**Lesson 04 Demo 02**

**Creating a React Application to combineReducer Function**

**Objective:** To create a React application with Redux for combining two reducers in one store to perform the task based upon the reducer

**Tools required:** Node.js and React.js

**Prerequisites:** None

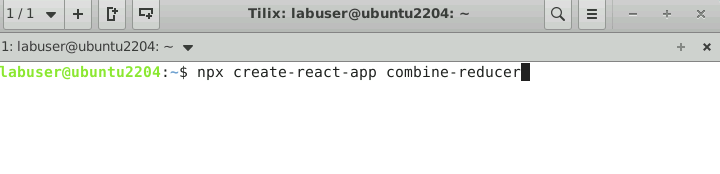
Steps to be followed:

1. Create and set up the React project
2. Create the actions and reducers folder
3. Create an App.js and index.js file
4. Add Bootstrap features in the index.html file
5. Test the application

**Step 1: Create and set up the React project**

1. Open a terminal window and run the following command to create a React application:

**npx create-react-app combine-reducer**



1. Open the created React application folder (**combine-reduce**) in VS Code by clicking on **File** in the top left corner and selecting **Open Folder**

**A screenshot of a computer

Description automatically generated**

1. Click on **Open** button

A screenshot of a computer

Description automatically generated

The folder structure appears as follows:

