## **React Training 2: Advanced React Development (2-3 Weeks)**

### **10. Advanced State Management**

* State Management Recap: Context API vs. Prop Drilling
* Using Redux Toolkit with React  
  + Store, Reducers, Actions, Middleware
  + Async logic with createAsyncThunk
* Comparison with Zustand, Jotai, and Recoil
* Combining Context with useReducer

### **11. Authentication and Authorization**

* Protecting Routes with React Router
* Storing Tokens Securely (LocalStorage vs. Cookies)
* JWT-based Authentication Flow
* OAuth2 & Social Login (Google, GitHub)
* Role-based Access Control in React UI

### **12. Custom Hooks & Reusable Logic**

* Building Your Own Custom Hooks
* Managing Refs, Event Listeners, and Subscriptions
* Hook Composition Patterns
* When not to use Hooks

### **13. React with TypeScript (Optional Track)**

* Typing Props, State, and Events
* Creating Reusable Typed Components
* Typing Custom Hooks
* Type-safe APIs with Axios + TypeScript

### **14. Unit Testing & Integration Testing**

* Why Testing Matters in React
* Tools: Jest, React Testing Library
* Writing Unit Tests for Components and Hooks
* Mocking APIs and Events
* Snapshot Testing

### **15. Building Scalable Architecture**

* Atomic Design & Component Structuring
* Folder Structure Best Practices
* Component Reusability and Isolation
* Performance Optimization Patterns  
  + Code Splitting
  + Memoization (React.memo, useMemo)
  + Debouncing & Throttling

### **16. Real-time Features in React**

* WebSockets & Socket.IO Basics
* Using WebSocket in React (Live Chat, Notifications)
* Managing Socket Lifecycle in Components
* Broadcasting and Listening to Events

### **17. Introduction to Next.js (Bonus Week)**

* Why Next.js? Benefits over CRA
* Pages vs. App Router
* File-based Routing and Dynamic Routes
* Data Fetching: SSR, SSG, ISR, Client-side
* API Routes & Middleware
* Deploying a Next.js App (Vercel/Netlify)

### **Module 4 Assignments**

* Role-Based Dashboard  
  + Auth login, conditional route rendering, role-switching logic
* Chat App with WebSocket  
  + Real-time communication using Socket.IO
* Custom Component Library  
  + Build a reusable React UI kit (buttons, modals, inputs)
* Advanced Todo App  
  + Drag & Drop (react-beautiful-dnd), filters, state persistence
* React + Redux Toolkit App  
  + Large-scale app architecture using Redux Toolkit
* Optional: Migrate a project from React to Next.js