# **Advanced React Application Development**

Prerequisites: Working Knowledge of React JS

## Day 1: Core React Features and State Management

## **Session 1: React Routing and Navigation**

- Duration: 2 hours (1-hour lecture, 1-hour lab)
- Topics Covered:
  - Introduction to React Router
  - Configuring routes (BrowserRouter, Route, Switch, Routes)
  - Dynamic routing and route parameters
  - Nested routes and layouts
  - o Using Link, NavLink, and useNavigate for navigation
  - Protected routes and authentication flow
- Lab Activity:
  - Build a multi-page application (e.g., a blog):
    - Home page, Blog list page, Single blog page (dynamic route).
    - Add navigation between pages with NavLink.
    - Implement a protected route for creating a new blog post.

## Session 2: State Management with Redux

- Duration: 2.5 hours (1.5-hour lecture, 1-hour lab)
- Topics Covered:
  - Introduction to Redux: Store, Reducers, Actions
  - Understanding the Redux data flow
  - Middleware (redux-thunk and redux-saga)
  - o Integration with React using react-redux
  - Using Redux DevTools for debugging
- Lab Activity:
  - Build a simple task management app:
    - Use Redux to manage a global state for tasks (add, update, delete tasks).
    - Implement async actions using redux-thunk to simulate saving tasks to a server.

Debug the app using Redux DevTools.

## Day 2: Data Handling, Testing, and Optimization

## Session 1: Data Fetching and API Integration

- **Duration:** 2.5 hours (1.5-hour lecture, 1-hour lab)
- Topics Covered:
  - Fetching data using the Fetch API and axios
  - o Understanding React Query for server state management
  - Handling loading and error states
  - Pagination and infinite scrolling
  - Integrating APIs into a Redux workflow
- Lab Activity:
  - Enhance the task management app:
    - Fetch task data from a mock API using axios.
    - Add loading and error indicators.
    - Implement pagination for tasks using React Query.

## Session 2: Testing and Debugging

- **Duration:** 2 hours (1.5-hour lecture, 30-minute lab)
- Topics Covered:
  - Introduction to testing frameworks: Jest and React Testing Library
  - Writing unit tests for React components
  - Testing Redux actions and reducers
  - Mocking API calls for testing
  - Debugging tools: React DevTools and console debugging
- Lab Activity:
  - Write tests for the task management app:
    - Test the rendering of task components.
    - Mock API calls to test data fetching logic.
    - Debug issues using React DevTools and console logs.

## **Session 3: Performance Optimization and Best Practices**

- **Duration:** 2 hours (1-hour lecture, 1-hour lab)
- Topics Covered:
  - Common performance issues in React applications
  - Optimizing re-renders with React.memo and useMemo
  - Lazy loading components using React.lazy and Suspense
  - Code splitting and bundling with Webpack
  - Best practices for folder structure, hooks, and component design
- Lab Activity:
  - Optimize the task management app:
    - Identify and reduce unnecessary re-renders.
    - Implement lazy loading for routes and components.