling UML and Patterns:An introduction to Object– Oriented Analysis and Design and Unified Process, CraigLarman,Pearson Education.

III Year I Semester	Automata Theory and Compiler Design (Professional Elective-I)	L	T	P	C
		3	0	0	3

COURSE OUTCOMES: After completion of this course

- Understand and apply formal language theory.
- Design and implement parsers.
- Understand the phases of a compiler.
- Apply semantic analysis and error handling.
- Optimize intermediate and target code.

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

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

UNIT – III:

Push Down Automata: Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA's and CFG's, Acceptance by final state Turing Machines: Introduction to Turing Machine, Formal Description, Instantaneous description, The language of a Turing machine Undecidability: Undecidability, A Language that is Not Recursively Enumerable, An Undecidable Problem That is RE, Undecidable Problems about Turing Machines

UNIT - IV Introduction: The structure of a compiler, Lexical Analysis: The Role of the Lexical Analyzer, Input Buffering, Recognition of Tokens, The Lexical- Analyzer Generator Lex, Syntax Analysis: Introduction, Context-Free Grammars, Writing a Grammar, Top-Down Parsing, Bottom-Up Parsing, Introduction to LR Parsing: Simple LR, More Powerful LR Parsers R18 B.Tech. CS&D Syllabus JNTU Hyderabad

UNIT - V Syntax-Directed Translation: Syntax-Directed Definitions, Evaluation Orders for SDD's, SyntaxDirected Translation Schemes, Implementing L-Attributed SDD's. Intermediate-Code Generation: Variants of Syntax Trees, Three-Address Code Run-Time Environments: Stack Allocation of Space, Access to Nonlocal Data on the Stack, Heap Management

Text Books:

1. Introduction to Automata Theory, Languages, and Computation, 3nd Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2. Compilers: Principles, Techniques and Tools, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman, 2nd Edition, Pearson.
3. Theory of Computer Science – Automata languages and computation, Mishra and Chandrashekaran, 2nd Edition, PHI.

Reference Books:

1. Introduction to Formal languages Automata Theory and Computation, Kamala Krithivasan, Rama R, Pearson.
2. Introduction to Languages and The Theory of Computation, John C Martin, TMH.
3. lex &yacc – John R. Levine, Tony Mason, Doug Brown, O’reilly
4. Compiler Construction, Kenneth C. Louden, Thomson. Course Technology.

III Year I Semester	Soft Computing (Professional Elective-I)	L	T	P	C
		3	0	0	3

Course Objectives:

To introduce the concepts in Soft Computing such as Artificial Neural Networks, Fuzzylogic-based systems, genetic algorithm-based systems and their hybrids.

Course Outcomes:

The students will be able to

- Learn soft computing techniques and their applications.
- Analyze various neural network architectures.
- Define the fuzzy systems.
- Understand the genetic algorithm concepts and their applications.
- Identify and select a suitable Soft Computing technology to solve the problem; construct a solution and implement a Soft Computing solution

UNIT-I:

Introduction to Soft Computing, Artificial neural networks, biological neurons, Basic models of artificial neural networks, Connections, Learning, Activation Functions, McCulloch and Pitts Neuron, Hebb network.

UNIT-II:

Perceptron networks, learning rule, Training and testing algorithm, Adaptive Linear Neuron, Back propagation Network, Architecture, Training algorithm

UNIT-III:

Fuzzy logic, fuzzy sets, properties, operations on fuzzy sets, fuzzy relations, operations on fuzzy relations, Fuzzy membership functions, fuzzification, Methods of membership, value assignments, intuition, inference, rank ordering, Lambda –Cuts for fuzzy sets, Defuzzification methods

UNIT-IV:

Truth values and Tables in Fuzzy Logic, Fuzzy propositions, Formation of fuzzy rules, Decomposition of rules, Aggregation of rules, Fuzzy Inference Systems, Mamdani and Sugeno types, Neuro-fuzzy hybrid systems, characteristics, classification

UNIT-V:

Introduction to genetic algorithm,operators in genetic algorithm,coding, selection, , crossover mutation,stopping condition for genetic algorithm flow, Genetic-neuro hybrid systems, Genetic Fuzzy rule-based system

Text Books:

1. S.N.Sivanandam and S.N.Deepa,Principles of soft computing–JohnWiley & Sons,2007.
2. Timothy J.Ross,Fuzzy Logic with engineering applications, John Wiley &Sons, 2016.

