ANNEXURE: 1 MODULE WISE COURSE CONTENT AND OUTCOME

SL.NO	MODULE NAME	MODULE CONTENT	MODULE LEARNING OUTCOME	DURATION (HRS)
1	HTML - Structure and Basics	1. HTML Fundamentals ● Document structure, tags, and attributes ● Semantic elements (<header>, <main>, <article>, etc.) ● Forms, tables, and multimedia embedding (images, videos) 2. Accessibility and SEO ● ARIA roles and attributes ● Meta tags for SEO	Participants will learn to structure web pages using HTML, semantic elements, and attributes, create forms, tables, and embed multimedia, while ensuring accessibility through ARIA roles and improving SEO with meta tags.	4
2	CSS - Styling and Layouts	1. CSS Basics ● Selectors, properties, and specificity ● Box model, colors, typography 2. Layouts and Responsive Design ● Flexbox and Grid systems ● Media queries for mobile-first design 3. Animations and Transitions ● Key frames and transforms for smooth effects	participants will gain a strong understanding of CSS basics, including selectors, properties, and specificity, along with the box model, color theory, and typography. They will learn how to create responsive layouts using Flexbox and CSS Grid, and implement mobile-first design with media queries. Additionally, participants will master CSS animations and transitions, using keyframes and transforms to create smooth, interactive effects, enhancing user experience and interface design.	5

3	Bootstrap - Rapid UI Development	1. Bootstrap Basics ● Setup and grid system ● Components: Navbar, buttons, cards, modals 2. Customizing Bootstrap ● Overriding default styles with custom CSS 3. Advanced Features ● Carousel for dynamic image sliders ● Form validation with Bootstrap utilities	Participants will learn to quickly build responsive and modern web interfaces using **Bootstrap**, including setting up the grid system and utilizing components like navbars, buttons, cards, and modals. They will also gain skills in customizing Bootstrap with custom CSS and explore advanced features like dynamic image sliders with carousels and form validation using Bootstrap utilities, enabling them to create functional and polished web applications efficiently.	4
4	JavaScript - Interactivity	1. JavaScript Essentials ● Variables, operators, and data types ● Functions, events, and DOM manipulation ● Control structures: Loops and conditionals 2. Modern JavaScript (ES6+) ● Arrow functions, destructuring, template literals ● Fetch API for HTTP requests 3. Dynamic Interactions ● Form validation, modal controls, and event handling	Participants will gain a solid understanding of **JavaScript fundamentals**, including variables, operators, data types, functions, and events, along with essential control structures like loops and conditionals. They will master modern JavaScript features such as arrow functions, destructuring, template literals, and the Fetch API for making HTTP requests. Additionally, participants will learn how to create dynamic interactions on web pages, including form validation, modal controls, and event handling, enhancing user experience and functionality.	8

| 5 | ReactJS - Front-End Framework | 1. React Basics
● Components, JSX, and props
● State management and lifecycle methods
2. React Router
● Navigation and dynamic routing
3. React Hooks
● useState, useEffect, and custom hooks

Project:

● Build a small dynamic app (e.g., a to-do list or weather app) | Participants will learn the fundamentals of **React**, including creating components, using JSX, and managing data with props and state. They will understand React's lifecycle methods and how to implement **React Router** for navigation and dynamic routing. Additionally, participants will master **React Hooks** like **useState, useEffect**, and custom hooks to manage state and side effects. Finally, they will apply their knowledge by building a small dynamic application, such as a to-do list or weather app, to demonstrate their skills in real-world project development. | 8 |
| 6 | Node.js and Express.js - Back-End Framework | 1. Introduction to Node.js
● Core modules, npm, and asynchronous programming
● Creating a simple server with Node.js
2. Express.js Basics
● Setting up an Express app
● Middleware, routing, and RESTful APIs
3. Connecting Front-End with Back-End
● Sending data between React and Node.js via APIs | Participants will gain a foundational understanding of **Node.js**, including core modules, npm, and asynchronous programming techniques. They will learn to create simple servers using Node.js and set up **Express.js** apps for building web servers and RESTful APIs. Key concepts such as middleware and routing will be covered, alongside techniques for connecting the **front-end** (React) to the **back-end** (Node.js) through API communication, enabling data exchange and full-stack development. | 6 |

| 7 | MongoDB - Database | 1. MongoDB Basics
● Database creation, collections, and CRUD operations
● Connecting MongoDB with Node.js using Mongoose
2. Schema and Model Design
● Defining schemas and working with models
3. Project Integration
● Using MongoDB to store and retrieve data for the app | Participants will learn the basics of **MongoDB**, including database creation, collections, and performing CRUD operations. They will explore how to connect MongoDB to **Node.js** using **Mongoose**, and understand how to define schemas and work with models for efficient data management. By integrating MongoDB into a project, participants will gain practical experience in storing and retrieving data, enabling them to build dynamic, data-driven applications. | 5 |
| 8 | Mini Project (End-to-End) | Project Title: Task Management System

1. Frontend (ReactJS):
● User interface for task creation and management
● Dynamic updates and routing for pages
2. Backend (Node.js + Express.js):
● RESTful API to handle task data (CRUD operations)
3. Database (MongoDB):
● Store and retrieve user tasks | Participants will apply their skills by building a **Task Management System**, creating a user interface in **ReactJS** for task creation and management with dynamic updates and routing. On the back-end, they will develop a **RESTful API** using **Node.js** and **Express.js** to handle task data with CRUD operations. The project will integrate **MongoDB** to store and retrieve user tasks, providing hands-on experience in full-stack development and end-to-end application building. | 5 |

ANNEXURE : 2 Industry Use Cases/Final Projects

LEARNING OUTCOME	ASSESSMENT CRITERIA	PERFORMANCE CRITERIA	USECASES
Master HTML fundamentals	Use of HTML structure, semantic elements, and embedding multimedia.	Proper HTML tag usage, semantic structure for accessibility, and embedding images/videos correctly.	Building a portfolio webpage with semantic HTML for accessibility and SEO optimization
Implement CSS for layout and styling	Application of CSS selectors, box model, Flexbox, Grid, and media queries.	Correct layout design with Flexbox/Grid, responsive design via media queries, and smooth transitions.	Designing a responsive product page layout that adapts across mobile and desktop views.
Develop dynamic front-end features with JavaScript and ReactJS	Using JavaScript for event handling, DOM manipulation, and React components.	Implementing dynamic interactions with JavaScript, managing state with React, and handling APIs.	Creating an interactive to-do list app with ReactJS, managing task data dynamically and handling form validation.

LIST OF FINAL PROJECTS	
SL.NO	**FINAL PROJECT**
1	TaskMaster: A Task Management Application
2	QuickShop: An E-commerce Store
3	Eventify: A Personal Event Planner
4	FitTrack: A Workout and Nutrition Tracker
5	RecipeHub: A Recipe Sharing Platform
6	TaskFlow: A Project Management Tool
7	StudyBuddy: A Study Plan and Resource Sharing App
8	BookNook: A Book Review and Recommendation System
9	MovieMate: A Movie Tracker and Recommendation App
10	TravelBuddies: A Trip Planning and Sharing Platform
11	HealthMate: A Personal Health Dashboard
12	ChatFlow: A Real-time Messaging Application
13	LocalConnect: A Community Forum Platform
14	WeatherWatch: A Weather Forecast App
15	MyNotes: A Note-taking and Organization App
16	CleanIt: A Cleaning Service Scheduling App
17	JobLink: A Job Listing and Application System
18	TaskForge: A Collaborative Task Scheduling App
19	BudgetEase: A Personal Finance and Budget Tracker
20	HomeConnect: A Smart Home Automation System

ANNEXURE 3 – COURSE ASSESSMENT

COURSE ASSESSMENT RUBRICS (TOTAL MARKS: 70)				
ASSESSMENT CRITERIA	**DESCRIBE THE CRITERIA OF THE BELOW CATEGORY PERFORMANCE**			**TOTAL MARKS**
	FAIR	**GOOD**	**EXCELLENT**	
Assessment	50% - 64%	65% - 79%	80% - 100%	25
Milestone	50% - 64%	65% - 79%	80% - 100%	25
C-LAB	50% - 64%	65% - 79%	80% - 100%	20

Category	Assessment Criteria	Performance Levels	Weightage (Marks)
Practical Skills Proficiency	Demonstrates ability to perform job-specific tasks effectively, using relevant tools, techniques, or methodologies (e.g., Tally for accounting, consignment tracking).	Fair, Good, Excellent	20
Technical Knowledge Application	Applies theoretical concepts to practical scenarios with accuracy and relevance (e.g., compliance with GST laws, financial planning, or logistics protocols).	Fair, Good, Excellent	15
Project Execution	Completes assigned projects or use cases demonstrating innovation, thoroughness, and skill application relevant to industry standards.	Fair, Good, Excellent	25
Communication and Reporting	Clearly presents findings, solutions, or project outcomes using professional communication and documentation standards (e.g., reports, presentations).	Fair, Good, Excellent	10