

**Lab Manual- React useState and useEffect**

**Prepared for**:

**Date:** 18th June 2024

**Prepared by:**

Document Name: Lab Manual **Document Number** SDLab333

**Contributor:**

Contents

[1. Objective 3](#_Toc172110393)

[2. Create new File Hookdemo1.js 3](#_Toc172110394)

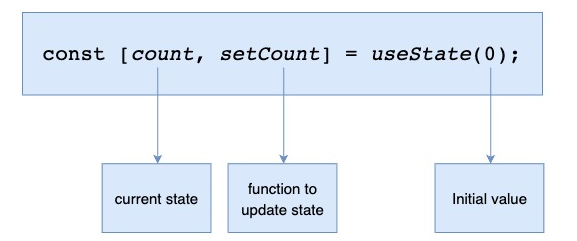
[3. Create new File Hookdemo2.js ( useEffect) 5](#_Toc172110395)

# Objective

React is a free, open-source JavaScript frontend library that you can use to build user interfaces by breaking up your project into components. It was released in 2013 with two significant ways of defining components: classes and functions.

Before React v16.8 in 2019, which included React hooks, developers have always had to use class components for data management (**with states**) and some additional operations (like **lifecycle methods**), relegating functional components to only be used for rendering UI.

Since the introduction of React Hooks in **React v16.8**, you can now manage data via states in functional components and work with lifecycle methods. This has led to many people adopting functional components over class components due to their cleaner syntax.



# Create new File Hookdemo1.js

* Create a new **Hookdemo.js** file and write following code

import React, { useState} from "react";

const Hookdemo1 =() => {

    const [counter , setCounter] = useState(0);

    const handleClick = () => setCounter (counter+1);

    return (

        <div>

        <button style = {{ marginTop: '50px'}} onClick={handleClick}>Click me</button>

        <h1>The Counter Increase</h1>

        <hr />

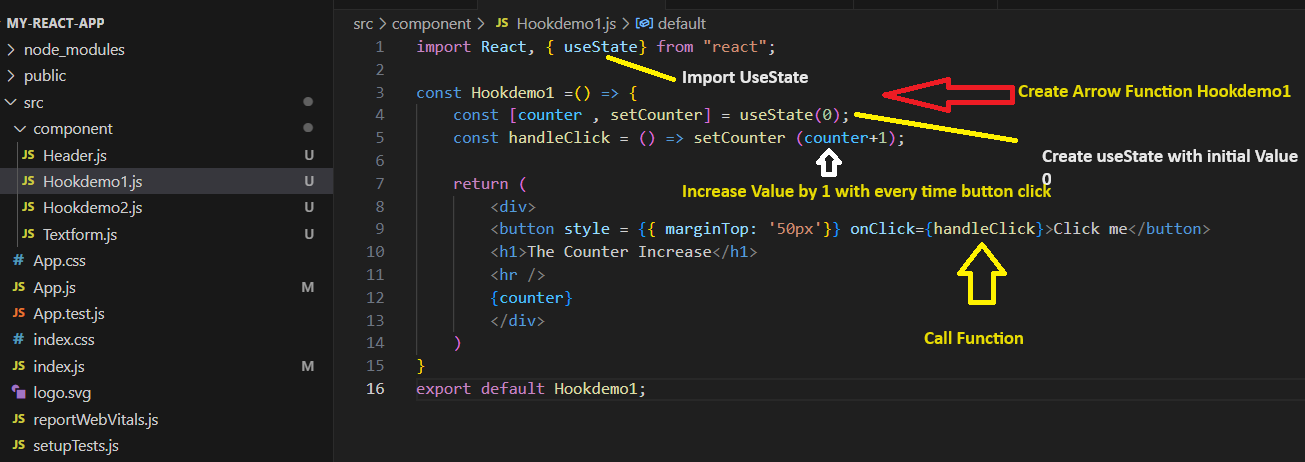
        {counter}

        </div>

    )

}

export default Hookdemo1;



