**Lesson 03 Demo 01**

**React HOC authorization component**

**Objective:** To demonstrate the react hoc component with authorization features.

**Tools required:** Node JS and React JS

**Prerequisites:** HTML, CSS, JavaScript ES5/ES6, Basic React Concept

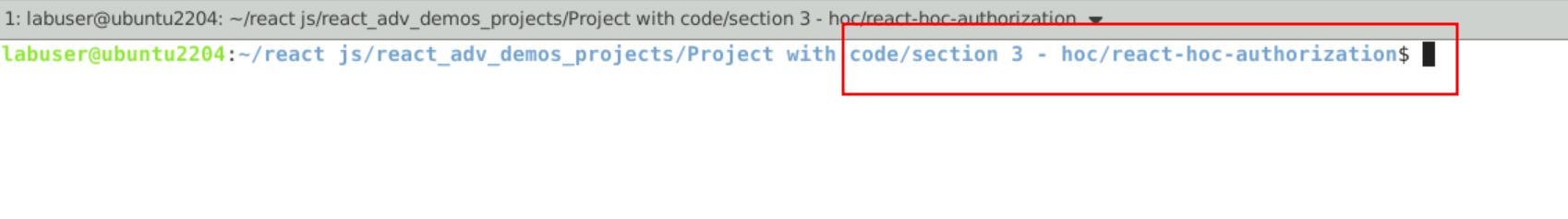
**Note** : All react js project already created with version 18.x with Sample App.js file

**Steps to be followed:**

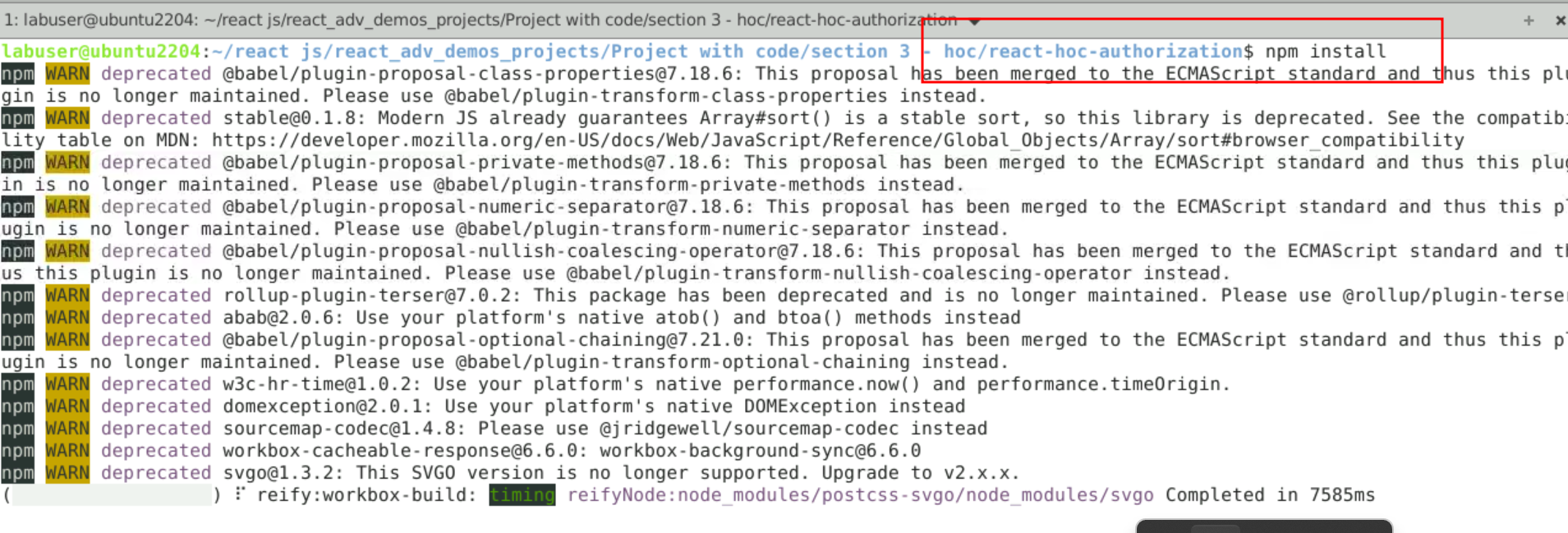
1. Set up for react js project
2. Create user defined components ie PrivateComponent, HOCComponent and checkPermission function.
3. Now we need to import these all component in App.js file.
4. Now test the application

**Step 1: Set up for react js project**

1. Open a terminal window inside a React JS pre-created project ie **react-hoc-example**

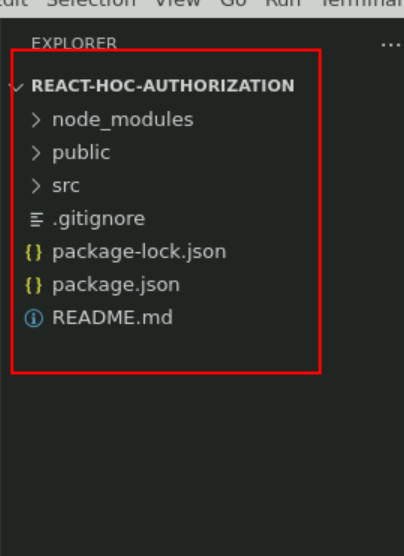


1. Now you need to run the command as **npm install.** This command helps us to installed all required dependencies mention in package.json file in local machine in the form of node\_module folder.



