Develop an app using Redux

Exp. No.: 11

Name: Akash M

Roll No.: 19CSR006

Code:

App.js:

```
import { useDispatch, useSelector } from "react-redux";
import { increment, decrement, reset } from "./actions";
import "./App.css";

function App() {
  const counter = useSelector((state) => state);
  const dispatch = useDispatch();
  return (
    <div className="app">
      <h1>Counter App using Redux</h1>
      <div class="counter__container">
        <h1>Counter Value: {counter} </h1>
        <button
          onClick={() => {
            dispatch(increment());
          }}
        >
          +
        </button>
        <button
```

```
        onClick={() => {
          dispatch(decrement());
        }}
      >
        -
      </button>
      <button
        onClick={() => {
          dispatch(reset());
        }}
      >
        Reset
      </button>
    </div>
  </div>
 );
}

export default App;
```

Index.js:

```
import React from "react";
import ReactDOM from "react-dom/client";
import "./index.css";
import App from "./App";
import reportWebVitals from "./reportWebVitals";
import { createStore } from "redux";
import counterReducer from "./reducers";
import { Provider } from "react-redux";

const root = ReactDOM.createRoot(document.getElementById("root"));
const store = createStore(counterReducer);
```

```
root.render(
  <React.StrictMode>
    <Provider store={store}>
      <App />
    </Provider>
  </React.StrictMode>
);
reportWebVitals();
```

actions.js:

```
export const decrement = () => {
  return {
    type: "DECREMENT",
  };
};

export const increment = () => {
  return {
    type: "INCREMENT",
  };
};

export const reset = () => {
  return {
    type: "RESET",
  };
};
```

reducers.js:

```
const counterReducer = (state = 0, action) => {
  switch (action.type) {
    case "INCREMENT":
```

```
    return state + 1;
  case "DECREMENT":
    return state - 1;
  case "RESET":
    return 0;
  default:
    return state;
 }
};

export default counterReducer;
```

OUTPUT: