React.js Interview Questions and Answers (3 Years Experience)

1. What is the virtual DOM and how does React use it?

The virtual DOM is a lightweight JavaScript representation of the real DOM. React compares the new virtual DOM tree with the previous one (diffing) and updates only the changed parts in the real DOM. This makes rendering more efficient.

2. Difference between functional and class components

Class components use ES6 classes and lifecycle methods. Functional components are simpler and use React Hooks to manage state and side effects.

3. What is JSX? Why do we use it?

JSX (JavaScript XML) allows us to write HTML-like syntax in JavaScript. It makes UI code more readable and is transpiled into React.createElement calls.

4. What are controlled and uncontrolled components?

Controlled components are managed by React state. Uncontrolled components manage their state via the DOM using refs.

5. Difference between props and state

Props are passed from parent to child and are read-only. State is managed inside the component and can change over time.

6. What are React Hooks?

Hooks let you use state and other React features in functional components. Examples: useState, useEffect, useRef, useContext.

7. How do useState and useEffect work?

useState declares a state variable. useEffect is used for side effects such as data fetching or subscriptions.

React.js Interview Questions and Answers (3 Years Experience)

8. Difference between useEffect and useLayoutEffect

useEffect runs after render is committed. useLayoutEffect runs before the browser paints the screen.

9. When would you use useRef?

To access a DOM element or persist a mutable value without triggering re-render.

10. useCallback vs useMemo

useCallback memoizes a function. useMemo memoizes a returned value. Both help avoid unnecessary recalculations or re-renders.

11. How do you structure a scalable React project?

Organize components by features or routes, use folders like /components, /hooks, /context, /utils, and keep concerns separated for maintainability.

12. What is prop drilling and how do you avoid it?

Prop drilling is passing data through many nested components. Avoid it using Context API, Redux, or other state management libraries.

13. Explain Higher-Order Components (HOC)

A HOC is a function that takes a component and returns a new one with additional functionality. Useful for cross-cutting concerns.

14. How do you implement lazy loading?

Use React.lazy and Suspense to load components only when needed. Helps improve performance by reducing bundle size.

15. Ways to optimize performance

Use memoization (React.memo, useMemo), avoid unnecessary re-renders, code splitting, lazy loading, and virtualization.

16. What are key props in lists?

Keys help React identify which items changed. They must be unique and stable to improve list rendering performance.

17. How does React Router work?

It uses the History API to change the URL without reloading the page. It maps URLs to components using routes.

18. How to redirect programmatically?

Use the useNavigate hook from react-router-dom and call navigate('/path') to redirect.

19. How do you handle forms?

Use controlled components with useState or libraries like Formik or React Hook Form for complex forms.

20. Form validation libraries

Popular choices include Formik with Yup, and React Hook Form. They offer schema-based validation and clean integration with UI.

21. Global state management options

Context API, Redux, Zustand, Recoil, or Jotai are commonly used based on project needs and complexity.

22. Context API vs Redux

Context is simple and built-in, suitable for small-scale state. Redux is more powerful, suited for large applications with complex state and middleware.

23. How to fetch data in React?

Use useEffect with fetch or axios. Ensure to handle loading and error states for a better user experience.

24. Loading and error states

Use useState to track loading and error. Display messages or spinners based on their values in the component.

25. Testing tools

Jest for unit tests, React Testing Library for component testing, and Cypress or Playwright for end-to-end testing.

26. What is SSR and hydration?

SSR renders HTML on the server. Hydration is when React binds events and takes control of the server-rendered HTML on the client.

27. What are React Portals?

They allow rendering components outside the main DOM tree. Useful for modals and tooltips.

28. What are error boundaries?

They catch JavaScript errors in the component tree and display a fallback UI. Implemented using class components with componentDidCatch.