**Counter Slice:**

import { createSlice } from "@reduxjs/toolkit";

const counterSlice = createSlice ({

    name: 'counter',

    initialState : { count: 0 },

    reducers: {

        increment: (state) => {state.count += 1},

        decrement: (state) => {state.count -= 1}

    }

})

export const {increment, decrement} = counterSlice.actions

export default counterSlice.reducer

**Store:**

import { configureStore } from "@reduxjs/toolkit";

import counterReducer from './counterSlice'

const store = configureStore ({

    reducer : {

        counter : counterReducer

    }

})

export default store;

**Main.jsx:**

import { StrictMode } from "react";

import { createRoot } from "react-dom/client";

import "./index.css";

import App from "./App.jsx";

import { Provider } from "react-redux";

import store from "../store.js";

createRoot(document.getElementById("root")).render(

    <Provider store={store}>

        <App />

    </Provider>

);

**App.jsx:**

import React from "react";

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

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

export default function App() {

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

    const dispatch = useDispatch();

    return (

        <div>

            <h1>Count: {count} </h1>

            <button

                onClick={() => {

                    dispatch(increment());

                }}

            >

                +

            </button>

            <button

                onClick={() => {

                    dispatch(decrement());

                }}

            >

                -

            </button>

        </div>

    );

}