React.js 4-Day

Day 1: React Basics

Topics Covered

1. **Introduction to React**
   - What is React?
   - Advantages of React
   - Understanding React's Component-Based Architecture
2. **Components**
   - Introduction to Functional Components
   - **Discussion: Class Components vs Functional Components**
     - Evolution from Class Components to Functional Components
     - Why Functional Components are Preferred Today
     - Key Differences (State, Lifecycle, and Hooks)
   - Writing Your First Functional Component
3. **Props and State**
   - Passing Data to Components Using Props
   - Managing Component Data Using State
   - Understanding the Difference Between Props and State
4. **Styling Basics**
   - Inline Styles in React
   - Using CSS and CSS Modules for Component-Level Styling
   - Dynamic Styles with Props and State
5. **Conditional Rendering**
   - Rendering Elements Conditionally Using `if`, Ternary Operators, and &&
6. **List Rendering**
   - Rendering Lists Using `map`
   - Using Keys in List Items

7. **Hands-On Practice**
   - Build a Dynamic User List with Add and Delete Features.
   - Style the List and Highlight Selected Users Dynamically.

Day 2: React Hooks and Forms

Topics Covered

1. **Functions as Props**
   - Passing Functions Between Components
   - Using Callback Functions for Communication
2. **Hooks**
   - `useEffect`: Fetching Data and Running Side Effects
   - `useRef`: Accessing DOM Elements and Maintaining Mutable References
3. **Forms in React**
   - Controlled vs Uncontrolled Components
   - Handling Form Submissions
   - Styling Forms with CSS
4. **Validation with react-hook-form**
   - Setting Up react-hook-form
   - Adding Form Validations
   - Custom Error Messages
   - Styling Error States
5. **Optimization**
   - Using `useMemo` to Optimize Expensive Calculations
   - Using `useCallback` to Prevent Unnecessary Function Re-Creation
6. **Hands-On Practice**
   - Build a User Registration Form with Validation.
   - Style the Form with CSS and Add Visual Feedback for Validations.

Day 3: React Router, HOCs, and API Integration

Topics Covered

1. **React Router**
   - Setting Up React Router
   - Creating and Navigating Between Routes
   - Dynamic and Nested Routes
2. **Higher-Order Components (HOCs)**
   - What Are HOCs?
   - Creating Layouts with HOCs
   - Styling Layouts Dynamically Using Themes
   - Implementing Private Routes with HOCs
3. **Error Handling**
   - Creating an Error Boundary
   - Building a Custom Error Page for Undefined Routes
   - Styling Error Pages for Better UX
4. **API Calling Using Axios**
   - Setting Up Axios
   - Making GET and POST Requests
   - Handling Errors in API Calls
5. **Hands-On Practice**
   - Build a Multi-Page Application with React Router and HOCs.
   - Add API Integration for Data Fetching.

Day 4: State Management and Advanced Topics

Topics Covered

1. **State Management with Redux**
   - Understanding Redux Basics
   - Setting Up Redux Toolkit
   - Creating Reducers, Actions, and Slices
   - Dispatching Actions and Accessing State
2. **Redux with API Calls**
   - Managing API State with Redux Toolkit
   - Using `createAsyncThunk` for Async Actions
3. **Styling Best Practices**
   - Organizing CSS in Larger Applications
   - Using Utility-First CSS Frameworks (e.g., Tailwind CSS)
   - Dynamic Themes in React Applications
4. **Performance Optimization**
   - Asynchronous Rendering in React
   - Optimizing State Updates with Redux Toolkit
   - Using Lazy Loading for Routes and Components
5. **Hands-On Practice**
   - Extend the Multi-Page Application:
     - Add Redux for State Management.
     - Integrate API Calls with Redux Toolkit.
     - Apply Advanced Styling Techniques.
