1. Introduction to React

A. What is React?
B. Features and advantages of React.js
C. Comparing React with other front-end libraries/frameworks
D. Setting up React development environment
E. Understanding JSX and Virtual DOM

2. React Fundamentals

A. React Components and Props
B. Functional vs Class Components
C. Rendering Elements in React
D. State and Lifecycle basics
E. Handling events in React

3. Components & Props

A. Component creation and reusability
B. Props and Default Props
C. List and keys rendering
D. React fragments and portals

4. State & Hooks

A. Introduction to Hooks
B. useState hook
C. useEffect hook
D. useRef, useMemo, useCallback
E. Custom Hooks creation

5. React Router & Navigation

A. React Router basics
B. Dynamic routing
C. Navigation using Link and NavLink
D. Route parameters and redirects

6. State Management

A. Context API for global state
B. useContext hook
C. Redux fundamentals (Actions, Reducers, Store)
D. Redux Toolkit basics

7. API Integration

A. Fetching data using Fetch API
B. Using Axios for HTTP requests
C. Handling API responses and errors
D. Asynchronous operations with async/await

8. Styling & UI in React

A. CSS in React (inline, stylesheet)
B. Styled Components
C. Integration with UI libraries (Material-UI, Tailwind CSS)
D. Responsive design principles

9. Performance & Advanced Concepts

A. Performance optimization techniques
B. Code splitting and lazy loading
C. Error boundaries and debugging
D. Higher-order components and render props

10. Testing & Deployment

A. Introduction to unit testing with Jest
B. React Testing Library basics
C. Debugging React applications
D. Deploying React apps on platforms like Netlify and Vercel

Practical:

- Build a To-Do List application using hooks
- Create a portfolio website with routing
- Fetch and display API data app
- Deploy a real React project online