React.js Interview Questions & Answers (Basic to Advanced)

Q1. What is React?

React is a JavaScript library developed by Facebook for building user interfaces.

Q2. What are components in React?

Components are reusable, independent UI building blocks. They can be functional or class-based.

Q3. What is JSX?

JSX is a syntax extension that lets you write HTML-like code inside JavaScript.

Q4. Difference between props and state?

Props are read-only and passed from parent to child, while state is mutable and local to the component.

Q5. What is Virtual DOM?

A lightweight copy of the real DOM. React updates it first, then syncs with the real DOM efficiently.

Q6. What are React hooks?

Hooks let you use state and lifecycle features in functional components. Examples: useState, useEffect.

Q7. Explain useEffect.

Used for side effects such as fetching data, subscriptions, or DOM updates.

Q8. What is conditional rendering?

Rendering components based on conditions using operators like ?: or &&.

Q9. How do you handle forms in React?

By using controlled components (state-driven) or uncontrolled components (ref-driven).

Q10. What is React Router?

A library for navigation and routing in React apps.

Q11. What is Redux?

A state management library using a central store, actions, and reducers.

Q12. What is Context API?

A way to share state globally without prop drilling using React.createContext.

Q13. What are Higher Order Components (HOC)?

A function that takes a component and returns a new component with extended behavior.

Q14. What is code splitting?

Splitting code into smaller bundles with React.lazy and Suspense for performance.

Q15. What is server-side rendering (SSR)?

Rendering React on the server before sending HTML to the client (e.g., Next.js).

Q16. Performance optimization in React?

Use memoization (React.memo, useMemo, useCallback), code splitting, and virtualization.

Q17. Controlled vs Uncontrolled components?

Controlled: React manages form state. Uncontrolled: DOM manages state via refs.

Q18. What is reconciliation?

React’s process of diffing virtual DOM trees and updating only changed nodes.

Q19. What is React Fiber?

A new reconciliation engine (React 16+) for smoother, incremental rendering.

Q20. What is hydration?

Attaching event listeners to server-rendered static HTML to make it interactive.