A screenshot of a computer

Description automatically generated

1. Inside the project, open the **TERMINAL** and run the following command to install the required dependencies:

**npm install**

**A screenshot of a computer

Description automatically generated**

1. Open the **package.json** file and view the external dependencies

**A screenshot of a computer

Description automatically generated**

**Step 2: Create the actions and reducers folder**

1. Right-click on the **src** folder and select **New Folder**

**A screenshot of a computer

Description automatically generated**

1. Create folders named **actions** and **reducers**

A screen shot of a computer

Description automatically generated

1. Inside the **actions** folder, create a file named **index.js** and enter the following code:

**//this function will be used by our /reducers/counter.js file**

**export const increment = (num) => {**

**return {**

**//we could name type key anything we want but we shouldnt**

**type: 'INCREMENT',**

**payload: num**

**}**

**}**

**export const decrement = () => {**

**return {**

**type: 'DECREMENT'**

**}**

**}**

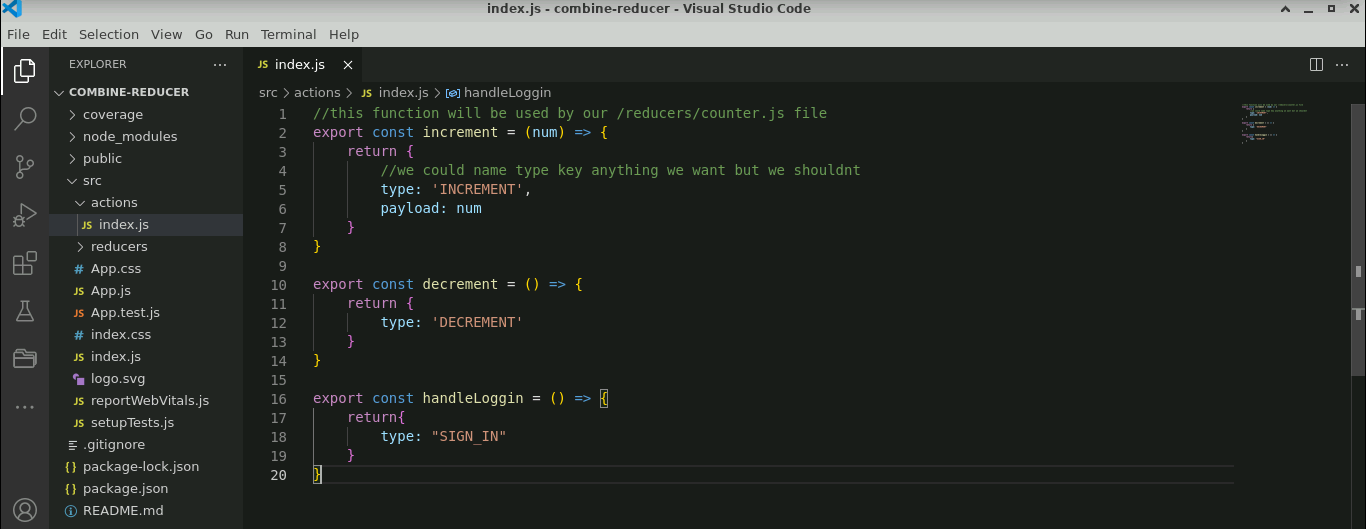
**export const handleLoggin = () => {**

**return{**

**type: "SIGN\_IN"**

**}**

**}**



1. Inside the **reducers** folder, create a file named **counter.js** and enter the following code:

**// counter REDUCER**

**//takes care of all the action related to our counter**

**// this will take 2 argument**

**const counterReducer = (state = 0, action) => {**

**switch(action.type){**

**case 'INCREMENT':**

**return state + action.payload**

**case 'DECREMENT':**

**return state - 1**

**default:**

**return state**

**}**

**}**

**export default counterReducer**

**A screenshot of a computer

Description automatically generated**

1. Create a file named **index.js** and enter the following code:

**import counterReducer from './couter'**

**import loggedReducer from './isLogged'**

**import {combineReducers} from 'redux'**

**const allReducers = combineReducers({**

**//we access this by any key we want**

**counter: counterReducer,**

**isLogged: loggedReducer**

**})**

**export default allReducers**

**A screenshot of a computer

Description automatically generated**

1. Create a file named **isLogged.js** and enter the following code:

**const loggedReducer = (state=false, action) =>{**

**switch(action.type){**

**case 'SIGN\_IN':**

**return !state;**

**default:**

**return state;**

**}**

**}**

**export default loggedReducer**

**A screenshot of a computer

Description automatically generated**

**Step 3: Create an App.js and index.js file**

1. Create an **App.js file** and enter the following code:

**import React, { useState } from 'react';**

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

