**Module-7) React - Applying Redux**

**State Management (Redux, Redux-Toolkit or Recoil)**

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

Redux is a state management library for JavaScript applications, most commonly used with React.  
It helps manage the global state of an application in a predictable and centralized way.  
In large applications where data needs to be shared among multiple components, Redux makes state handling easier, consistent, and maintainable.

* **Why is Redux used in React applications?**
* React components have their own local state, which works well for small apps.
* But in large apps, passing data between multiple nested components (prop drilling) becomes difficult.
* Redux provides a single source of truth (the store) for the entire app, allowing components to easily access and update shared data.
* **Key benefits:**
* Centralized and predictable state management
* Easy debugging (via Redux DevTools)
* Better scalability for large React projects
* Simplifies communication between components
* **Core Concepts of Redux:-**

Redux is a state management library used in React to manage and share data across components.  
It works on three main concepts:

1. Actions:  
   Actions are plain JavaScript objects that describe what happened in the app.  
   Each action has a type and an optional payload.  
   Example: { type: "INCREMENT" }
2. Reducers:  
   Reducers are pure functions that take the current state and an action, then return a new state based on the action type.
3. Store:  
   The Store is a single source of truth that holds the entire application’s state.  
   Components can access and update data through the store using dispatch() and getState().

Redux ensures a predictable, centralized, and organized way of managing application state.

**4. Redux Data Flow**

1. A component dispatches an action.
2. The reducer processes the action and returns a new state.
3. The store updates the state.
4. React components subscribed to the store automatically re-render with the new data.

**Q:-2 How does Recoil simplify state management in React compared to Redux?**

Recoil is a simple and flexible state management library for React developed by Facebook.  
It simplifies state handling compared to Redux by removing complex setups like actions, reducers, and store.  
Recoil uses atoms (units of state) and selectors (derived state) to manage both global and local states easily.  
It integrates directly with React using hooks such as useRecoilState() and useRecoilValue(), reducing boilerplate code.  
Unlike Redux, only components using a specific atom re-render when that state changes, improving performance.