ReferenceBooks:

1. N.K.Sinha and M.M.Gupta,Soft Computing& Intelligent Systems:Theory &Applications-Academic Press /Elsevier. 2009.
2. Simon Haykin,Neural Network-AComprehensive Foundation-Prentice Hall International, Inc.1998
3. R. Eberhart and Y. Shi, Computational Intelligence:
4. Concepts to Implementation, Morgan Kaufman/Elsevier, 2007.
5. DriankovD.,Hellendoorn H.and Reinfrank M.An Introduction toFuzzy Control Narosa Pub., 2001.
6. BartKosko,NeuralNetworkandFuzzySystems-PrenticeHall,Inc.,Englewood Cliffs, 1992
7. GoldbergD.E.,Genetic Algorithms in Search,Optimization,and Machine LearningAddison Wesley, 1989

III Year I Semester	Internet of Things (Professional Elective-I)	L	T	P	C
		3	0	0	3

Course Objectives:

- Vision and Introduction to Internet of Things (IoT).
- Understand IoT Market perspective.
- Data and Knowledge Management and use of Devices in IoT Technology.
- Understand State of the Art – IoT Architecture.
- Understand Real World IoT Design Constraints, Industrial Automation and Commercial.

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

- Explain in a concise manner how the general Internet as well as Internet of Things work.
- Understand constraints and opportunities of wireless and mobile networks for Internet of Things.
- Use basic sensing and measurement and tools to determine the real-time performance of network of devices.
- Develop prototype models for various applications using IoT technology.

UNIT-I:

The Internet of Things: An Overview of Internet of things, Internet of Things Technology, behind IoTs Sources of the IoTs, M2M Communication, Examples of IoTs, Design Principles For Connected Devices Internet Connectivity Principles, Internet connectivity, Application Layer Protocols: HTTP, HTTPS, FTP, Telnet.

UNIT-II:

Business Models for Business Processes in the Internet of Things, IoT/M2M systems LAYERS AND designs standardizations, Modified OSI Stack for the IoT/M2M Systems, ETSI M2M domains and High- level capabilities, Communication Technologies, Data Enrichment and Consolidation and Device Management Gateway Ease of designing and affordability

UNIT-III:

Design Principles for the Web Connectivity for connected-Devices, Web Communication protocols for Connected Devices, Message Communication protocols for Connected Devices, Web Connectivity for connected-Devices.

UNIT-IV:

Data Acquiring, Organizing and Analytics in IoT/M2M, Applications /Services /Business Processes, IOT/M2M Data Acquiring and Storage, Business Models for Business Processes in the Internet of Things, Organizing Data, Transactions, Business Processes, Integration and Enterprise Systems.

UNIT-V:

Data Collection, Storage and Computing Using a Cloud Platform for IoT/M2M Applications/Services, Data Collection, Storage and Computing Using cloud platform Everything as a service and Cloud Service Models, IOT cloud-based services using the Xively (Pachube/COSM), Nimbits and other platforms Sensor, Participatory Sensing, Actuator, Radio Frequency Identification, and Wireless, Sensor Network Technology, Sensors Technology, Sensing the World.

