React\_Redux\_Thiyori

Question 1: What is Redux, and why is it used in React applications? Explain the core concepts of actions, reducers, and the store.

Answer :1 Redux is a popular state management library often used with JavaScript applications, particularly with frameworks like React. It helps manage the state of an application in a predictable manner, making it easier to scale and debug complex apps.

### **🔹 Core Concepts of Redux**

1. **Store** – The centralized state container.
2. **Actions** – Objects that describe what to do (e.g., { type : "INCREMENT" }).
3. **Reducers** – Functions that determine how the state should change based on actions.
4. **Dispatch** – A function that sends an action to update the store.
5. **Selectors** – Functions that extract specific data from the store.

Question 2: How does Recoil simplify state management in React compared to Redux?

Answer 2: Recoil simplifies React state management compared to Redux by using a more intuitive, React-centric approach with atoms and selectors, requiring less boilerplate and offering a more streamlined experience, especially for smaller to medium-sized applications.

Here's a breakdown of how Recoil simplifies state management:

**No Centralized Store:**

Unlike Redux, which relies on a single global store, Recoil uses "atoms" as individual units of state, allowing for a more granular and flexible approach.

**Atoms and Selectors:**

Recoil uses "atoms" to represent state and "selectors" to derive data from atoms, offering a more intuitive and React-friendly way to manage state.

**Simplified API:**

Recoil's API is designed to be more straightforward and less complex than Redux, requiring less boilerplate code and making it easier to get started with state management in React.

**Mutable Data Structures:**

Recoil allows for mutable data structures, which can be easier to work with than the immutable data structures required by Redux.

**React-Centric Approach:**

Recoil is designed specifically for React applications, making it a natural fit for React developers and aligning well with React's principles of simplicity.

**Performance:**

Recoil is designed to be more performant than Redux, especially for applications with a large number of components that need to access the same state.