**Lesson 03 Demo 02**

**React HOC Component to handle asyn – data**

**Objective:** To demonstrate the react hoc component to handle aysn data. Those asynchronous data you can use your own rest api or fake rest api

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

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

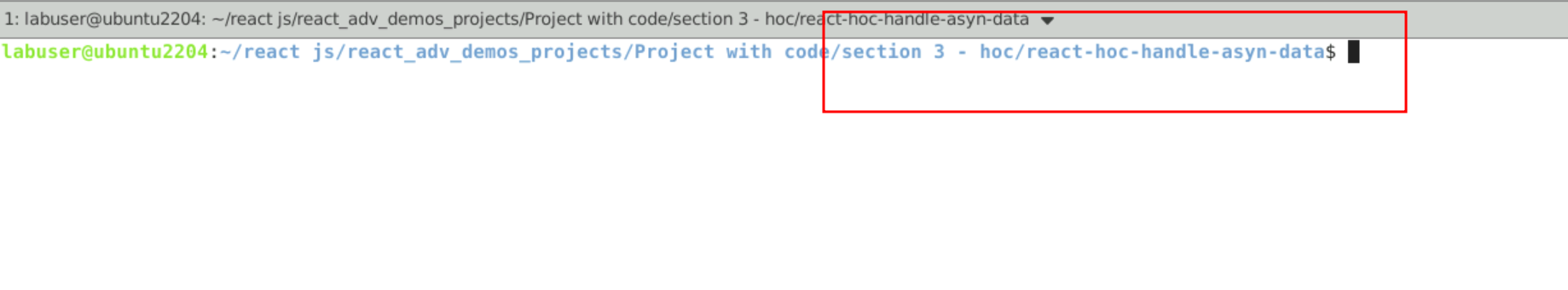
**Note** : All react js project already created with version 18.x with Sample App.js file

**Steps to be followed:**

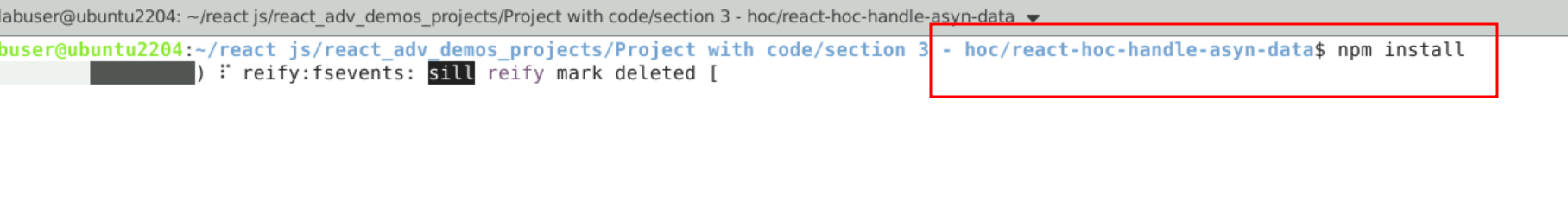
1. Set up for react js project
2. Create user defined function as well as components.
3. Now we need to import these all component in App.js file.
4. Now test the application

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

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

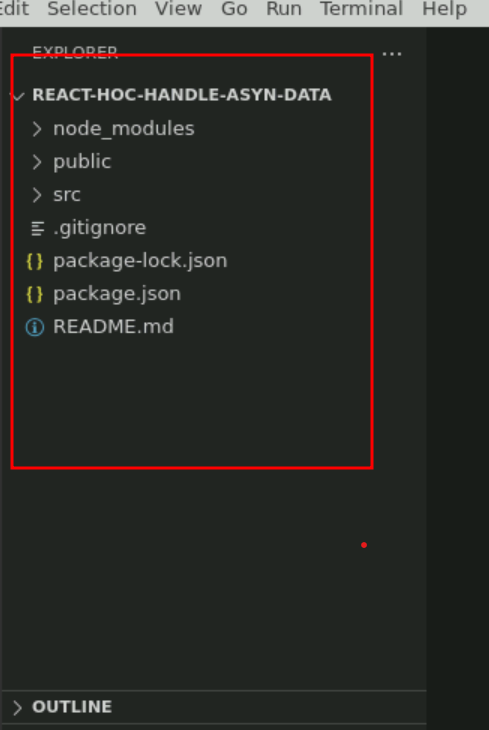


1. Now you need to run the command as **npm install.** This command helps us to installed all required dependencies mention in package.json file in local machine in the form of node\_module folder.



