**Front-End With ReactJS**

**(LIVE BATCH-2403)**

**Week 1: HTML, CSS, JavaScript Revision & Introduction to React.js**

* **HTML Revision**
  + Basic HTML structure
  + Common HTML tags (div, span, header, footer, etc.)
  + Forms and input elements
  + Semantic HTML
* **CSS Revision**
  + CSS basics (selectors, properties, values)
  + Box model, Flexbox, Grid layout
  + CSS positioning and z-index
  + Responsive design principles and media queries
* **JavaScript Revision**
  + Basic syntax and data types
  + Functions, loops, and conditional statements
  + DOM manipulation
  + ES6 features:
    - let, const
    - Arrow functions
    - Template literals
    - Destructuring (Object and Array)
    - Spread and Rest operators
    - Modules (import/export)
    - Promises and async/await
* **Tailwind CSS Introduction**
  + Setting up Tailwind CSS
  + Basic utility classes
  + Responsive design with Tailwind CSS
* **Introduction to React**
  + What is React?
  + Setting up the development environment (Node.js, npm/yarn, create-react-app)
  + Understanding components, JSX, and virtual DOM
* **First React Component**
  + Creating and rendering components
  + Component structure and file organization
  + Basic component styling
* **Basic Project**
  + Build a simple web page using HTML, CSS (including Tailwind CSS), and JavaScript
  + Focus on layout, styling, and basic interactivity

**Week 2: React Components and State Management**

* **Props and State**
  + Passing data with props
  + Managing state in class components
* **Component Lifecycle**
  + Understanding component lifecycle methods
  + Mounting, updating, and unmounting phases
* **Event Handling**
  + Handling events in React
  + Binding event handlers
* **Conditional Rendering**
  + Rendering elements based on conditions
  + Inline if-else with logical && and ternary operator
* **Lists and Keys**
  + Rendering lists in React
  + Using keys to identify elements

**Week 3: Advanced React Concepts**

* **Forms in React**
  + Controlled vs. uncontrolled components
  + Handling form submissions and inputs
* **Lifting State Up**
  + Sharing state between components
  + Drilling props vs. context API
* **React Router**
  + Setting up React Router
  + Navigation between pages
  + Route parameters
* **Context API**
  + Context creation and usage
  + Using context to manage global state
* **Hooks Introduction**
  + Introduction to hooks
  + useState and useEffect hooks

**Week 4: Project Setup and Initial Development**

* **Custom Hooks**
  + Creating custom hooks
  + Using multiple hooks together
* **Project Planning**
  + Choosing a project topic
    - E-Learning Platform
    - T-shirt Store
    - Flight Booking
    - Room Booking
  + Setting up project structure
* **Initial Setup**
  + Creating the project repository
  + Setting up basic components and routing
* **Building Components**
  + Creating reusable components
  + Structuring and styling components
* **Managing State**
  + Setting up state management for the project
  + Using Context API or a state management library (Redux)

**Week 5: Integrating APIs and Advanced Features**

* **Connecting Components**
  + Integrating components with state management
  + Handling data flow and component communication
* **Fetching Data from APIs**
  + Using fetch and axios to make API calls
  + Handling asynchronous data in React
* **Handling API Responses**
  + Displaying fetched data
  + Error handling in API calls
* **Authentication**
  + Implementing login and registration forms
  + Using authentication tokens (JWT)
* **Advanced Routing**
  + Protecting routes with authentication
  + Nested routes and lazy loading

**Week 6: Project Completion and Deployment**

* **Finalizing Features**
  + Adding additional features (e.g., search, filtering)
  + Optimizing performance
* **Testing and Debugging**
  + Writing tests for components (Jest, React Testing Library)
  + Debugging common issues
* **Styling and UI Enhancements**
  + Using CSS-in-JS libraries (styled-components, Emotion)
  + Enhancing the user interface
* **Final Touches**
  + Finalizing project features
  + Code refactoring and cleanup
* **Deployment Preparation**
  + Preparing the project for deployment
  + Building the project for production

**Week 7: Project Presentations and Review**

* **Deployment**
  + Deploying to platforms like Netlify, Vercel, or GitHub Pages
  + Post-deployment checks and optimizations
* **Project Presentations**
  + Students present their projects
  + Code review and feedback
* **Advanced Topics** 
  + Introduction to TypeScript in React
  + Performance optimization techniques

**Upon Completion, Students Will:**

* **Understand Fundamental Web Technologies**
  + Gain a solid understanding of HTML, CSS, and JavaScript ES6+ features.
* **Develop Proficiency in React.js**
  + Build and manage components, handle state, and use React lifecycle methods.
  + Utilize hooks and the Context API for state management.
  + Implement routing with React Router.
* **Work with APIs and Authentication**
  + Make API calls and handle asynchronous data.
  + Implement authentication mechanisms using JWT.
* **Build and Deploy a Complete React Project**
  + Plan, develop, and deploy a full-fledged React application.
  + Apply best practices in code organization, styling, and optimization.
* **Improve Debugging and Testing Skills**
  + Write tests for React components and debug common issues.
* **Present and Review Code**
  + Present projects and receive feedback to enhance code quality and application performance.