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.

A-1: Redux is a state management library used in JavaScript applications, particularly with React, to manage application-wide state in a predictable and centralized manner.

→ It helps maintain a single source of truth for state, making debugging and state management more efficient, especially in large applications.

Why it is used in react application :-

- **Centralized State Management**: Redux stores the entire application state in a single store, making it easier to access and manage.
- **Improved Performance**: With React-Redux, only necessary components re-render when the state updates.

→ **Core concepts** :-

1. Store:- The store is the central container holding the entire state of the application.

- It provides methods to:

  1. Get the current state(store.getstate())
  2. Dispatch actions (store.dispatch(action))
  3. Subscribe to state changes(store.subscribe(listener))

State Management (Redux, Redux-Toolkit or Recoil)

2. Actions:- Actions are plain JavaScript objects that describe an event or intention to change the state.

→ Every action must have a type field, and it can also carry additional payload.

3. Reducers:- Reducers are pure functions that take the current state and an action, then return a new state.

→ They must be immutable.

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

A-2:

Recoil	Redux
• Minimal setup, state defined with atoms	• Requires store, actions, reducers, and middleware
• Easy to understand and use	• Steeper learning curve
• Components automatically re-render when atom state changes	• Entire Redux store updates when state changes
• Uses independent atoms, making state modular and local-friendly	• Requires middleware like Redux Thunk or Redux Saga