**import {increment,decrement,handleLoggin} from './actions'**

**import './App.css';**

**function App() {**

**//next line will allow us to grab the state inside counter. So it goes to /reducers/index.js**

**//find the counter key in allReducers function, which gets it from /reducers/counter.js where the**

**//state and counterReducer function reside, now depends on the action type being used**

**let [buttonValue,setButtonValue]=useState("Login");**

**const counter = useSelector(state => state.counter)**

**const isLogged = useSelector(state => state.isLogged)**

**const dispatch = useDispatch()**

**return (**

**<div className="container">**

**<h1>Counter {counter}</h1>**

**<button onClick={()=> dispatch(increment(5))} disabled={!isLogged} className='btn btn-success'>+</button>**

**<button onClick={()=> dispatch(decrement())} disabled={!isLogged} className='btn btn-success'>-</button>**

**<div>**

**<input type="button" onClick={()=> {**

**dispatch(handleLoggin())**

**isLogged ?setButtonValue("Login"):setButtonValue("Logout")**

**}} value={buttonValue} className='btn btn-primary'/>**

**</div>**

**{isLogged ? <div><h1>Only show when loggin is true</h1></div> : ''}**

**</div>**

**);**

**}**

**export default App;**

**A screen shot of a computer screen

Description automatically generated**

1. Create an **index.js file** and enter the following code:

**import React from 'react';**

**import ReactDOM from 'react-dom/client';**

**import './index.css';**

**import App from './App';**

**import reportWebVitals from './reportWebVitals';**

**import {legacy\_createStore as createStore} from 'redux'**

**import allReducer from './reducers'**

**import { Provider } from 'react-redux'**

**const store = createStore(allReducer);**

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

**root.render(**

**<React.StrictMode>**

**<Provider store={store}>**

**<App />**

**</Provider>**

**</React.StrictMode>**

**);**

**// If you want to start measuring performance in your app, pass a function**

**// to log results (for example: reportWebVitals(console.log))**

**// or send to an analytics endpoint. Learn more:** [**https://bit.ly/CRA-vitals**](https://bit.ly/CRA-vitals)

**reportWebVitals();**

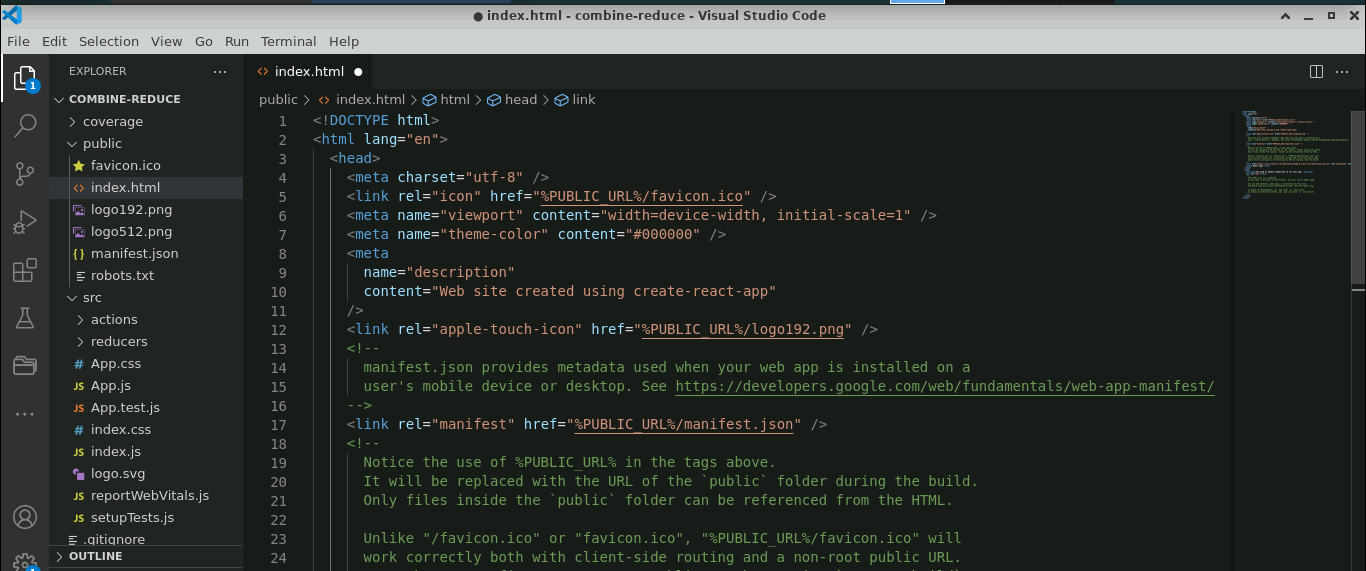
**A screen shot of a computer

Description automatically generated**

**Step 4: Add Bootstrap features in the index.html file**

1. Inside the **index.html** file, copy and paste the Bootstrap URL as shown below:

**<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">**

****

**A screenshot of a computer screen

Description automatically generated**

**Step 5: Test the application**

* 1. Open the terminal and execute the application by running the following command:

**npm start**

**A screen shot of a computer

Description automatically generated**

The output appears as shown below:

A screenshot of a computer

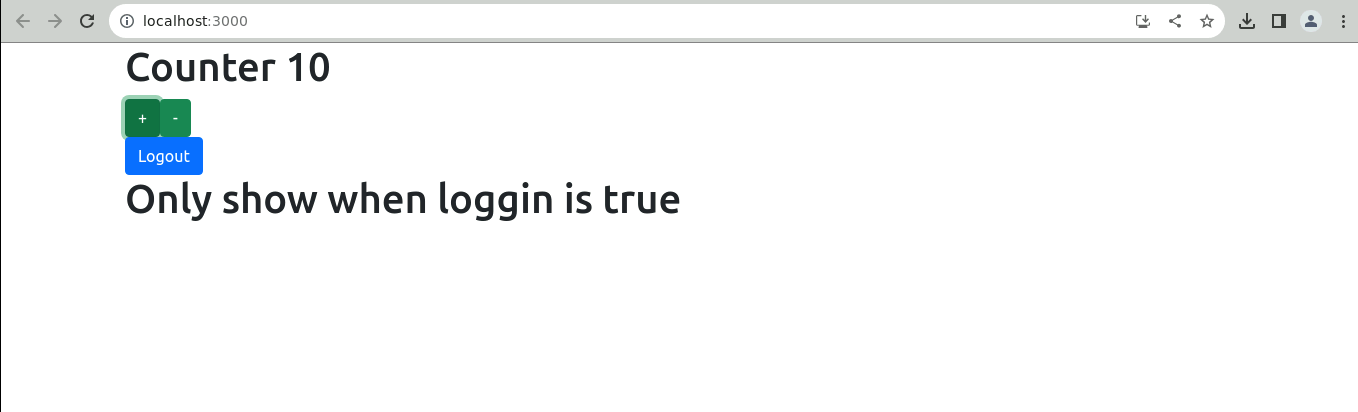
Description automatically generated

* 1. Click on the **Login** button (refer to the above screenshot)

A screenshot of a computer

Description automatically generated

* 1. Click on the **+** button to increment the values



* 1. Click on the‘-‘button to decrement the values

A screenshot of a computer

Description automatically generated

* 1. Click on the **Logout** button to logout from the page

A screenshot of a computer

Description automatically generated

You have now successfully created a React application with Redux, combining two reducers in one store to perform the task based on the reducer for state management.