**1.What is Redux?**

**A.**React Redux is the official React UI bindings layer for Redux. It lets your React components read data from a Redux store, and dispatch actions to the store to update state.

**2.What is Redux Thunk used for?**

**A**.The word "thunk" is a programming term that means "a piece of code that does some delayed work". Rather than execute some logic now, we can write a function body or code that can be used to perform the work later.

For Redux specifically, "thunks" are a pattern of writing functions with logic inside that can interact with a Redux store's dispatch and getState methods.

Using thunks requires the redux-thunk middleware to be added to the Redux store as part of its configuration.

Thunks are a standard approach for writing async logic in Redux apps, and are commonly used for data fetching. However, they can be used for a variety of tasks, and can contain both synchronous and asynchronous logic.

**3. What is Pure Component? When to use Pure Component over**

**Component**?

A. A Component is one of the core building blocks of React. In other words, we can say that every application you will develop in React will be made up of pieces called components. But React has two types of Components:

React.PureComponent: It is one of the most significant ways to optimize React applications. By using the pure component, there is no need for shouldComponentUpdate() Lifecycle Method as ReactJS Pure Component Class compares current state and props with new props and states to decide whether the component should re-render or Not.

React.Component: But on the other hand, React.Component re-renders itself every time the props passed to it changes, parent component re-renders or if the shouldComponentUpdate() method is called. It doesn’t optimize the React application. They are easy and fast to implement and also are good for very small UI views where a re-render wouldn’t matter that much. They provide cleaner code and fewer files to deal with.

When to use React.PureComponent?

You may choose React.PureComponent over React.Component if any of the below conditions is satisfied:

State/Props should be an immutable object

State/Props should not have a hierarchy

You should call forceUpdate() when data changes

**4.What is the second argument that can optionally be passed**

**tosetState and what is its purpose?**

**A**.The second parameter to setState() is an optional callback function that will be executed once setState is completed and the component is re-rendered. componentDidUpdate should be used instead to apply such logic in most cases. You may directly pass an object as the first argument to setState instead of a function