React.js Performance and Rendering

■ React Fiber kya hai aur ye performance kaise improve karta hai?

React Fiber ek new reconciliation engine hai jo tasks ko small units mein tod kar kaam karta hai. Ye asynchronous rendering support karta hai, jisse smooth and responsive UI milta hai.

■ Code Splitting kya hota hai aur React mein iska kya fayda hai?

Code splitting ek technique hai jisme app ke code ko chhoti files mein tod diya jata hai. React mein `React.lazy` aur `Suspense` ka use karke lazy loading kiya jata hai. Ye app ko fast banata hai.

■ SSR vs CSR — kya difference hai?

Server Side Rendering (SSR) mein HTML server se render hoke aata hai. CSR mein JavaScript client side render karta hai. SSR SEO friendly aur faster first load deta hai.

■ Render Props kya hote hain? Use cases kya hain?

Render props ek technique hai jahan component ek function ke through UI render karta hai. Iska use reusable and dynamic components banane ke liye hota hai.

■ Badi lists ko render karte waqt optimization techniques kya hoti hain?

Virtualization techniques jaise `react-window`, `react-virtualized` use hote hain jisme sirf visible items render hote hain. Isse performance improve hoti hai.

Hooks & State Management

■ Memoization kya hota hai (React.memo, useMemo)? Iska kya use hai?

Memoization ka use expensive function calls ko cache karne ke liye hota hai. `React.memo` component ko unnecessary re-render se bachata hai. `useMemo` functions ke result ko memoize karta hai.

■ useEffect hook ka kya purpose hai aur kaise kaam karta hai?

useEffect ka use side-effects handle karne ke liye hota hai jaise API call, DOM update, subscriptions. Ye lifecycle methods ka combination hai.

■ Side Effects ko manage karne ke liye kaunse tarike hain?

`useEffect` (basic side-effects), `useLayoutEffect` (synchronous DOM changes), `useReducer` (complex state management) use hote hain.

■ React mein state management large applications ke liye kaise handle kiya jata hai?

Redux, Recoil, Zustand, Context API jaise tools ka use karke centralized aur scalable state management kiya jata hai.

Routing & Async Handling

■ Routing & Navigation React Router se kaise handle karte hain?

React Router DOM ka use karke `Route`, `Link`, `useNavigate` se navigation handle hota hai. Ye dynamic routing support karta hai.

■ Async/Await operations React mein kaise manage karte hain?

Async operations ko `useEffect` ke andar async function define karke handle karte hain. Try/catch blocks se error handle kiya jata hai.

■ Suspense ka use kya hai React mein?

Suspense lazy loaded components ke liye fallback UI dikhata hai jab tak component load nahi ho jata. Ye UX ko smooth banata hai.

Component Architecture

■ Context API kya hoti hai aur iska kya benefit hai?

Context API global data (like user, theme) ko direct child components tak provide karta hai bina props drilling ke.

■ Higher Order Components (HOC) kya hote hain aur kaise banate hain?

HOC ek function hai jo ek component ko input leta hai aur ek enhanced component return karta hai. Ye code reuse ke liye hota hai.

■ Controlled vs Uncontrolled components ka kya farq hai?

Controlled component React ke state se control hota hai. Uncontrolled component DOM ke reference (`ref`) se control hota hai.

■ Reusability, Modularity, Testability React components mein kaise maintain karte hain?

Chhote, reusable, pure components banakar aur unka independent logic likhkar in features ko maintain kiya jata hai.

Advanced Optimization

■ React mein reconciliation kya hota hai?

Reconciliation ek algorithm hai jisme React naya Virtual DOM purane se compare karta hai aur sirf changes apply karta hai.

■ Unnecessary re-renders ko kaise prevent karte hain?

`React.memo`, `useMemo`, `useCallback`, `shouldComponentUpdate` se performance improve ki jati hai.

■ Components ko style karne ke alag-alag tarike kya hain?

CSS Modules, Styled Components, Sass, Emotion, Tailwind CSS, CSS-in-JS jaise modern tools use kiye jaate hain.

■ Scalable React apps banate waqt key considerations kya hoti hain?

Folder structure, modular components, lazy loading, error boundaries, state separation, performance monitoring zaroori hote hain.