* Now This Component in **App.js**

import Hookdemo1 from './component/Hookdemo1';

function App() {

  return (

    <div>

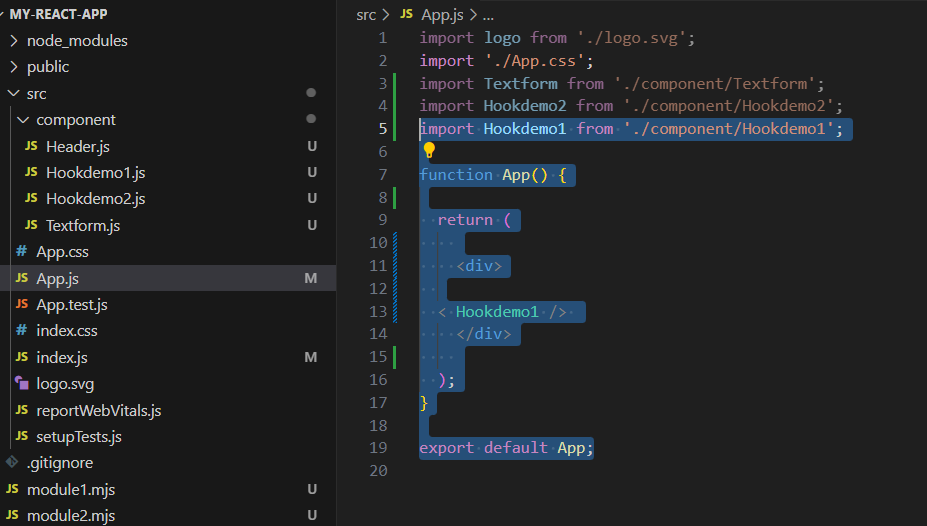
  < Hookdemo1 />

    </div>

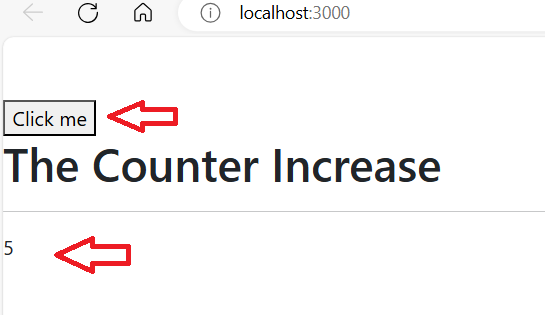
  );

}

export default App;



* Now Run and Compile it, every time when you click the **button** it will change the value



# Create new File Hookdemo2.js ( useEffect)

* Create New Files **Hookdemo2.js** and write below code

import React, { useEffect, useState} from "react";

const Hookdemo2 =() => {

    const [counter1 , setCounter1] = useState(0);

    const handleClick1 = () => setCounter1 (counter1+1);

    const [counter2 , setCounter2] = useState(0);

    const handleClick2 = () => setCounter2 (counter2+1);

    useEffect (() => {

        console.log ('inside effct')

    },[counter1]);

