Interview Questions

Q1.Name a few techniques to optimize React app performance. (Gojek)

Answer: There are many ways through which one can optimize the performance of a React app, let's have a look at some of them:

- Using useMemo( ) -

It is a React hook that is used for caching CPU-Expensive functions.
Sometimes in a React app, a CPU-Expensive function gets called repeatedly due to re-renders of a component, which can lead to slow rendering.
useMemo( ) hook can be used to cache such functions. By using useMemo( ), the CPU-Expensive function gets called only when it is needed.

- Using React.PureComponent -

It is a base component class that checks the state and props of a component to know whether the component should be updated.
Instead of using the simple React.Component, we can use React.PureComponent to reduce the re-renders of a component unnecessarily.

- Maintaining State Colocation -

This is a process of moving the state as close to where you need it as possible. Sometimes in React app, we have a lot of unnecessary states inside the parent component which makes the code less readable and harder to maintain. Not to forget, having many states inside a single component leads to unnecessary re-renders for the component.
It is better to shift states which are less valuable to the parent component, to a separate component.

- Lazy Loading -

It is a technique used to reduce the load time of a React app. Lazy loading helps reduce the risk of web app performances to a minimum.

Q2.What Are The Reasons Behind Re-Rendering In React? (Amazon)

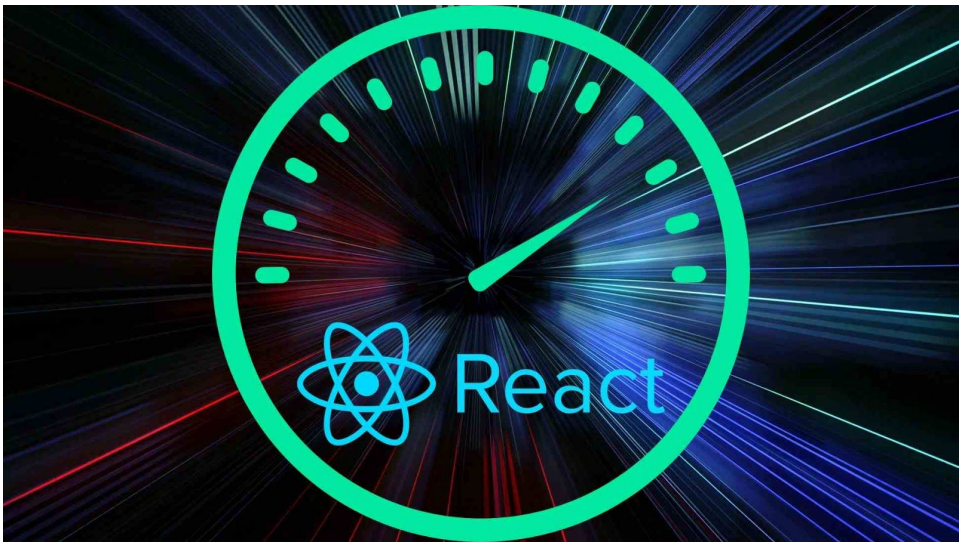
Answer: Reason for re-renders in React:

- Re-rendering of a component and its child components occurs when props or the state of the component has been changed.
- Re-rendering components that are not updated, affects the performance of an application.

Q3.What is ReactDOM, and what is the Difference Between ReactDOM and React? (OYO)

Answer: Earlier ReactDOM was part of React but later React and ReactDOM were split into two different libraries. Basically, ReactDOM works like glue between React and the DOM. We can use it for one single thing: mounting with ReactDOM.

ReactDOM.findDOMNode() which is another useful feature of ReactDOM can be used to access the DOM element. For the rest of the things React is there. React is used to define and create the elements, for lifecycle hooks, etc.



Q4.What is Redux? (Goldman Sachs)

Answer: Redux is a great way to store the entire application's state in a single store. When your application is small, you wouldn't be facing issues in handling the state. But when it starts growing you will find that state in various components is becoming unmanageable. Here Redux solves your problem.

Redux mainly works on three components:

- **Action**: Actions are payloads of information that send data from the application to the store. Actions are the only source of information for the store. We send them to the store using the store.dispatch().
- **Reducer**: Reducer specifies how the applications' state changes in response to actions sent to the store. Actions describe what happened, but it doesn't describe how the application's state changes. Basically, a reducer determines how the state will change to action.
- **Store**: Store objects bring the action and reducer together. You can access the state via getState(); It allows the state to be updated via dispatch (action);