1. Now open **react-hoc-authorization** folder in VS Code Editor

Note: short cut to open write **code .**



**Step 2: Create user defined components ie PrivateComponent, HOCComponent and checkPermission function.**

2.1 Create file PrivateComponent.js file which return static data.

const PrivateComponent = () => {

return <div>This is a Private Component, only visible to Admin Users.</div>;

};

export default PrivateComponent;

2.2. Now create checkPermission.js file which takes props as parameter and check the rules and return the jsx.

const checkPermission = (props) => {

return props.userRole === "admin";

};

export default checkPermission;

2.3 Now we will create HOC component ie withAuthorization.js file. The which take two parameter ie component and check function and provide the new wrapper component which is responsible to provide new component or else message.

// Higher Order Component

function withAuthorization(WrappedComponent, checkPermissions) {

return function (props) {

// You can also wrap it in useEffect for async permission checks.

return checkPermissions(props) ? (

<WrappedComponent {...props} />

) : (

<p>Please login with appropriate role</p>

);

};

}

export default withAuthorization;

**Step 3: Now we need to import these all component in App.js file.**

3.1 In App.js file import all component and pass the Private component and checkPermission function as parameter to HOC components.

import { useState } from "react";

import PrivateComponent from "./PrivateComponent";

import checkPermission from "./checkPermission";

import withAuthorization from "./withAuthorization";

const PrivateComponentComponentWithAuthorization = withAuthorization(

PrivateComponent,

checkPermission

);

export default function App() {

let [role,setRole]=useState("");

return (

<div className="App">

<input type="text" value={role} onChange={(e)=>setRole(e.target.value)}/>

<br/>

<PrivateComponentComponentWithAuthorization userRole={role} />

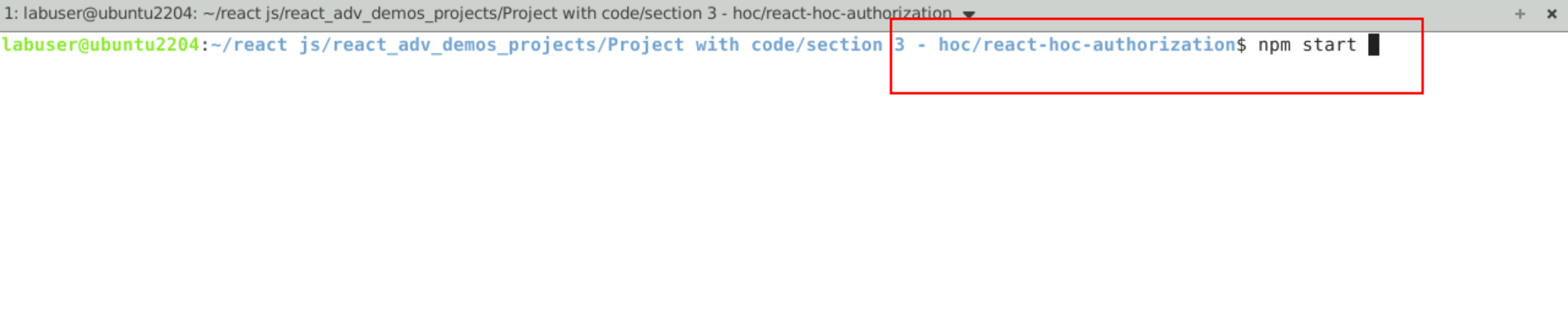
</div>

);

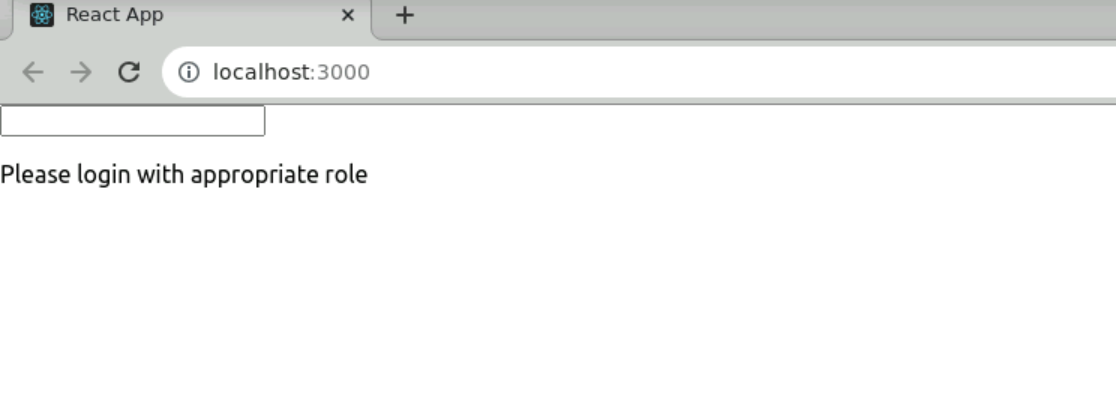
}

**Step 4. Now test the application**

**4.1 npm start**



**4.2 Now you can view the output on browser.**



**4.3 if rule is admin then only you can view the private components.**

