**Lesson 04 Demo 03**

**Creating a React with Redux Counter Application**

**Objective:** To create a React application with Redux to access the counter value in each component and perform the operations on counter variables like increment and decrement

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

**Prerequisites:** None

Steps to be followed:

1. Create and set up the React project
2. Configure the Redux
3. Create a user-defined component
4. Test the application

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

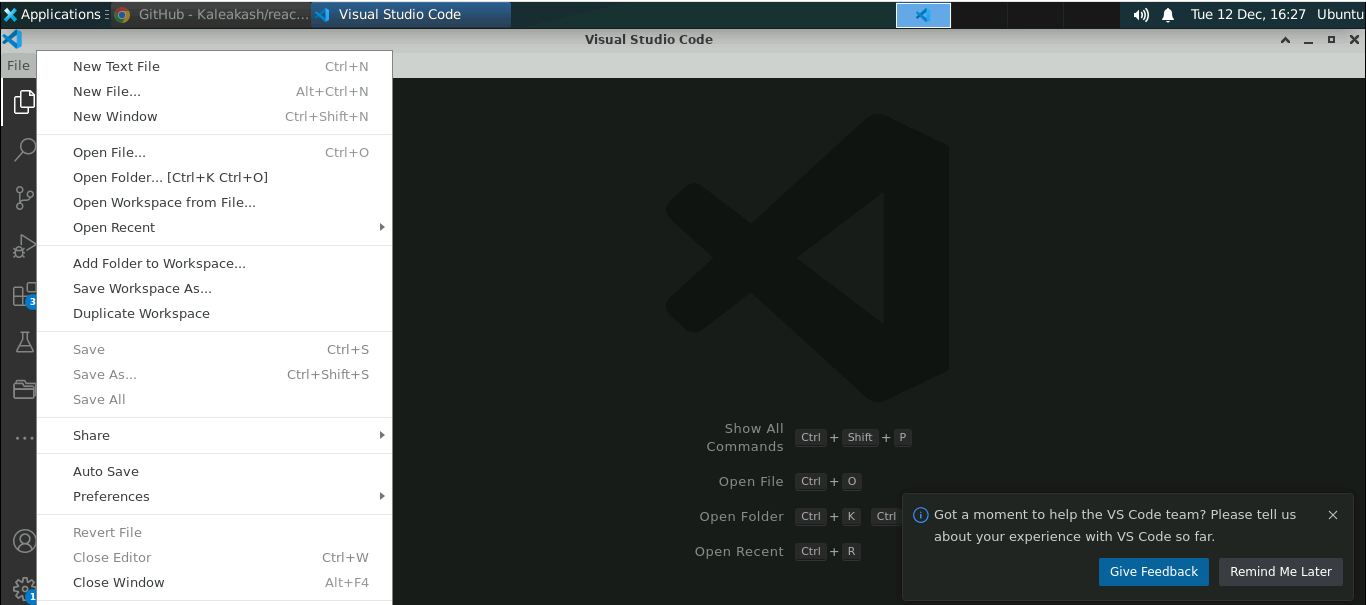
* 1. Open a terminal window and run the following command to create a React application:

**npx create-react-app** **react-redux-simple-counter**

**A screenshot of a computer

Description automatically generated**

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

****

* 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**

**Note**: This command helps to install all the required dependencies mentioned in the **package.json** file in the local machine on the form of a **node\_module** folder.

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

A screenshot of a computer

Description automatically generated

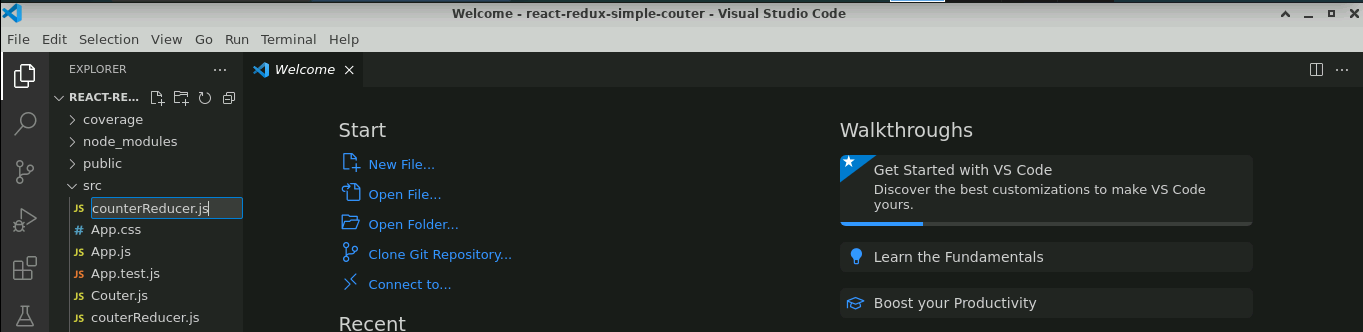
**Step 2: Configure the Redux**

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

**A screenshot of a computer

Description automatically generated**

1. Create a file named **counterReducer.js**

****

1. In the **counterReducer.js** file, enter the following code

**// Action Type**

**const INCREMENT = 'INCREMENT';**

**const DECREMENT = 'DECREMENT';**

**// Reducer**

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

**switch (action.type) {**

**case INCREMENT:**

**return state + 1;**

**case DECREMENT: return state - 1;**

**default:**

**return state;**

**}**

**};**

A screenshot of a computer

Description automatically generated

1. Inside the **src** folder, create a **counterStore.js** file and enter the following code to connect the reducer to a Redux store to make the state variable a global variable.

