Module: 7 React - Applying Redux

1. What is Redux?
Redux is a pattern and library for managing and updating application state, using events called actions. Redux is a predication state container for Javascript apps. Redux is simplely a store to store the state of the variables in your app. Redux is a predictable state container designed to help you write Java script apps that behave consistently across client, server and native Redux.

2. What is Redux Thunk used for?
Redux Thunk middleware allows you to write action creators that return a function instead of an action. The thunk can be used to delay the dispatch of an action, or to dispatch only if a certain condition is met. Thanks are a standard approach for writing async logic in Redux apps, and are commonly used for data fetching. Redux Thunk is middleware that allows you to return functions, rather than just actions within Redux. Redux is a state management tool, which is used to store the state of different variables in our react application.

Redux Thunk is a popular middle ware for React Redux. You can create services that return a function hence the name Thunk. Thunk is a logical concept in programming where you deal with a function that is primarily used to delay the calcula or evaluation of any operation.

3. What is the pure component? When to use pure component over component?

Pure component: Implements a shallow comparison in shouldComponentUpdate and only re-renders when there are changes in its state or props.

Pure component the child component only re-renders if the props passed to it changes. A purecomponet in React is a class-based component that extends the React pure component class. pure component is similar to component but it skips re-renders for same props and state class components

Pure component is similar to component but it skips re-renders for same props and state. class components are still supported by React. but we don't recommend using them in new code.

4. What is the second argument that can optionally be passed to setstate and what is its purpose?

The second argument that can optionally be passed to setstate is a callback function which gets called immediatery after the setstate is completed and the components get re-rendered.

The setstate callback function is the second argument to setstute and is called after update is applied, and the component has re-rendered.

Setstate enqueues changes to the component state and tells React that this component and its children need to be re-rendered with the updated state. This is the primary method you use to update the user interface in response to event handlers and server responses. The callback function executed after the state update has been applied and the component has re-rendered. In the above example, the setstate callback function logs the message after the state has been updated with the data fetched from the server.