    return (

        <div>

        <button style = {{ marginTop: '50px'}} onClick={handleClick1}>Click me1</button>

        <h1>The Counter1 Increase</h1>

        <hr />

        {counter1}

        <button style = {{ marginTop: '50px'}} onClick={handleClick2}>Click me2</button>

        <h1>The Counter2 Increase</h1>

        <hr />

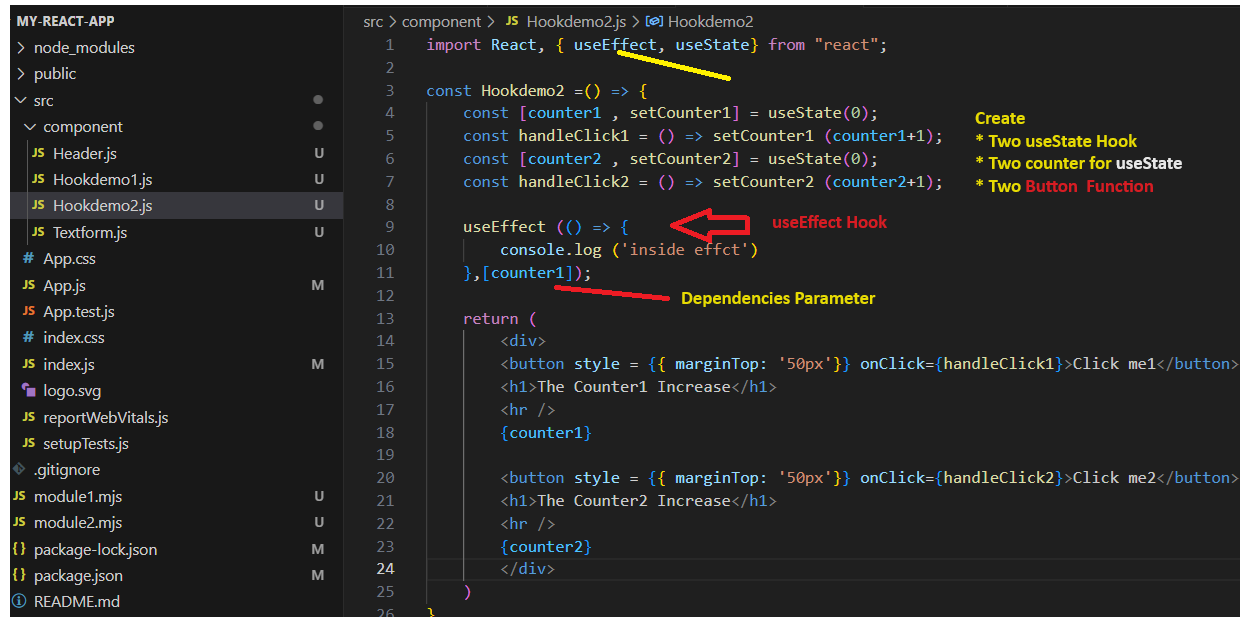
        {counter2}

        </div>

    )

}

export default Hookdemo2;



* Now Call Hookdemo2 component in **app.js**

import Hookdemo2 from './component/Hookdemo2';

import Hookdemo1 from './component/Hookdemo1';

function App() {

  return (

    <div>

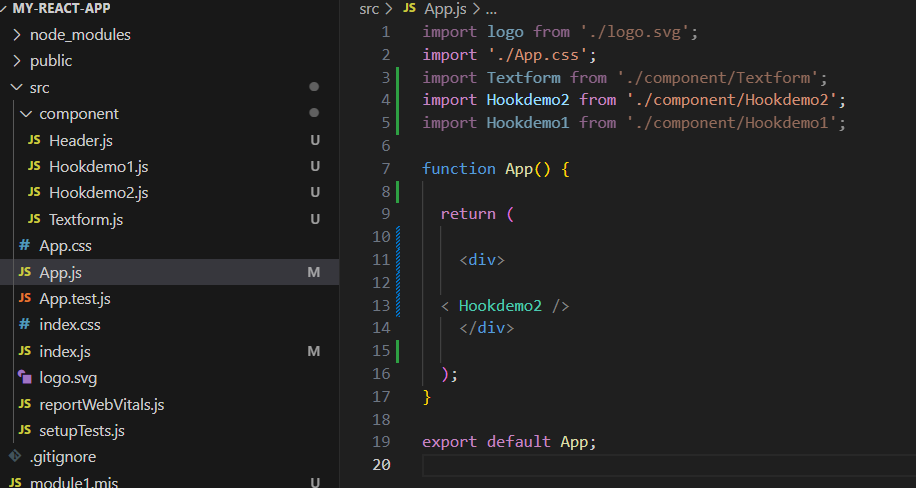
  < Hookdemo2 />

    </div>

  );

}

export default App;



Now Save and Recompile it

When you click **Click me** button , it **increase** the counter value and also in **console log** you can see no of time it effect

