**Lesson 03 Demo 02**

**Creating a React HOC to Handle Asynchronous Data**

**Objective:** To create the React HOC to handle asynchronous data

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

**Prerequisites:** None

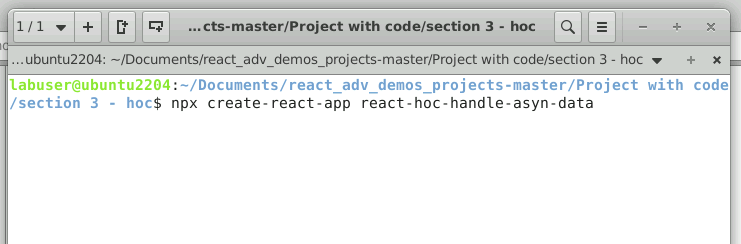
Steps to be followed:

1. Create and set up the React project
2. Create user-defined functions and components
3. Import components in the App.js file
4. Run the application

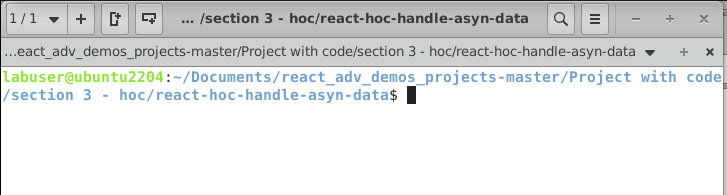
**Step 1: Create and set up the React project**

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

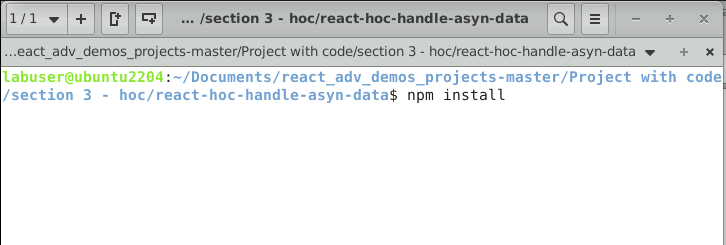
**npx create-react-app react-hoc-handle-asyn-data**

****

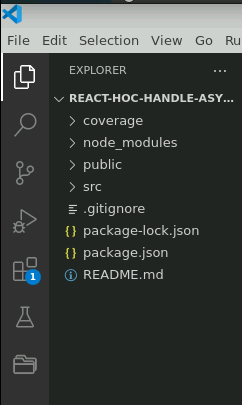
1. Open a terminal window inside a React project **react-hoc-handle-asyn-data**



1. Run the **npm install** command



1. Open the **REACT-HOC-HANDLE-ASYN-DATA** folder in VS code editor



**Step 2: Create user-defined functions and components**

1. Create a **fetchProducts.js** file that contains a user-defined function responsible for consuming the REST API using axios or fetch functions.

**const fetchProducts = async () => {**

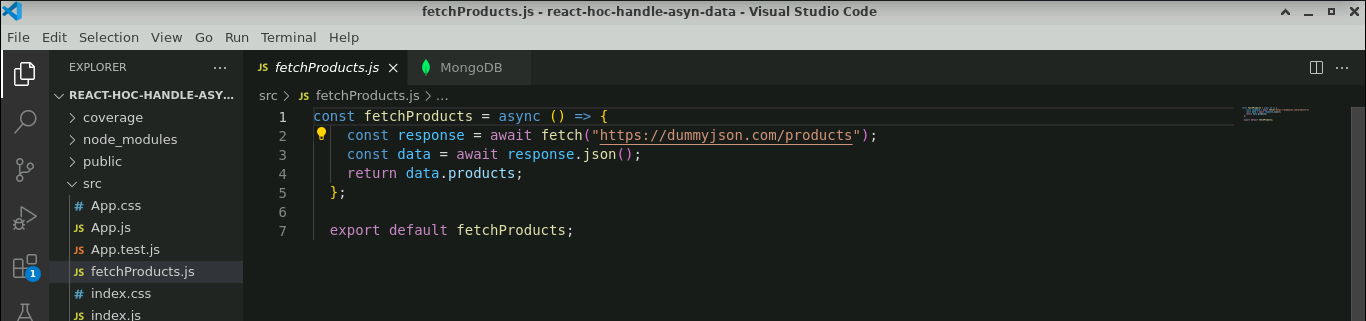
**const response = await fetch("https://dummyjson.com/products");**

