**React - Applying Redux**

**THEORY EXERCISE**

**Question 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** used with React (and other libraries) to manage global application state in a predictable way.
* It helps in maintaining a **single source of truth** (the store), making state changes predictable, traceable, and easier to debug.

**Core Concepts:**

* **Actions** – Plain JavaScript objects that describe *what happened*.
* { type: 'INCREMENT' }
* **Reducers** – Pure functions that specify *how the state changes* based on actions.
* function counterReducer(state = 0, action) {
* switch(action.type) {
* case 'INCREMENT': return state + 1;
* case 'DECREMENT': return state - 1;
* default: return state;
* }
* }
* **Store** – Holds the whole state tree of the app. Only way to change the state is by dispatching an action.

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

* **Recoil** is a state management library by Facebook specifically designed for React.
* It is **simpler and more React-friendly** than Redux — no need for actions, reducers, or boilerplate.
* Recoil uses **atoms** (pieces of state) and **selectors** (derived state) and allows shared and component-local state with minimal setup.

**Advantages:**

* Less boilerplate than Redux
* Works well with React's Concurrent Mode
* Directly integrates with functional components and hooks

**LAB EXERCISE**

**Task 1: Counter App using Redux**

**Steps:**

1. Install Redux & React-Redux:

npm install redux react-redux

1. Create Redux setup:

// counterSlice.js

const initialState = { count: 0 };

export const increment = () => ({ type: 'INCREMENT' });

export const decrement = () => ({ type: 'DECREMENT' });

export const counterReducer = (state = initialState, action) => {

switch (action.type) {

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

case 'DECREMENT': return { count: state.count - 1 };

default: return state;

}

};

// store.js

import { createStore } from 'redux';

import { counterReducer } from './counterSlice';

const store = createStore(counterReducer);

export default store;

// App.js

import React from 'react';

import { useSelector, useDispatch } from 'react-redux';

import { increment, decrement } from './counterSlice';

function App() {

const count = useSelector(state => state.count);

const dispatch = useDispatch();

return (

<div>

<h2>Counter: {count}</h2>

<button onClick={() => dispatch(increment())}>+</button>

<button onClick={() => dispatch(decrement())}>-</button>

</div>

);

}

export default App;

// index.js

import React from 'react';

import ReactDOM from 'react-dom';

import { Provider } from 'react-redux';

import store from './store';

import App from './App';

ReactDOM.render(<Provider store={store}><App /></Provider>, document.getElementById('root'));