Full-Stack Web Development

Full-Stack viva/interview questions.

1. What is React?

React is a free and open-source JavaScript library for building user interfaces (UIs) or UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies. React is not a framework; it's a library focused on the view layer of an application.

---

2. What is the difference between a library and a framework?

A library is a collection of pre-written code that developers can use to perform specific tasks. It gives developers control over the flow of the application. A framework, on the other hand, provides a complete structure for an application, dictating the flow and architecture. In simple terms, with a library, you call the code; with a framework, the framework calls your code.

---

3. What is JSX?

JSX stands for JavaScript XML. It's a syntax extension for JavaScript that allows you to write HTML-like code directly within your JavaScript files. React uses JSX to describe what the UI should look like. It's not mandatory to use JSX in React, but it makes the code much more readable and easier to write.

4. What is a component in React?

A component is a reusable, independent piece of UI. It's the core building block of any React application. Components let you split the UI into separate, isolated parts, making the code more organized and easier to maintain. There are two main types of components:

- **Functional Components:** These are JavaScript functions that return JSX. They are the preferred way to write components in modern React due to their simplicity and use of hooks.
- **Class Components:** These are ES6 classes that extend React.Component and include a render() method to return JSX.

---

5. What is state in React?

State is an object that holds data that may change over time. When a component's state changes, React re-renders the component to reflect the new data. State is local to a component and cannot be accessed by other components directly. You manage state in functional components using the useState hook.

---

6. What are props?

Props (short for properties) are a way to pass data from a parent component to a child component. They are read-only, meaning a child component cannot change its props directly. Props are essential for creating dynamic and reusable components.

---

7. What is a hook in React?

A hook is a special function that lets you "hook into" React features from functional components. Hooks allow you to use state and other React features (like lifecycle methods) without writing a class. The most common hooks for beginners are:

- useState(): To add state to a functional component.
- useEffect(): To perform side effects (like data fetching or DOM manipulation) in a functional component.

---

8. What is the Virtual DOM?

The Virtual DOM (VDOM) is a lightweight copy of the real DOM. When the state of a component changes, React creates a new VDOM tree and compares it to the previous one. It then calculates the most efficient way to update the real DOM to match the new VDOM. This process, called reconciliation, is much faster than directly manipulating the real DOM and is why React is so performant.