**Lesson 03 Demo 01**

**Creating a React HOC**

**Objective:** To create the React higher-order component (HOC)

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

**Prerequisites:** None

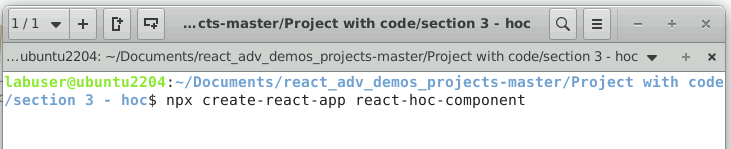
Steps to be followed:

1. Create and set up a React project
2. Create one user-defined component to demonstrate without the HOC concept
3. Create a HOC and user-defined components
4. Import all components in the App.js file
5. Test the application

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

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

**npx create-react-app react-hoc-component**



1. Open a terminal window inside a React project 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 on the local machine in the form of a **node\_module** folder.

1. Open the **react-hoc-example** folder in the VS Code editor

A screenshot of a computer

Description automatically generated

**Step 2: Create one user-defined component to demonstrate without HOC concept**

1. Create **TextComponentWithoutHOC.js** file, which takes properties as a parameter and returns some styling properties with JSX

**import { useState } from "react";**

**const TextComponentWithoutHOC = ({ text }) => {**

**const [isHovered, setHovered] = useState(false);**

**function handleMouseEnter() {**

**setHovered(true);**

**}**

**function handleMouseLeave() {**

**setHovered(false);}**

**return (**

**<><p**

**style={{ backgroundColor: isHovered ? "orange" : "white" }}**

**onMouseEnter={handleMouseEnter}**

**onMouseLeave={handleMouseLeave}**

**>**

**{ text }</p></>);**

**};**

**export default TextComponentWithoutHOC;**

A screenshot of a computer program

Description automatically generated

1. Import the component in **App.js** file

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

**import TextComponentWithoutHOC from './TextComponentWithoutHOC';**

**function App() {**

**return (**

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

**<TextComponentWithoutHOC text="This is simple text change style without HOC!"></TextComponentWithoutHOC></div>**

**);**

**}**

**export default App;**

A screenshot of a computer

Description automatically generated

1. Run the application in the terminal with the **npm start** command

A screenshot of a computer

Description automatically generated

The following screenshot shows the output on the browser window:

A screenshot of a computer

Description automatically generated

1. Hover on the text to see the styling effect as shown in the screenshot below:

A screenshot of a computer

Description automatically generated

**Step 3: Create a HOC and user-defined components**

1. Create a **TextComponent.js** file that takes properties as **text**, **isHovered** as Boolean, and returns some styling effect

**const TextComponent = ({ text, isHovered }) => {**

**return (**

**<>**

**<p style={{ backgroundColor: isHovered ? "orange" : "white" }}>{text}</p>**

**</>**

**);**

**};**

**export default TextComponent;**

**Note:** In this file, import all components as well as the HOC component, which takes two user-defined components, TextComponent and InputComponent, as parameters and returns a new component with added styling effects.

A screenshot of a computer

Description automatically generated

1. Create an **InputComponent.js** file that takes properties as a type, **isHovered** as a Boolean, and returns a styling effect

**const InputComponent = ({ type, isHovered }) => {**

**return (**

**<input**

**type={type}**

**style={{ backgroundColor: isHovered ? "orange" : "white" }}/>**

**);**

**};**

**export default InputComponent;**

A screenshot of a computer

Description automatically generated

**Step 4: Import all components in the App.js file**

1. Import all user-defined components in a **App.js** file, that is, **TextComponent** and **ButtonComponent**, and pass these components as parameters to the HOC component **withHoverHOC**

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

**import InputComponent from './InputComponent';**

**import TextComponent from './TextComponent';**

**import TextComponentWithoutHOC from './TextComponentWithoutHOC';**

**import withHoverHOC from './withHoverHOC';**

**// Creating components that contain hover logic using**

**// Higher Order Component**

**const TextComponentWithHover = withHoverHOC(TextComponent);**

**const InputComponentWithHover = withHoverHOC(InputComponent);**

**function App() {**

**return (**

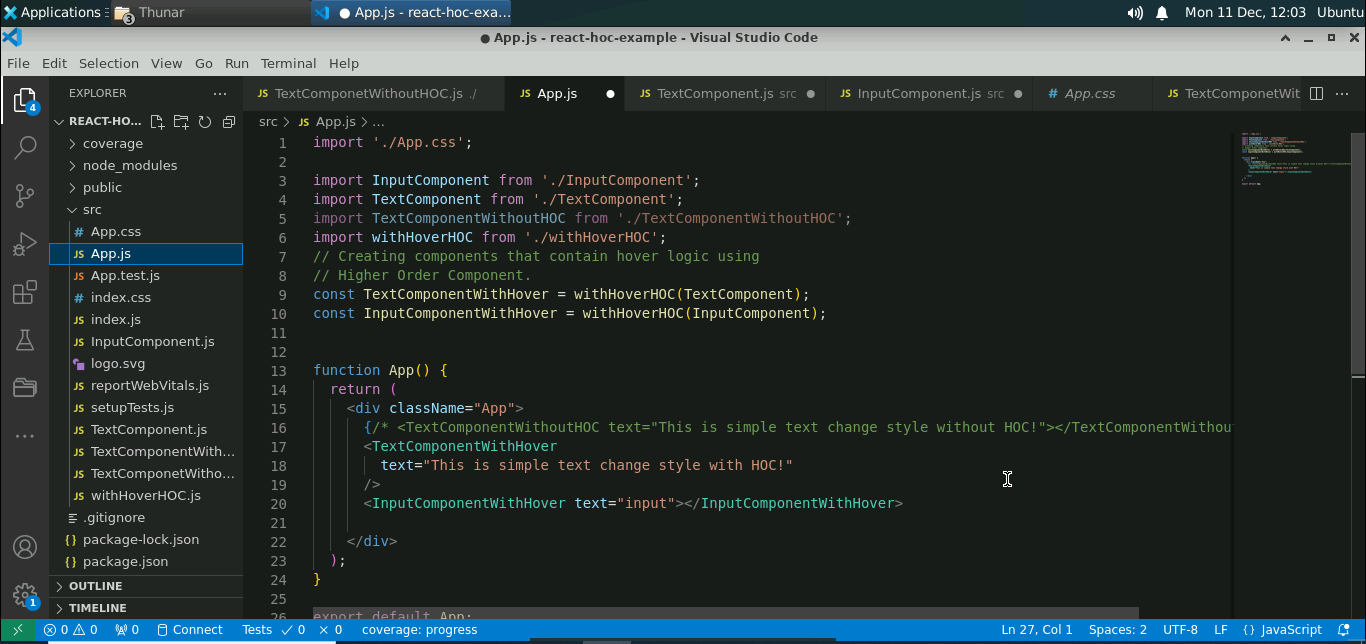
**<div className="App">**

**{/\* <TextComponentWithoutHOC text="This is simple text change style without HOC!"></TextComponentWithoutHOC> \*/}**

**<TextComponentWithHover text="This is simple text change style with HOC!"/>**

**<InputComponentWithHover text="input"></InputComponentWithHover></div>);}**

**export default App;**

****

**Step 5: Test the application**

1. Run the program with the **npm start** command

A screenshot of a computer

Description automatically generated

After successful compilation, the following screen will appear as an output in the browser:

A screenshot of a computer

Description automatically generated

1. Move the cursor on the text and text box to see the effect as shown below:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

With this, you have successfully created the React HOC.