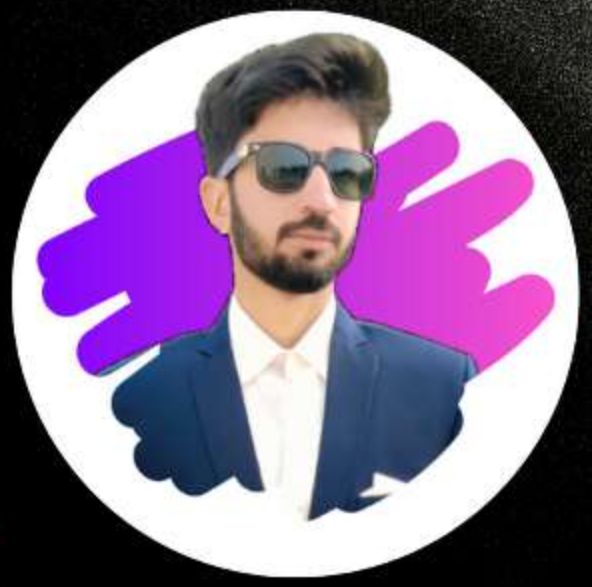
@codingbloodhound



10 React.js Interview Questions

Muneer Ahmed
@codingbloodhound

@codingbloodhound

What is React?

React is a JavaScript library used for building interactive and dynamic user interfaces, especially for single-page applications. It allows developers to create reusable components that manage their own state.

What are components in React?

Components in React are reusable building blocks of the user interface. Each component represents a part of the UI and can manage its own content, logic, and behavior. Components can be functional or class-based and can be composed together to build complex interfaces.

Muneer Ahmed
@codingbloodhound

@codingbloodhound

What is JSX?

JSX (JavaScript XML) is a syntax extension for JavaScript used in React. It allows you to write HTML like code within JavaScript, making it easier to create and visualize the structure of the UI. JSX is then compiled into regular JavaScript before being rendered to the DOM.

What is the Virtual DOM?

The Virtual DOM is a lightweight, in memory representation of the actual DOM in React. When the state of a component changes, React updates the Virtual DOM first, compares it with the previous version (a process called “diffing”), and the efficiently updates only the parts of the actual DOM that have changed. This improves performance by minimizing direct manipulation of the Real DOM.

Muneer Ahmed
@codingbloodhound

@codingbloodhound

What are Props?

Props (short for properties) are a mechanism in React for passing data from one component to another, typically from a parent component to a child component. Props allow components to be dynamic and reusable by providing them with different data and configurations. They are read-only and cannot be modified by the child component that receives them.

What is State in React?

State in React is an object that holds data or information about the component's current situation. It allows components to create dynamic and interactive user interfaces by managing and responding to change over time. Unlike props, which are passed from parent to child components, state is managed within the component itself and can be updated using setState function (for class components) or the useState hook (for functional components). Changes in state trigger re-renders of the component, reflecting the updated data in the UI.

Muneer Ahmed
@codingbloodhound

@codingbloodhound

How do you handle Events in React?

In React, events are handled using event handlers, which are functions that respond to user actions such as clicks, inputs, or form submissions. Here's how to handle events.

- Define an Event Handler: Create a function that will handle the event.

```
function handleClick() {
    alert('Button clicked!');
}
```

- Attach the Event Handler: Use JSX to attach the event handler to a component by adding the event attribute, like onClick, onChange, etc.

```
<button onClick={handleClick}>Click Me</button>
```

Muneer Ahmed
@codingbloodhound

@codingbloodhound

What are Hooks in React?

Hooks in React are special functions that allow you to use state and other React features in functional components. They are enable you to manage state, side effects, context, and more without needing class components. The most commonly used hooks are:

- useState: Allows you to add state to a functional component.

```
const [count, setCount] = useState(0);
```

- useEffect: Enables you to perform side effects, such as data fetching or subscriptions, in functional components.

```
useEffect(() => {
    // Code to run on component mount or update
}, [dependencies]);
```

Muneer Ahmed
@codingbloodhound

@codingbloodhound

What are Hooks in React? (Cont...)

- useContext: Lets you access context values in a functional component without needing to use the Context.Consumer.
- useReducer: An alternative to useState for managing complex state logic.
- useRef: Provides a way to create mutable reference that persist across renders.

Hooks must be called at the top level of a functional component and cannot be called conditionally. They simplify the process of sharing logic between components and improve code readability.

Muneer Ahmed
@codingbloodhound

@codingbloodhound

What is the purpose of useEffect?

The useEffect hook in React is used to perform side effects in functional components. Its primary purposes include:

- Data Fetching: It allows you to fetch data from APIs when the component mounts or updates.
- Subscriptions: You can setup subscriptions (e.g. to WebSocket or event listeners) and clean them up when the component unmounts.
- Manipulating the DOM: It enables direct DOM manipulation after React has rendered the component.
- Running Cleanup: You can return a cleanup function from useEffect to run when the component unmounts or before the effect run again, which helps prevent memory leaks.

Muneer Ahmed
@codingbloodhound

@codingbloodhound

What is the purpose of useEffect? (Cont...)

- Dependency Management: You can specify dependencies in an array to control when the effect should re-run, based on changes to those dependencies.

Here's a basic Example:

```
useEffect(() => {
    // Code to run on mount or when dependencies change
    const fetchData = async () => {
        const response = await fetch('https://api.example.com/data');
        const data = await response.json();
        setData(data);
    };
    fetchData();

    // Cleanup function (optional)
    return () => {
        // Cleanup code here
    };
}, [dependencies]);
```

In this example, the effect runs on component mount and whenever the specified dependencies change.

Muneer Ahmed
@codingbloodhound

@codingbloodhound

What is context in React?

Context in React is a feature that allows you to share data across components without having to pass props explicitly through every level of the component tree. It is used to manage global state or values, such as user settings or themes, that need to be accessible by multiple components.

Muneer Ahmed
@codingbloodhound

@codingbloodhound

CALL-TO-ACTION

want to Learn More?

Follow me for more tips on Web Development!

Muneer Ahmed
@codingbloodhound

@codingbloodhound

SHARE THIS INSIGHT NOW!

Muneer Ahmed
@codingbloodhound