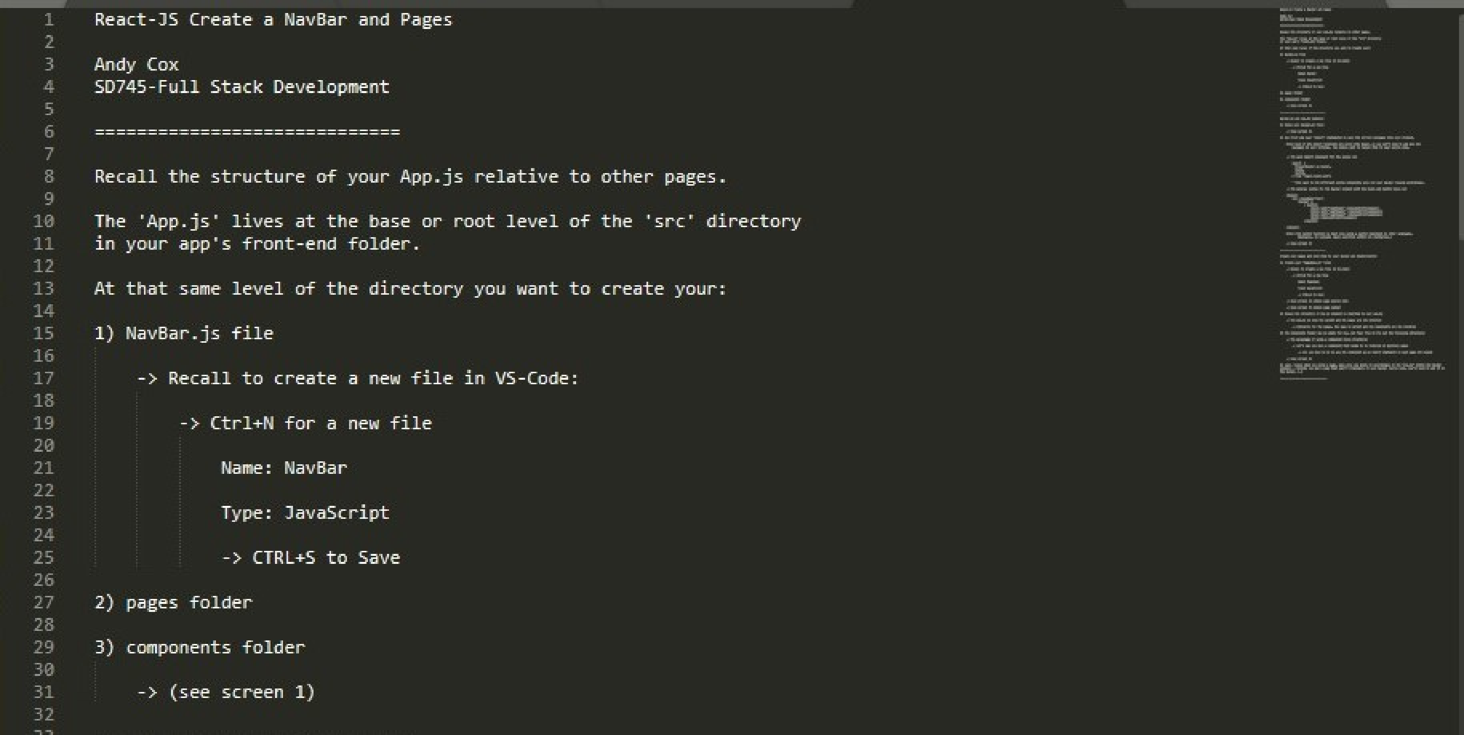
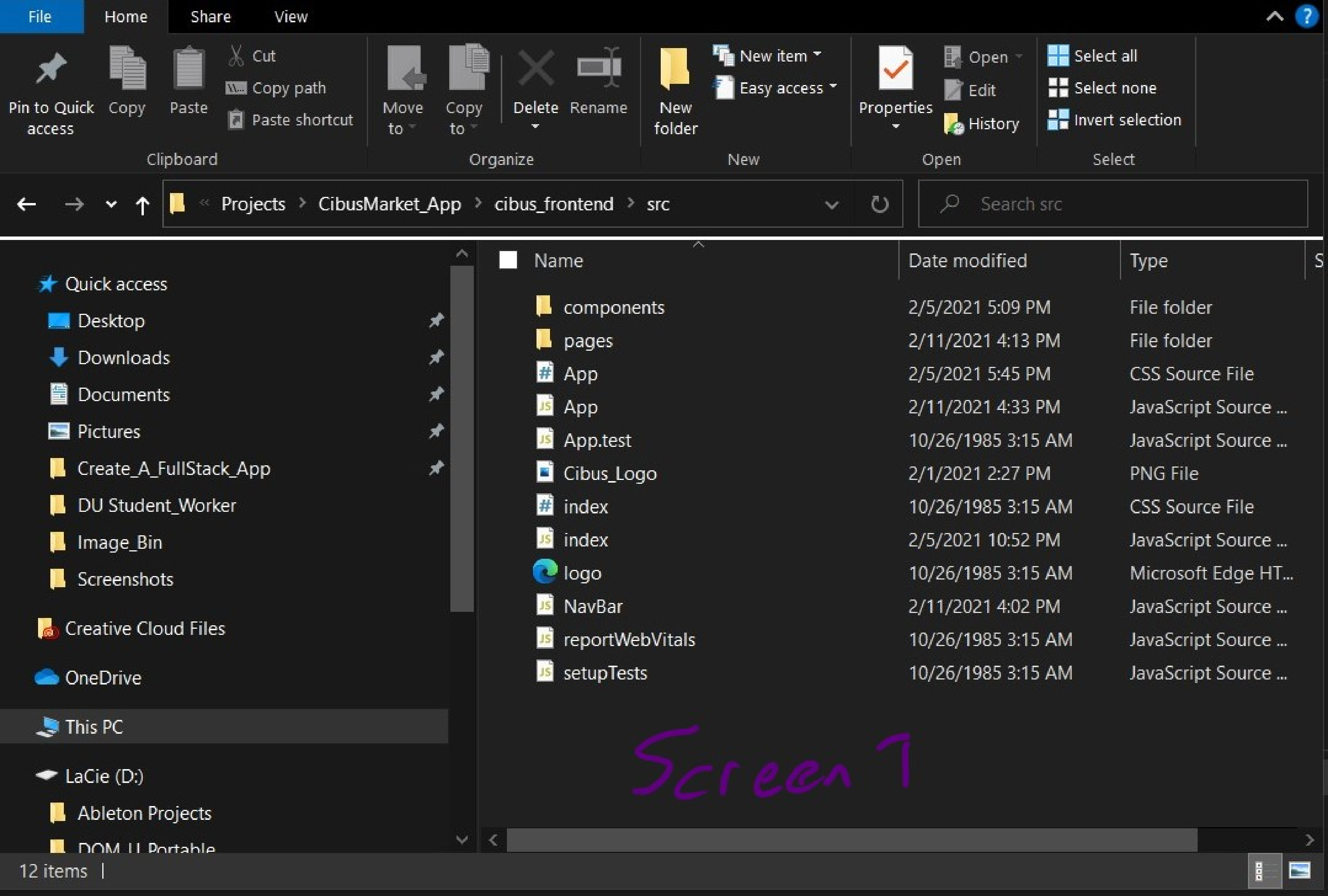


```
React-JS Create a NavBar and Pages

Andy Cox
SD745-Full Stack Development

=============================

Recall the structure of your App.js relative to other pages.

The 'App.js' lives at the base or root level of the 'src' directory
in your app's front-end folder.

At that same level of the directory you want to create your:

1) NavBar.js file

    -> Recall to create a new file in VS-Code:

        -> