

## **Full Stack Developer - MERN Course Content**

### **Section 1: HTML 5**

- Introduction - Web
- What are the different Types of Web Apps overview?
- Introduction to HTML
- Define HTML Structure
- Difference between Tag vs Element
- Difference between Semantic vs Non-semantic elements
- Difference between Block level elements vs Inline elements
- HTML Elements
- HTML Forms & its Attributes
- HTML Input Elements
- HTML Global Attributes
- HTML Element Specific Attributes

### **Section 2: CSS & Bootstrap**

- What is CSS?
- Understanding the CSS Syntax.
- CSS Selectors
- How To Add CSS in HTML
- CSS Colors & Backgrounds
- CSS Borders, Margins, Padding, Height and Width
- Responsive Web Design – Media queries
- What is Bootstrap?
- Get Start with Bootstrap

- Containers
- Grid System
- Structure of a Bootstrap Grid
- Bootstrap Colors
- Bootstrap Tables
- Bootstrap Jumbotron
- Bootstrap Alerts
- Bootstrap Buttons

### **Section 3: JavaScript**

- Introduction – Javascript
- JavaScript Events
- JavaScript Functions
- Inbuilt methods in JS
- Arrays in JavaScript
- Objects in JavaScript
- Conditional and loops in JavaScript
- HTML DOM Manipulation

### **Section 4: ES6 JavaScript**

- History of JavaScript
- Features
- let & const and its example
- Arrow Functions
- Alternative
- Tips and Arrow Functions

- Exports and Imports
- Tips for exports and imports
- Classes
- Classes example
- Inheritance
- Spread and rest Parameter
- Destructing

## **Section 5: Getting Started with React**

- What is ReactJS?
- Installation or Setup
- Create React App
- Advantages Of React JS
- Workflow Of React JS
- Node Setup
- How to use NPM and its purpose
- ES6 Introduction
- JS vs JSX vs TS vs TSX

## **Section 6: Components**

- Creating Components
- Basic Component
- Nesting Components
- Higher order components

## **Section 7: OVERVIEW OF JSX**

- Introduction of Virtual DOM.
- Difference between JS and JSX.
- Why use JSX?
- JSX Attributes
- JSX Styling

## **Section 8: Props in React**

- Introduction
- Default props
- PropTypes
- Passing down props using spread operator
- Props.children and component composition
- Detecting the type of Children components
- Props Drilling

## **Section 9: State in React**

- Basic State
- Common Antipattern
- setState()
- State, Events And Managed Controls

## **Section 10: React Component Lifecycle**

- Component willmount
- Component UnMount
- Component Update

- Lifecycle method call in different states
- React Component Container

### **Section 11: Handling Events**

- Event handling in React
- Binding event handlers
- Arrow functions vs. regular functions
- Primary Events
- Sharing the events between components

### **Section 12: Communicate Between Components**

- Child to Parent Components
- Parent to Child Components
- Not-related Components

### **Section 13: Conditional Rendering**

- Conditional rendering with if statements
- Conditional Rendering using logical && in JSX
- Conditional Rendering using ternary operator
- Preventing Component from Rendering

### **Section 14: Lists and Keys**

- Steps to Create and Traverse React JS Lists
- Rendering lists inside Components
- Key in React List

### **Section 15: React JS Keys**

- What is a key in React ?
- Assigning keys to the list
- Difference between keys and props in React

## **Section 16: React Hooks**

- useState()
- useEffect()
- useContext()
- useRef()
- useReducer()
- Custom React Hooks
- useDispatch(), useSelector, useMemo, useEffect()

## **Section 17: Styling in React**

- CSS in React
- Different approaches for styling (CSS, CSS-in-JS, CSS Modules)
- Inline styles
- Styling Libraries
- Popular CSS frameworks (e.g., Bootstrap, Material-UI)

## **Section 18: Router**

- Introduction to React Router
- Setting up React Router
- Creating routes
- Navigating with React Router
- Using Link and NavLink

- Nested Routes and Dynamic Routing
- Nested routes
- Passing parameters to routes
- Query String, accessing current URL

### **Section 19: React JS forms**

- React Forms
- Lists of Form components
- Control Input elements.
- Controlled and Uncontrolled form components
- Adding Forms in React
- Handling React Forms
- Submitting React Forms
- Multiple Input Fields
- React JS Form validations
- Custom Validations