**const data = await response.json();**

**return data.products;**

**};**

**export default fetchProducts;**

****

1. Create a **ProductComponent.js** file, which is used to create a user-defined component that is responsible for displaying the data in a 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"}}/>**

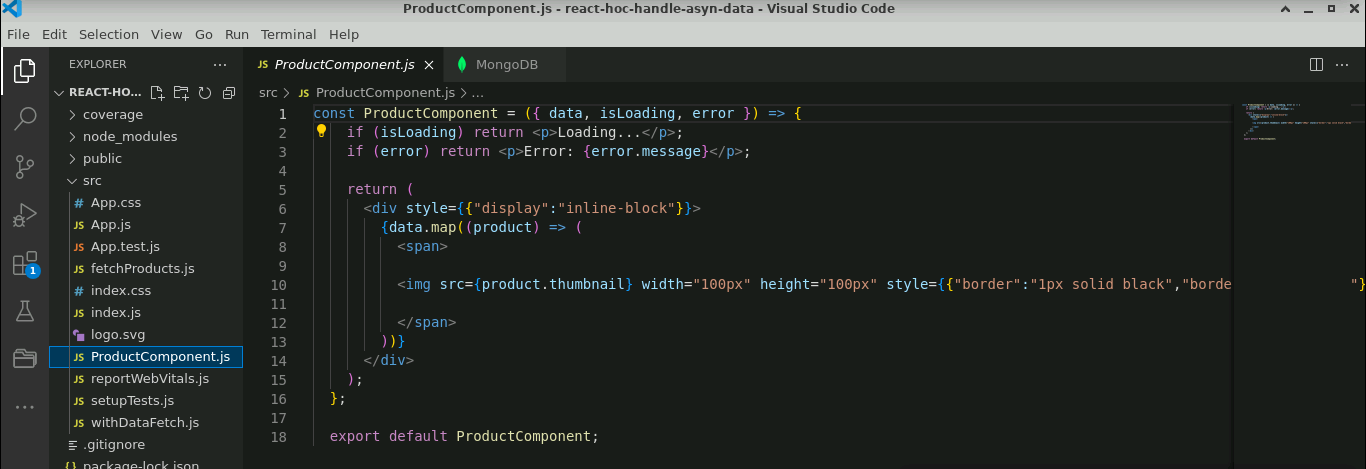
**</span>**

**))}**

**</div>**

**);};**

**export default ProductComponent;**



1. Create an HOC **withDataFetch.js** file that takes two parameters, the component, and fetch function, and provides the new wrapper component, which is responsible for loading the data from the 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}**

