Assignment (Redux)

Que : 1) What is Redux ?

Ants : Redux is a predictable state container for JavaScript apps.

- It helps you write applications that behave consistently, run in different environments (client, server, and native), and are easy to test. On top of that, it provides a great developer experience, such as live code editing combined with a time traveling debugger.

**- Redux Toolkit =>**

Redux Toolkit is our official recommended approach for writing Redux logic. It wraps around the Redux core, and contains packages and

functions that we think are essential for building a Redux app. Redux Toolkit builds in our suggested best practices, simplifies most Redux tasks,

prevents common mistakes, and makes it easier to write Redux applications.

- Redux Toolkit (also known as "RTK" for short) is our official recommended approach for writing Redux logic. The @reduxjs/toolkit package wraps

around the core redux package, and contains API methods and common dependencies that we think are essential for building a Redux app. Redux

Toolkit builds in our suggested best practices, simplifies most Redux tasks, prevents common mistakes, and makes it easier to write Redux applications.

Redux is really:

A single store containing "global" state

Dispatching plain object actions to the store when something happens in the app

Pure reducer functions looking at those actions and returning immutably updated state

**Que : 2) What is Redux Thunk used for?**

Ans : 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. The inner function receives the store methods dispatch and getState

as parameters.

One of the main use cases for this middleware is for handling actions that might not be synchronous, for example, using axios to send a GET request.

Redux Thunk allows us to dispatch those actions asynchronously and resolve each promise that gets returned.

Once Redux Thunk has been installed and included in your project with applyMiddleware(thunk), you can start dispatching actions asynchronously.

**Que : 3) What is Pure Component? When to use Pure Component over Component?**

Ans : PureComponent Is Primarily Used for Performance Optimization. As outlined in the React docs: If your React component's render() function renders the same result given the same props and state, you can use React. PureComponent for a performance boost in some cases. Sometimes, it won’t make much sense to use PureComponent, even if it does help optimize performance. Given our example above, there wasn't really much of a need to make Child a stateful class component other than to use a console.log(). If there's no reason for it being a class, then it won't be extending either Component or PureComponent, and turning it into a class just to use PureComponent is unnecessary.

Que : 4) What is the second argument that can optionally be passed tosetState and what is its purpose?

Ans : 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.

If you want your program to update the value of a state using setState and then perform certain actions on the updated value of state then you must specify those actions in a function which should be the second argument of the setState. If we would not do so then those actions will be performed on the previous value of state because of asynchronous nature of setState.