actions.js

export const IncAction = (value) => async dispatch => {

    dispatch({

        type: "INCREMENT",

        payload:value

    })

}

export const DecAction = () => async dispatch => {

    dispatch({

        type:"DECREMENT"

    })

}

Reducer.js

const count = 0;

export default function reducer(state = count,action){

    const {type,payload} = action;

    switch(type){

        case "INCREMENT":

            return state + payload;

        case "DECREMENT":

            return state - 1;

        default:

            return state;

    }

}

Store.js

import { createStore, applyMiddleware } from 'redux';

import { composeWithDevTools } from 'redux-devtools-extension';

import thunk from 'redux-thunk';

import reducer from './reducer';

const middleware = [thunk];

const store = createStore(

    reducer,

    composeWithDevTools(applyMiddleware(...middleware))

);

export default store;

App.js

import React from "react";

import { connect } from "react-redux";

import { IncAction } from "./actions";

import { DecAction } from "./actions";

function App({ local\_variable, IncAction, DecAction }) {

  return (

    <div>

      <center>

        <h1>{local\_variable}</h1><br/>

        <button onClick={()=>IncAction(5)} className="btn btn-primary">Increment</button>&nbsp;

        <button onClick={DecAction} className="btn btn-success">Decrement</button>

      </center>

    </div>

  );

}

const mapStateToProps=(state)=>({

  local\_variable :state

})

export default connect(mapStateToProps,{IncAction, DecAction})(App);

index.js

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import 'bootstrap/dist/css/bootstrap.min.css';

import App from './App';

import reportWebVitals from './reportWebVitals';

import store from './store';

import { Provider } from 'react-redux';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

  <React.StrictMode>

    <Provider store={store}>

       <App />

    </Provider>

  </React.StrictMode>

);

reportWebVitals();