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

Redux is a state management library that helps manage the global state of an application in a predictable and centralized way.

Why Use Redux in React Applications?

1. **Centralized State Management** – All application state is stored in a single source of truth (the store), making debugging and tracking changes easier.
2. **Predictable State Updates** – State changes follow a strict unidirectional data flow, ensuring consistency.
3. **Easier Debugging** – With tools like Redux DevTools, developers can track and revert state changes.
4. **Simplifies Prop Drilling** – Redux helps avoid "prop drilling" by making state accessible from a global store rather than passing it through multiple components.

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

1. **No Need for Actions and Reducers** – Unlike Redux, which requires defining actions, reducers, and a store, Recoil directly manages state using **atoms** and **selectors**.

2. **More Granular State Updates** – In Redux, the entire state tree updates when an action is dispatched. In Recoil, only components that subscribe to a specific atom (state unit) re-render.

3.**Less Boilerplate** – Recoil uses **atoms** for state and **selectors** for derived state, making it more intuitive and concise.

4. **Built-in Asynchronous Support** – Recoil **natively** supports asynchronous state updates, whereas Redux typically requires middleware like **Redux Thunk**.