**Lesson 01 Demo 01**

**Creating a Simple React Custom Hook**

**Objective:** To create a simple React custom Hook application to demonstrate the basic click operation

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

**Prerequisites:** None

Steps to be followed:

1. Create and set up the React project
2. Create a custom Hook file
3. Use a custom Hook in user-defined components
4. Import user-defined functional components in the App.js file
5. Run the application

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

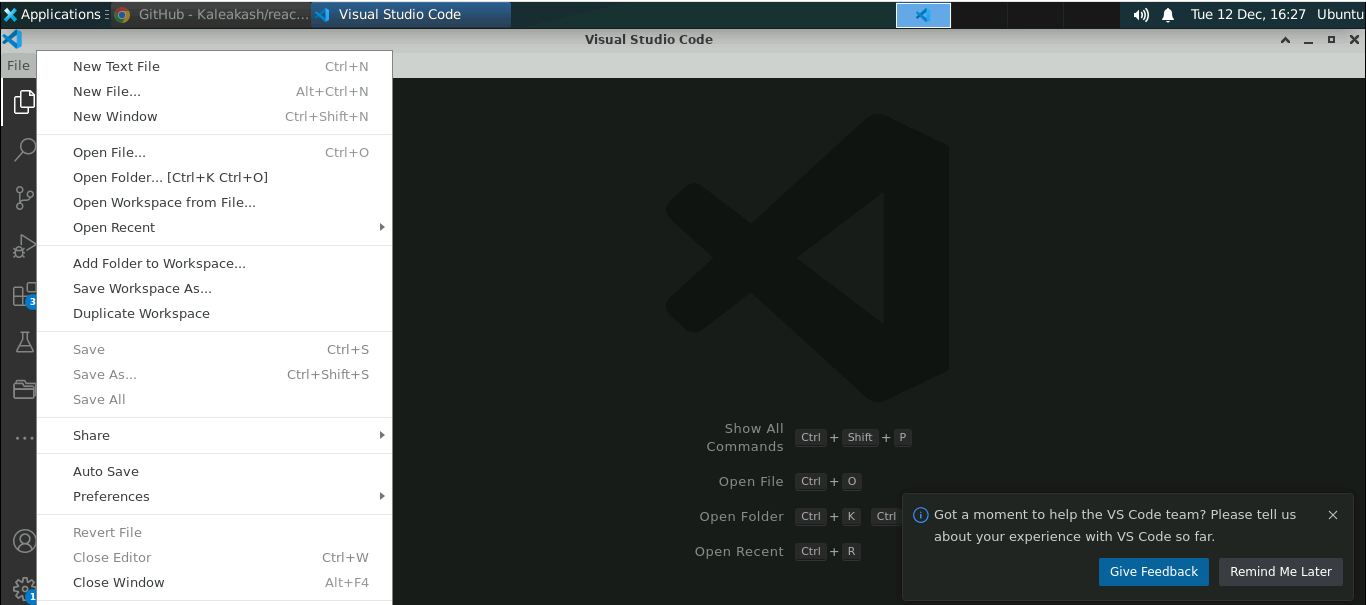
1. Open a terminal window to run the following command to create a React application:

**npx create-react-app simple-custom-hook**

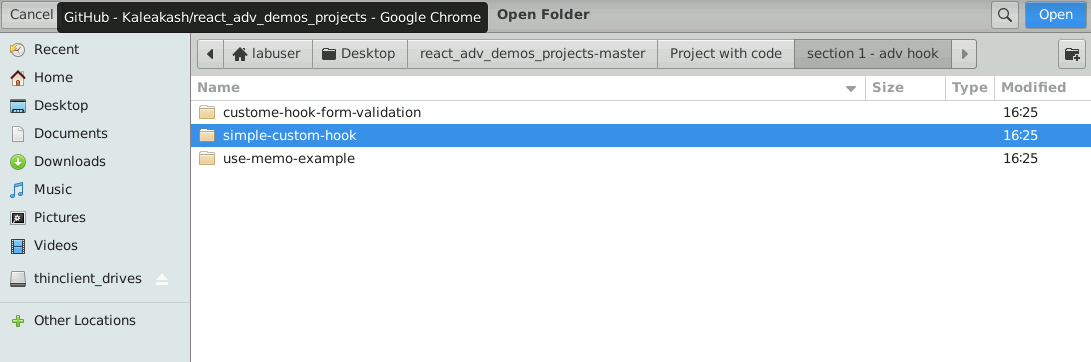
**A screenshot of a computer

Description automatically generated**

1. Open the created React application folder (**simple-custom-hook**) in Visual Studio Code (VS Code) by clicking on **File** in the top left corner and selecting **Open Folder**