### **Section 20: State Management with Redux**

- Introduction to Redux
- Understanding the need for state management
- Basic concepts: actions, reducers, store
- Setting Up Redux
- Installing Redux and setting up a store
- Creating actions and reducers
- Connecting React with Redux
- Using connect to connect components to the store

- Dispatching actions

## **Section 21: Asynchronous Programming and API Integration**

- AJAX and Fetch API
- Making HTTP requests in React
- Fetching data from an API
- Async/Await and Promises

## **Section 22: Handling errors in React application**

- Error Handling and Debugging
- Debugging React apps
- Performance Optimization
- Memoization
- React.memo and PureComponent

## **Section 23: React JS Virtual DOM**

- What is DOM ?
- Disadvantages of real DOM
- Virtual DOM
- How does virtual DOM actually make things faster?
- How virtual DOM Helps React?
- Virtual DOM Key Concepts
- Differences between Virtual DOM and Real DOM

## **Section 24: Unit Testing Overview**

## **Section 25: Lazy Loading**



## **Section 26: Code Splitting**

## **Section 27: Server-Side Rendering**

## **Section 28: Micro FrontEnd Overview**

## **Section 29: Deploying a React application**

## **Section 30: Node JS AND Express JS**

- What is Node JS?
- NPM Overview
- How to install NPMs
- Introduction about package.json
- What are the core Modules Available?
- How to create & run a Node Server
- How to create Local Module?
- Node Routing in detail
- Error Handling in Node JS

## **Section 31: Working with top NPMs**

- Nodemon
- JWT
- Body Parser
- HTTP Request & Response – GET, POST, PUT & DELETE
- How to handle Headers?
- How to receive form data?
- Authorization & Authentication

## **Section 32: Express JS**

- Express JS Overview
- Routing in Express JS
- How to Run Express on top of Node JS?
- What is Middleware?
- How Middleware works?
- How to create RESTful API?
- Debugging

## **Section 33: MongoDB**

- MongoDB Overview
- How to use MongoDB in Cloud?
- Cluster in MongoDB
- Create database
- Create Users
- Create table
- What is Collection & Document?
- Insert & Insert many Document
- Select & Select One Document
- Delete & Delete One Document.
- Update & Update one Document

## **Section 34: Real Time Projects**

- 5+ Real-time projects (Full stack coverage)
- Setup Git for local repository

- Create a GitHub account
- Sync the code base with GitHub repo
- Firebase Hosting Overview
- Setting up Firebase in local
- Host your project in Firebase

### **Section 35: Placements**

- Build your Professional Resume
- Update LinkedIn Profile
- Interview based Training – GD, Tech round, HR panel
- Minimum 5 Mock interviews before Real interview
- 100% Placement Guaranteed
- Join your Dream Job

### **Free Complementary Courses – 25hrs**

#### **AWS – Cloud Computing:**

#### **Section 1: Cloud Computing Introduction**

- History of Cloud Computing
- What is Cloud Computing?
- Why Cloud Computing?
- Advantages of Cloud Computing

#### **Section 2: Cloud Computing Service Models**

- What is Software as a Service (SaaS)?
- Platform as a Service (PaaS)?

- Infrastructure as a Service (IaaS)?

### **Section 3: Introduction to AWS Architecture**

- Key Considerations for Web Hosting in AWS

### **Section 4: AWS – Management Console & Account**

- How to Access AWS?
- AWS Identity & Access Management (IAM).
- Creating Free Tier Account in AWS
- Introduction of AWS Management Console / CLI / SDK

### **Section 5: Compute Services in AWS**

- AWS EC2
- AWS Lambda

### **Section 6: Storage and Content Delivery Services**

- AWS S3
- AWS EBS

### **Section 7: Database Services**

- Amazon RDS
- AWS DynamoDB

### **Section 8: Networking Services in AWS**

- VPC AWS
- Route 53

## **Agile Scrum**

### **Section 1: Definition of SCRUM**

### **Section 2: Agile Manifesto**

- Scrum Master
- Product Owner

### **Section 3: Scrum Roles and Responsibilities**

- Development Team

### **Section 4: Typical Sprint Phases**

- Product Backlog
- Sprint Planning
- Sprint Backlog
- Sprint
- Daily Scrum
- Sprint Review
- Sprint Retrospective