

Reactjs Course Outline by Naveed Sarwar

1. Introduction to Advanced React.js

- **1.1. Overview of React 18.x Features**
 - Concurrent Rendering
 - Automatic Batching
 - Transition API
- **1.2. React.js Ecosystem in 2024**
 - Popular Libraries and Tools
 - Trends and Best Practices
- **1.3. Project Setup**
 - Setting up a Modern React Development Environment
 - Introduction to Vite, Next.js 13, and other build tools

Project Idea: "React Portfolio 2024" - Build a modern, responsive portfolio website utilizing React 18.x features, including concurrent rendering for fast-loading content and transitions for smooth animations.

2. Advanced Components and Props

- **2.1. Component Composition**
 - Understanding Component Hierarchies
 - Composing Components for Reusability
- **2.2. Advanced Props Handling**
 - PropTypes and Default Props
 - Handling Complex Props (Arrays, Objects, Functions)
- **2.3. Higher-Order Components (HOCs)**
 - Creating and Using HOCs
 - HOCs vs. Render Props vs. Custom Hooks
- **2.4. Render Props Pattern**
 - Understanding and Implementing Render Props
 - Comparison with Other Patterns

Project Idea: "Component Library" - Develop a reusable component library for forms, buttons, and other UI elements, emphasizing the use of advanced props handling, HOCs, and render props patterns.

3. Designing with React: Material-UI and Tailwind CSS

- **3.1. Introduction to Material-UI**
 - Setting up Material-UI in a React Project
 - Customizing Themes and Components

- Advanced Components (Data Grid, Dialogs, etc.)
- **3.2. Introduction to Tailwind CSS**
 - Setting up Tailwind in a React Project
 - Utility-First CSS in React
 - Building Responsive Layouts with Tailwind
- **3.3. Combining Material-UI and Tailwind**
 - Best Practices for Using Both Libraries Together
 - Case Studies: When to Use Material-UI vs. Tailwind

Project Idea: "Dashboard UI Kit" - Create a comprehensive UI kit for a dashboard application using both Material-UI and Tailwind CSS, focusing on custom theming and responsive design.

4. Handling Events in React

- **4.1. Overview of Event Handling in React**
 - Understanding Synthetic Events
 - Event Propagation and Bubbling
- **4.2. Common Event Listeners**
 - Handling onClick, onBlur, onChange, and Other Events
- **4.3. Custom Event Handling**
 - Creating Custom Event Handlers
 - Managing Performance in Event-Heavy Applications

Project Idea: "Interactive Form Elements" - Create a set of form elements (e.g., inputs, checkboxes, dropdowns) that respond to user interactions with custom event handling, focusing on accessibility and performance.

5. Advanced Forms and Validation

- **5.1. Managing Complex Forms**
 - Formik and React Hook Form
 - Nested Forms and Dynamic Form Fields
- **5.2. Validation Techniques**
 - Implementing Validation with Yup
 - Custom Validation Logic
- **5.3. Integrating Forms with APIs**
 - Handling Form Submissions and Asynchronous Validation
 - Implementing File Uploads and Image Previews

Project Idea: "Event Registration System" - Create a complex event registration form with dynamic fields, validation, and integration with a backend API to handle submissions.

6. Advanced State Management

- **6.1. Context API Beyond Basics**
 - Optimizing Context for Performance
 - Dynamic Context Providers
- **6.2. Redux Toolkit in 2024**
 - Advanced Patterns with Redux Toolkit
 - Creating Custom Middleware
- **6.3. Recoil.js**
 - Introduction to Recoil
 - Building a Recoil State Management System

Project Idea: "Real-Time Collaborative Text Editor" - Implement a text editor that allows multiple users to collaborate in real-time. Use Context API or Recoil for state management, integrating WebSockets for real-time updates.

7. Advanced Hooks and Custom Hooks

- **7.1. Deep Dive into Built-in Hooks**
 - useMemo and useCallback for Performance Optimization
 - useRef for DOM Manipulation and Persisting State
- **7.2. Custom Hooks**
 - Creating Reusable Hooks
 - Integrating External APIs
- **7.3. Handling Side Effects with useEffect**
 - Best Practices for useEffect
 - Managing Asynchronous Operations

Project Idea: "Personalized Dashboard" - Create a dashboard that pulls in data from various APIs (e.g., weather, news, and social media) using custom hooks to manage API calls and state.

8. Performance Optimization Techniques

- **8.1. Code Splitting and Lazy Loading**
 - Implementing React.lazy and Suspense
 - Route-based Splitting with React Router
- **8.2. Memoization Techniques**
 - useMemo and useCallback in Complex Components
 - Memoizing Components with React.memo
- **8.3. Profiling and Debugging**
 - Using React Profiler
 - Debugging Performance Bottlenecks

Project Idea: "E-commerce Product Gallery" - Develop a dynamic product gallery for an e-commerce site with image lazy loading, code splitting, and optimized performance for large data sets.