**// Store**

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

**import { counterReducer } from './couterReducer';**

**export const couterStore = createStore(counterReducer);**

A screenshot of a computer

Description automatically generated

1. Configure the **store** using the **Provider** in **index.js** with the help of 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 {couterStore} from './couterStore';**

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

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

**root.render(**

**<React.StrictMode>**

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

**<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

**reportWebVitals();**

**A screen shot of a computer program

Description automatically generated**

**Step 3:** **Create a user-defined component**

1. Inside the **src** folder, create a **counter.js** file and enter the following code to access the global state with **useSelector**.

**import {useSelector} from 'react-redux';**

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

**const data = useSelector(state => state);**

**return(**

**<div>**

**<h3>Counter Component</h3>**

**<p>Counter value in Counter component {data}</p>**

**</div>**

**)**

**}**

**A screenshot of a computer

Description automatically generated**

1. Inside the **src** folder, create an **IncrementCounter.js** file and enter the following code to dispatch actions using **useDispatch**.

**import {useDispatch} from 'react-redux';**

**// Action Type**

**const INCREMENT = 'INCREMENT';**

**// Actions**

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

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

**const dispatch = useDispatch();**

**return(**

**<div>**

**<h3>Increment Component</h3>**

**<input type="button" value="Increment" onClick={()=>dispatch(increment())}/>**

**</div>**

**)**

**}**

A screenshot of a computer

Description automatically generated

1. Now, inside the **src** folder, create the **DecrementCounter.js** file, and enter the following code to dispatch actions using **useDispatch**.

**import {useDispatch} from 'react-redux';**

**// Action Type**

**const DECREMENT = 'DECREMENT';**

**// Actions**

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

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

**const dispatch = useDispatch();**

**return(**

**<div>**

**<h3>Decrement Component</h3>**

**<input type="button" value="Decrement" onClick={()=>dispatch(decrement())}/>**

**</div>**

**)**

**}**

A screenshot of a computer program

Description automatically generated

**Step 4: Test the application**

1. In the **App.js** file, import **Counter**, **IncrementCounter,** and **DecrementCounter**

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

**import { Counter } from './Counter';**

**import { DecrementCounter } from './DecrementCounter';**

**import { IncrementCounter } from './IncrementCounter';**

**function App() {**

**return (**

**<div className="App">**

**<Counter></Counter>**

**<hr/>**

**<IncrementCounter></IncrementCounter>**

**<DecrementCounter></DecrementCounter>**

**</div>**

**);**

**}**

**export default App;**

A screen shot of a computer

Description automatically generated

1. Open the terminal and run the below command to execute the application:

**npm start**

A screen shot of a computer

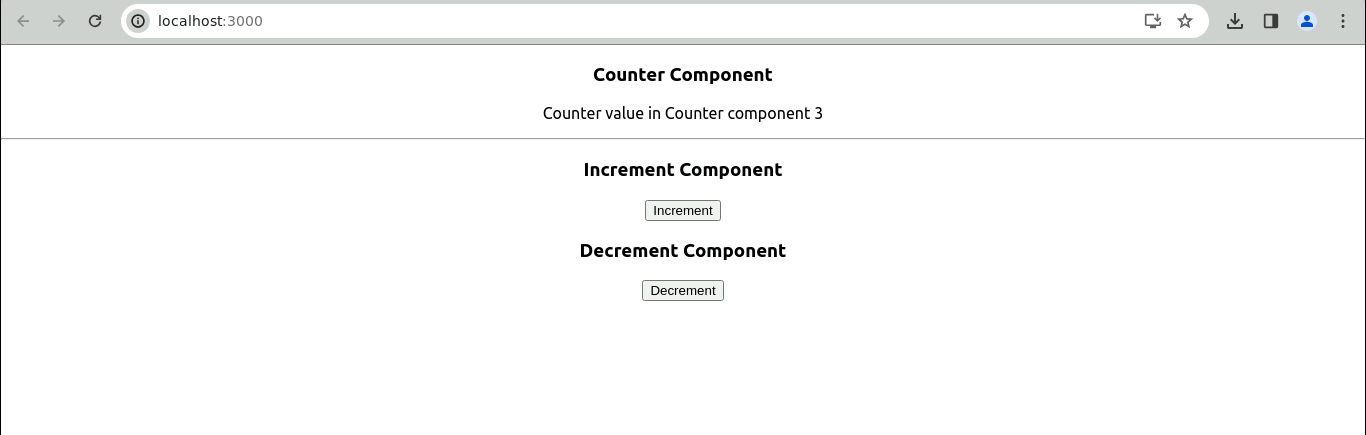
Description automatically generated

The output appears as shown below:

A screenshot of a computer

Description automatically generated

1. Under **Increment Component**, click on the **Increment** button to increment the counter value



1. Under **Decrement Component**, click on the **Decrement** button to decrement the counter value

A screenshot of a computer

Description automatically generated

With this, you have successfully created a React application with Redux to access the counter value in each component and perform the operation on counter variables like increment and decrement.