Text Books:

1. Internet of Things: Architecture, Design Principles and Applications, Rajkamal, McGraw Hill Higher Education
2. Internet of Things, A.Bahgya and V.Madisetti, University Press, 2015

Reference Books:

1. Designing the Internet of Things, Adrian McEwen and Hakim Cassimally,WileyGetting Started with the Internet of Things, Cuno Pfister, Oreilly

III Year I Semester	Deep Learning Lab	L	T	P	C
		0	0	3	1.5

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

- Implement deep neural networks to solve real world problems
- Choose appropriate pre-trained model to solve real time problem
- Interpret the result soft different deep learning models

Software Packages required:

- Keras
- Tensorflow
- PyTorch

List of Experiments:

1. Implement multi-layer perceptron algorithm for MNISTH and written Digit Classification.
2. Design neural network for classifying movie reviews(BinaryClassification) using IMDB dataset.
3. Design a neural Network for classifying news wires (Multiclassclassification) using Reuters dataset.
4. Design neural network for predicting house prices using Boston Housing Price dataset.
5. Build a Convolution Neural Network for MNISTH and written Digit Classification.
6. Build a Convolution Neural Network for simple image (dogs and Cats) Classification
7. Use a pre-trained convolution neural network (VGG16)for image classification.
8. Implement one hot encoding of words or characters.
9. Implement word embeddings for IMDB dataset.
10. Implement a Recurrent Neural Network for IMDB movie review classification problem.

Text Books:

1. Reza Zadeh and Bharath Ramsundar,"Tensorflow for DeepLearning", O'Reilly publishers, 2018

References:

1. https://github.com/fchollet/deep-learning-with-python-notebooks

III Year I Semester	Computer Networks Lab	L	T	P	C
		0	0	3	1.5

List of Experiments:

1. Study of Network devices in detail and connect the computers in Local Area Network.
2. Write a Program to implement the data link layer farming methods such as i) Character stuffing ii) bit stuffing.
3. Write a Program to implement data link layer farming method checksum.
4. Write a program for Hamming Code generation for error detection and correction.
5. Write a Program to implement on a data set of characters the three CRC polynomials – CRC 12, CRC 16 and CRC CCIP.
6. Write a Program to implement Sliding window protocol for Goback N.
7. Write a Program to implement Sliding window protocol for Selective repeat.
8. Write a Program to implement Stop and Wait Protocol.
9. Write a program for congestion control using leaky bucket algorithm
10. Write a Program to implement Dijkstra‘s algorithm to compute the Shortest path through a graph.
11. Write a Program to implement Distance vector routing algorithm by obtaining routing table at each node (Take an example subnet graph with weights indicating delay between nodes).
12. Write a Program to implement Broadcast tree by taking subnet of hosts.
13. Wireshark
    i. Packet Capture Using Wire shark
    ii. Starting Wire shark
    iii. Viewing Captured Traffic
    iv. Analysis and Statistics & Filters.
14. How to run Nmap scan
15. Operating System Detection using Nmap
16. 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.

III Year I Semester	Full Stack Frontend Development – Module – II (Skill Oriented Course)	L	T	P	C
		0	1	2	2

Course Objectives:

The main objectives of the course are to

- Make use of router, template engine and authentication using sessions to develop application in ExpressJS.
- Build a single page application using RESTful APIs in ExpressJS
- Apply router and hooksin designing ReactJS application

- Make use of MongoDB queries to perform CRUD operations on document database

Experiments covering the Topics:

- ExpressJS – Routing, HTTP Methods, Middleware, Templating, Form Data
- ExpressJS – Cookies, Sessions, Authentication, Database, RESTful APIs
- ReactJS – Render HTML, JSX, Components – function & Class, Props and States, Styles, Respond to Events
- ReactJS – Conditional Rendering, Rendering Lists, React Forms, React Router, Updating the Screen
- ReactJS – Hooks, Sharing data between Components, Applications – To-do list and Quiz
- MongoDB – Installation, Configuration, CRUD operations, Databases, Collections and Records

Sample Experiments:

1. ExpressJS – Routing, HTTP Methods, Middleware.

a. Write a program to define a route, Handling Routes, Route Parameters, Query Parameters and URL building.

b. Write a program to accept data, retrieve data and delete a specified resource using http methods.

c. Write a program to show the working of middleware.

2. ExpressJS – Templating, Form Data

a. Write a program using templating engine.

b. Write a program to work with form data.

