React - Applying Redux

1. What is Redux?

Ans. Redux is like a **manager** that helps JavaScript applications handle a lot of information in a clean, organized way. When you're building an app, you often need to keep track of data (like user information, shopping carts, or app settings). Redux helps manage this data **in one central place**, so every part of your app can access it easily.

2. What is Redux Thunk used for?

Ans.

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

Ans. Pure component: A **Pure Component** in React is a special type of component. It only re-renders when there's a change in its **props** or **state**. If there's no change, React skips the rendering of that component, which can improve performance in certain cases.

Pure Components help you avoid unnecessary re-renders when the data (props/state) has not changed

=> When to Use `Component` (Instead of `Pure Component`)?

- When your component's state or props involves **complex objects** or **deep nested data structures**, where shallow comparison would not work well.
- If you manually want to control when the component should re-render (for example, by using `shouldComponentUpdate`).

- **Pure Component** = Checks if instructions are new before doing anything.
- **Component** = Always does the action even if it's the same as before.

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

Ans. The second argument that can be passed to the setState function in React is an optional callback function. This callback is executed after the state has been updated and the component has re-rendered.

Purpose of the Second Argument:

The purpose of this callback is to allow you to perform actions after the state update has been applied and the component has finished updating (re-rendering). Since setState is asynchronous, sometimes you need to do something right after the state change, and this callback helps with that.