1. A controlled component is one where form data is handled by the component’s state using hooks like useState. An uncontrolled component manages its own state internally using refs (useRef). Controlled components offer more control and validation, while uncontrolled components are simpler and less verbose.
2. The Virtual DOM is a lightweight copy of the real DOM in memory. When a state changes, React updates the Virtual DOM first, calculates the difference (diffing), and efficiently updates only the changed parts of the real DOM. This improves performance and UI rendering.
3. Use effect runs asynchronously after Dom has painted and it’s mainly for data fetching and coming to useLayoutEffect runs synchronous before Dom is painted it’ll uses mainly to measure the Dom or prevent flickering
4. Hooks let you use state and other React features without writing a class. So e of the Hooks are UseState,useEffect,UseContext

useState: For managing component state

UseEffect: For side effects like fetching data.

UseContext: For accessing global state provided by React.Context

1. For large-scale apps, I would use Redux Toolkit or Context API with reducers. Redux is great for predictable state management and devtools, while Context is better for smaller apps.
2. Function components use hooks instead of lifecycle methods:

* useEffect(() => {}, []) = componentDidMount
* useEffect(() => {… return () => {…} }, [dep]) = componentDidUpdate + componentWillUnmount
* This way, we handle mount, update, and cleanup within the same hook.