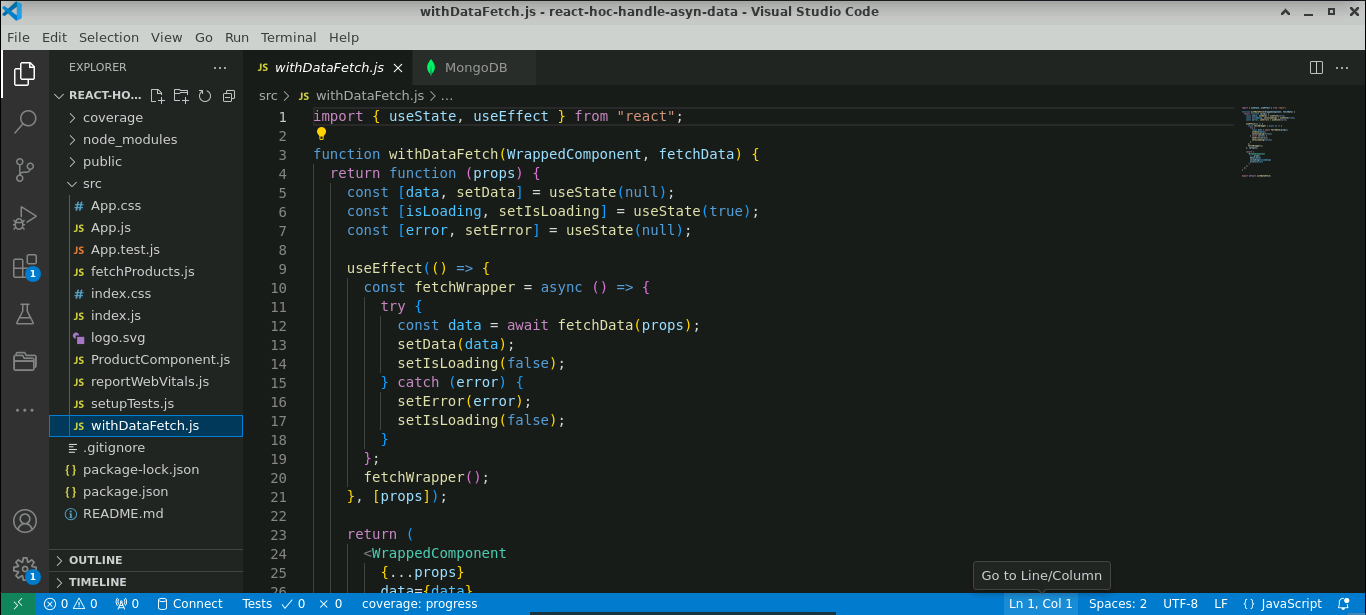
**error={error}/>**

**);**

**};**

**}**

**export default withDataFetch;**

****

**Step 3: Import components in the App.js file**

1. Import all components in the **App.js** file, pass the product component, and fetch the function as a 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 (**

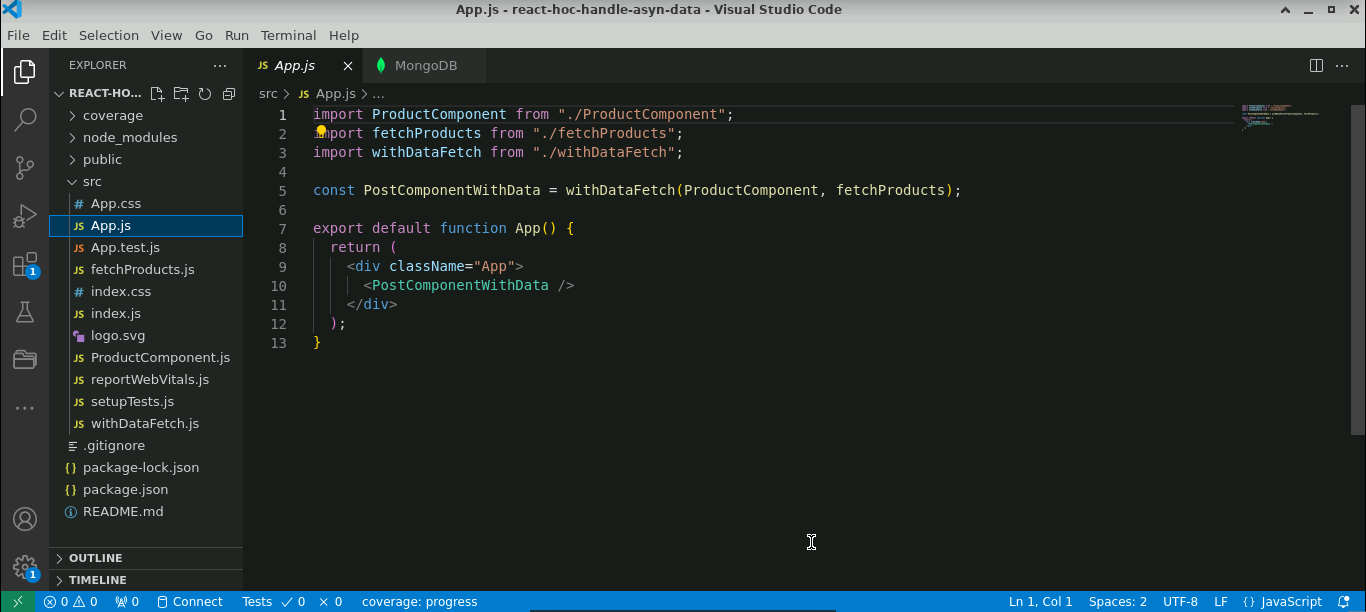
**<div className="App">**

**<PostComponentWithData />**

**</div>**

**);**

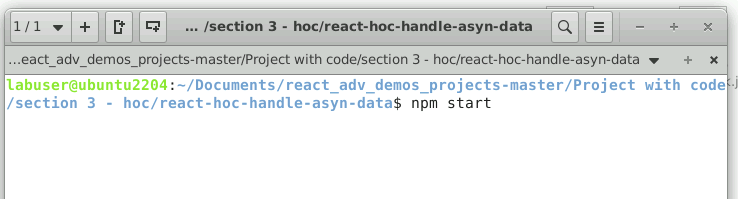
**}**



**Step 4: Run the application**

1. Run the application using the following command:

**npm start**

****

The output will look as follows:

****

With this, you have successfully created the React HOC to handle asynchronous data.