**Question 1: What is Redux, and why is it used in React applications?**

**ANS:**

* **Redux Overview:**

Redux is a state management library for JavaScript applications, often used with React to

manage and centralize the application's state in a predictable way. It helps avoid prop drilling and

makes state predictable across large applications.

* **Why Redux is Used:**

1. Ensures a single source of truth (the store).
2. Helps in debugging and state tracking using Redux DevTools.
3. Makes app behavior predictable and easier to test.
4. Ideal for large-scale applications with complex state.

* **Core Concepts:**

**1. Actions**

Actions are plain JavaScript objects that describe what you want to do.

Example:

{ type: 'INCREMENT' } or { type: 'ADD\_TODO', payload: 'Buy Milk' }

**2. Reducers**

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

Example:

function counterReducer(state = { count: 0 }, action) { switch (action.type) {

case 'INCREMENT': return { count: state.count + 1 }; default: return state; }

}

**3. Store**

The central object that holds the entire application state.

Example:

import { createStore } from 'redux'; const store =

createStore(counterReducer);

**Question 2: How does Recoil simplify state management in React**

**compared to Redux?**

**ANS:**

* **Recoil Overview**

Recoil is a state management library developed by Facebook for React. It simplifies how state is

shared across components using atoms and selectors, with minimal setup and better

performance.

* **Advantages of Recoil Over Redux:**

1. Simpler Setup: No need for actions, reducers, or middleware.
2. Less Boilerplate: Uses React hooks like `useRecoilState`.
3. Scoped State: Each atom re-renders only the components that depend on it.
4. Async Support: Handles asynchronous data using selectors.
5. Integrated with React: Works with React’s concurrent features and suspense.