9. Server-Side Rendering (SSR) and Static Site Generation (SSG)

- **9.1. Introduction to Next.js 13**
 - SSR vs. SSG: When to Use Each
 - Implementing SSR and SSG with Next.js
- **9.2. Dynamic Routing in Next.js**
 - Creating Dynamic Pages with `getStaticPaths` and `getServerSideProps`
- **9.3. Hybrid Applications**
 - Combining SSR, SSG, and Client-Side Rendering

Project Idea: "Next.js Blog Platform" - Build a blog platform using Next.js that supports both static generation for SEO and server-side rendering for dynamic content.

10. TypeScript with React

- **10.1. Introduction to TypeScript in React**
 - TypeScript Basics in React Components
 - Typing Props, State, and Hooks
- **10.2. Advanced TypeScript Patterns**
 - Union Types, Generics, and Utility Types
 - Creating Reusable Components with TypeScript
- **10.3. Integration with Redux and Context API**
 - Typing Redux Actions, Reducers, and Thunks
 - Typing Context Providers and Consumers

Project Idea: "TypeScript Todo App with Drag and Drop" - Create a Todo application with drag-and-drop functionality, ensuring all components and states are strongly typed using TypeScript.

11. Advanced Routing and Navigation

- **11.1. Advanced React Router Techniques**
 - Nested Routes and Route Guards
 - Programmatic Navigation and Redirects
- **11.2. Dynamic Routing and Query Parameters**
 - Handling Dynamic URLs and Query Strings
 - Route-based Code Splitting
- **11.3. Handling Authentication and Authorization**
 - Implementing Protected Routes
 - Role-Based Access Control

Project Idea: "Multi-User Blogging Platform" - Develop a blogging platform with multi-user support, including roles and permissions, and dynamic routing for user profiles and posts.

12. Testing React Applications

- **12.1. Introduction to Testing in React**
 - Unit Testing with Jest and React Testing Library
 - Snapshot Testing
- **12.2. End-to-End Testing**
 - Introduction to Cypress
 - Writing and Running E2E Tests
- **12.3. Test-Driven Development (TDD)**
 - Implementing TDD in React Projects
 - Mocking and Testing Asynchronous Code

Project Idea: "Test-Driven Todo App" - Develop a Todo application following TDD principles, ensuring every component and functionality is covered by tests.

13. Integrating React with Backend Services

- **13.1. Fetching Data with Axios and Fetch API**
 - Handling API Requests and Responses
 - Caching API Responses
- **13.2. WebSockets and Real-Time Communication**
 - Integrating WebSockets for Real-Time Data
 - Handling WebSocket Connections in React
- **13.3. GraphQL Integration**
 - Querying and Mutating Data with Apollo Client
 - Optimizing GraphQL Queries

Project Idea: "Real-Time Chat Application" - Build a real-time chat application using WebSockets for instant messaging and GraphQL for fetching user data and messages.

14. React and Microfrontends

- **14.1. Introduction to Microfrontends**
 - Benefits and Use Cases
 - Implementing Microfrontends with React
- **14.2. Module Federation with Webpack**
 - Sharing Components Across Applications
 - Lazy Loading and Version Control in Microfrontends
- **14.3. Deploying Microfrontends**
 - Deployment Strategies for Microfrontends

- Handling Shared State and Communication

Project Idea: "Microfrontend E-commerce Platform" - Create an e-commerce platform where different parts of the application (e.g., product listings, checkout) are managed as separate microfrontends.

15. Advanced Animations and Transitions

- **15.1. Introduction to React Transition Group**
 - Implementing Enter/Exit Animations
 - Managing Animation States
- **15.2. Advanced Animation Libraries**
 - Using Framer Motion for Complex Animations
 - Creating Custom Animations with CSS-in-JS
- **15.3. Performance Considerations**
 - Optimizing Animations for Performance
 - Avoiding Jank and Lag in Animations

Project Idea: "Interactive Storybook" - Develop an interactive, animated storybook application using Framer Motion and React Transition Group to create smooth, engaging animations.

16. Progressive Web Apps (PWA) with React

- **16.1. Introduction to PWAs**
 - What Makes a PWA?
 - Benefits of PWAs
- **16.2. Building a PWA with React**
 - Implementing Service Workers
 - Caching Strategies and Offline Support
- **16.3. Deploying a PWA**
 - Best Practices for PWA Deployment
 - Handling Updates and Notifications

Project Idea: "PWA Weather App" - Build a weather application as a PWA, including offline support, push notifications, and caching strategies for fast performance.

Regards:

Naveed Sarwar

+923001011134

[Youtube](#) | [Facebook](#) | [Tiktok](#) | [Instagram](#) | [Linkedin](#) | [Facebook Shorts](#) | [WhatsApp Channel](#)