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

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

****

**Step 2: Create user defined function as well as components.**

2.1 Now create fetchProducts.js file. Which contains user defined function which responsible to consume rest api using axios or fetch functions.

const fetchProducts = async () => {

const response = await fetch("https://dummyjson.com/products");

const data = await response.json();

return data.products;

};

export default fetchProducts;

2.2. Now create **ProductComponent.js** file. This file create user defined component which is responsible to display the data in proper format.

const ProductComponent = ({ data, isLoading, error }) => {

if (isLoading) return <p>Loading...</p>;

if (error) return <p>Error: {error.message}</p>;

return (

<div style={{"display":"inline-block"}}>

{data.map((product) => (

<span>

<img src={product.thumbnail} width="100px" height="100px" style={{"border":"1px solid black","borderRadius":"30px"}}/>

</span>

))}

</div>

);

};

export default ProductComponent;

2.3 Now we will create HOC component ie withDataFetch.js file. The which take two parameter ie component and fetch function and provide the new wrapper component which is responsible to load the data from rest api.

import { useState, useEffect } from "react";

function withDataFetch(WrappedComponent, fetchData) {

return function (props) {

const [data, setData] = useState(null);

const [isLoading, setIsLoading] = useState(true);

const [error, setError] = useState(null);

useEffect(() => {

const fetchWrapper = async () => {

try {

const data = await fetchData(props);

setData(data);

setIsLoading(false);

} catch (error) {

setError(error);

setIsLoading(false);

}

};

fetchWrapper();

}, [props]);

return (

<WrappedComponent

{...props}

data={data}

isLoading={isLoading}

error={error}

/>

);

};

}

export default withDataFetch;

**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 Product component and fetch function as parameter to HOC components.

import ProductComponent from "./ProductComponent";

import fetchProducts from "./fetchProducts";

import withDataFetch from "./withDataFetch";

const PostComponentWithData = withDataFetch(ProductComponent, fetchProducts);

export default function App() {

return (

<div className="App">

<PostComponentWithData />

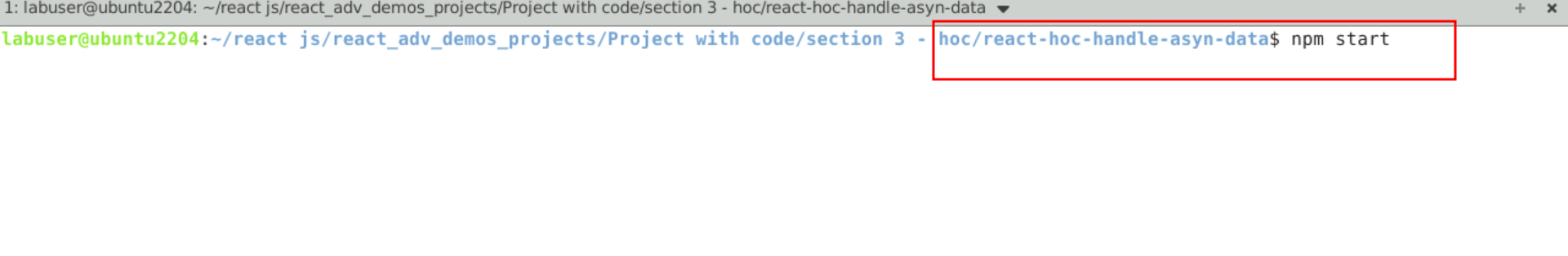
</div>

);

}

**Step 4. Now test the application**

**4.1 npm start**



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