****

1. Click on **Open** button

****

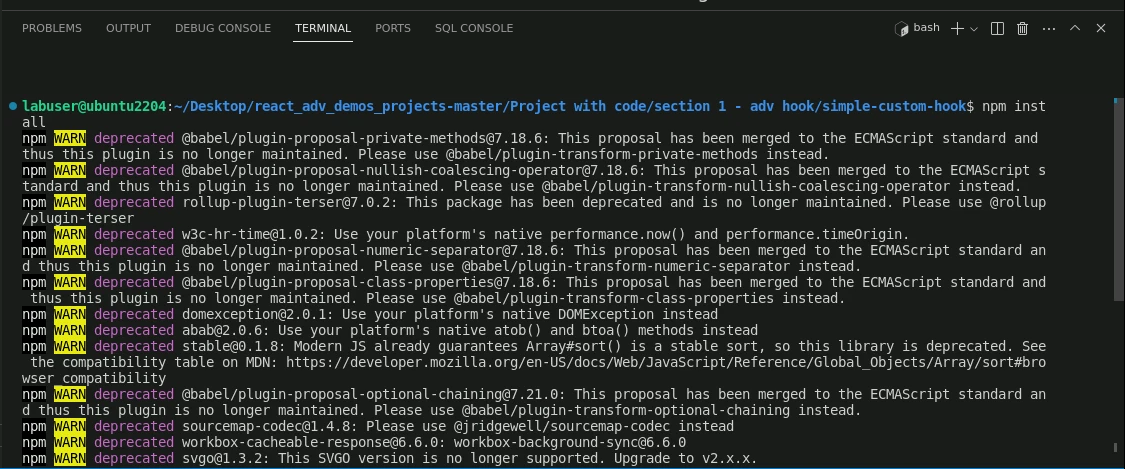
The folder structure appears as follows:

A screenshot of a black screen

Description automatically generated

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

**npm install**

****

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

**Step 2: Create a custom Hook file**

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

**A screenshot of a computer

Description automatically generated**

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

**A screenshot of a computer

Description automatically generated**

* 1. In the **userCounter.js** file enter the following code. It consists of logic to create the custom Hook.

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

**function useCounter(intialCount = 0){**

**const [count, setCount] = useState(intialCount);**

**// Similar to componentDidMount and componentDidUpdate:**

**useEffect(() => {**

**// Update the document title using the browser API**

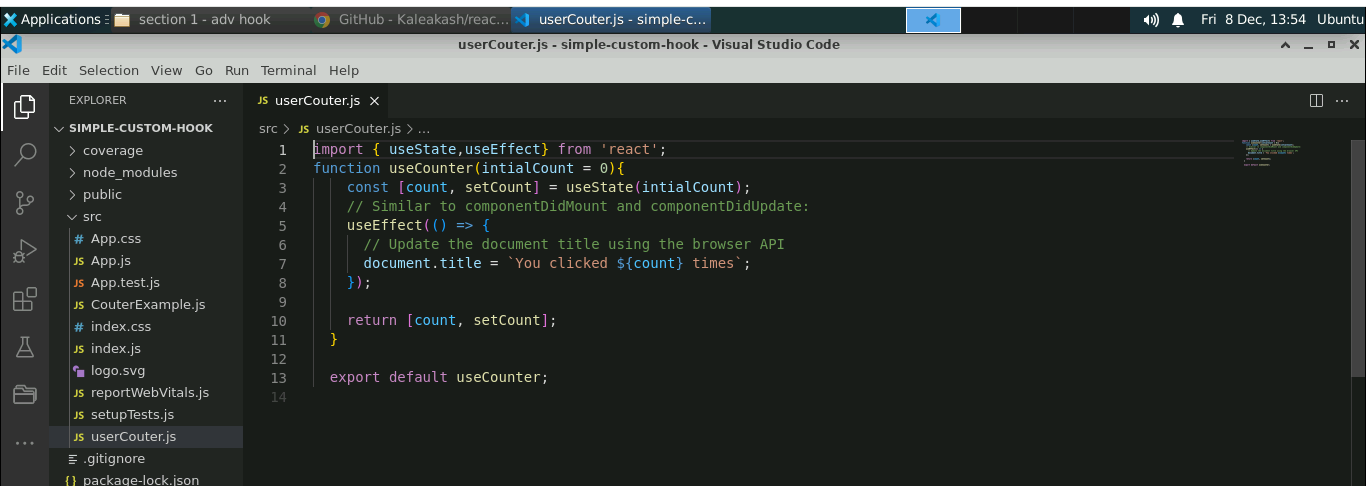
**document.title = `You clicked ${count} times`;**

**});**

**return [count, setCount];**

**}**

**export default useCounter;**



**Step 3: Use a custom Hook in user-defined components**

1. Inside the **src** folder, create a **CouterExample.js** file and enter the following code:

**import useCounter from "./userCounter";**

**function CouterExample() {**

**// Call custom hook `useCounter` to reuse Counter logic**

**const [count, setCount] = useCounter(0);**

**return (**

**<div>**

**<p>You clicked {count} times</p>**

**<button onClick={() => setCount(count + 1)}>**

**Click me**

**</button>**

**</div>**

**);**

**}**

**export default CouterExample;**

A computer screen shot of a computer code

Description automatically generated

**Step 4: Import user-defined functional component in the App.js file**

1. Inside the **App.js** file, import the user-defined component **CounterExample**

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

**import CouterExample from './CouterExample';**

**function App() {**

**return (**

**<div>**

**<h2>Simple React Custom Hook Example </h2>**

**<CouterExample></CouterExample>**

**</div>**

**);**

**}**

**export default App;**

A screenshot of a computer

Description automatically generated

**Step 5: Run the application**

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

**npm start**

A screen shot of a computer

Description automatically generated

**Note**: The project runs on the default browser with the URL [**http://localhost:3000**](http://localhost:3000)

The output appears as shown below:

A blue and white rectangular object

Description automatically generated

1. Now, click on **Click me** button to see the value increment

A white rectangular object with a black border

Description automatically generated

With this, you have successfully created a simple React custom Hook application to demonstrate the basic click operation.