

React Learning Plan

1. Advanced Core Concepts

Topics:

- React Fiber (under the hood rendering mechanism)
- Reconciliation and Virtual DOM optimizations
- Strict Mode & Concurrent Features
- Error Boundaries

Problem Statements & Tasks:

- **Task 1:** Implement a custom error boundary that catches errors in child components and displays a fallback UI.
 - **Task 2:** Build a component tree with heavy renders and measure rendering performance using React Profiler. Optimize it using `React.memo`, `useCallback`, and `useMemo`.
-

2. Advanced Hooks & Patterns

Topics:

- Custom hooks with complex logic
- `useReducer` with middleware (logger, async actions)
- `useLayoutEffect` vs `useEffect`
- Context optimization with selectors
- Compound Components & Render Props

Problem Statements & Tasks:

- **Task 3:** Create a custom hook `useOnlineStatus` that tracks if the user is online/offline.
- **Task 4:** Implement a `useLocalStorage` hook that syncs state with `localStorage`.
- **Task 5:** Build a **multi-step form wizard** using `useReducer` for state management.

- **Task 6:** Create a **Modal component** using the **Compound Component pattern** (`<Modal><Modal.Header /><Modal.Body /><Modal.Footer /></Modal>`).
-

3. Performance Optimization

Topics:

- React Profiler and flame graph debugging
- Code splitting with React.lazy & Suspense
- Windowing/Lazy loading (react-window, react-virtualized)
- Debouncing & throttling inside React apps

Problem Statements & Tasks:

- **Task 7:** Build a search component with **debounced input**.
 - **Task 8:** Render a list of **10,000+ items efficiently** using `react-window`.
 - **Task 9:** Add code-splitting to a React app using `React.lazy` and `Suspense` for route-based components.
-

4. Advanced State Management

Topics:

- Redux Toolkit advanced patterns (createSlice, RTK Query)
- Zustand / Jotai / Recoil vs Redux
- Server-state vs Client-state separation
- Query caching and background refetching

Problem Statements & Tasks:

- **Task 10:** Implement a shopping cart using Redux Toolkit with persisted state.
- **Task 11:** Use **React Query** or **RTK Query** to fetch data with caching, refetch intervals, and optimistic updates.
- **Task 12:** Build a small project (e.g., Trello-like board) with Zustand or Jotai for global state.

5. Testing & Quality

Topics:

- React Testing Library (RTL) advanced use
- Jest mocks, mocha.
- Cypress / Playwright for E2E testing

Problem Statements & Tasks:

- **Task 16:** Write unit tests for a form component (validation + API call).
- **Task 17:** Add E2E tests for a login/logout flow with Playwright or Cypress.