3. ExpressJS – Cookies, Sessions, Authentication

a. Write a program for session management using cookies and sessions.

b. Write a program for user authentication.

4. ExpressJS – Database, RESTful APIs

a. Write a program to connect MongoDB database using Mangoose and perform CRUD operations.

b. Write a program to develop a single page application using RESTful APIs.

5. ReactJS – Render HTML, JSX, Components – function & Class

a. Write a program to render HTML to a web page.

b. Write a program for writing markup with JSX.

c. Write a program for creating and nesting components (function and class).

d.

6. ReactJS – Props and States, Styles, Respond to Events

a. Write a program to work with props and states.

b. Write a program to add styles (CSS & Sass Styling) and display data.

c. Write a program for responding to events.

7. ReactJS – Conditional Rendering, Rendering Lists, React Forms

a. Write a program for conditional rendering.

b. Write a program for rendering lists.

c. Write a program for working with different form fields using react forms.

8. ReactJS – React Router, Updating the Screen

a. Write a program for routing to different pages using react router.

b. Write a program for updating the screen.

9. ReactJS – Hooks, Sharing data between Components

a. Write a program to understand the importance of using hooks.

b. Write a program for sharing data between components.

10. MongoDB – Installation, Configuration, CRUD operations

a. Install MongoDB and configure ATLAS

b. Write MongoDB queries to perform CRUD operations on document using insert(), find(), update(), remove()

11. MongoDB – Databases, Collections and Records

a. Write MongoDB queries to Create and drop databases and collections.

b. Write MongoDB queries to work with records using find(), limit(), sort(), createIndex(), aggregate().

12. Augmented Programs: (Any 2 must be completed)

a. Design a to-do list application using NodeJS and ExpressJS.

b. Design a Quiz app using ReactJS.

c. Complete the MongoDB certification from MongoDB University website.

Text Books:

1. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node, Vasan Subramanian, 2nd edition, APress, O’Reilly.
2. Node.Js in Action, Mike Cantelon, Mark Harter, T.J. Holowaychuk, Nathan Rajlich,Manning Publications. (Chapters 1-11)
3. React Quickly, AzatMardan,Manning Publications(Chapters 1-8,12-14)

Web Links:

1. ExpressJS - https://www.tutorialspoint.com/expressjs
2. ReactJS - https://www.w3schools.com/REACT (and) https://react.dev/learn#
3. MongoDB -https://learn.mongodb.com/learning-paths/introduction-to-mongodb

III Year I Semester	Tinkering Lab (UI Design Using Flutter)	L	T	P	C
		0	0	2	1

Course Objectives:

- Learns to Implement Flutter Widgets and Layouts
- Understands Responsive UI Design and with Navigation in Flutter
- Knowledge on Wedges and customize widgets for specific UI elements, Themes
- Understand to include animation apart from fetching data

List of Experiments:

Students need to implement the following experiments

17. a) Install Flutter and Dart SDK.
    b) Write a simple Dart program to understand the language basics.

18. a) Explore various Flutter widgets (Text, Image, Container, etc.).
    b) Implement different layout structures using Row, Column, and Stack widgets.

19. a) Design a responsive UI that adapts to different screen sizes.
    b) Implement media queries and breakpoints for responsiveness.

20. a) Set up navigation between different screens using Navigator.
    b) Implement navigation with named routes.

21. a) Learn about stateful and stateless widgets.
    b) Implement state management using set State and Provider.

22. a) Create custom widgets for specific UI elements.
    b) Apply styling using themes and custom styles.

23. a) Design a form with various input fields.
    b) Implement form validation and error handling.

24. a) Add animations to UI elements using Flutter's animation framework.
    b) Experiment with different types of animations (fade, slide, etc.).

25. a) Fetch data from a REST API.
    b) Display the fetched data in a meaningful way in the UI.

26. a) Write unit tests for UI components.
    b) Use Flutter's debugging tools to identify and fix issues..

Text Book:

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

2. Rap Payne, Beginning App Development with Flutter: Create Cross-Platform Mobile Apps 1stEdition, Apres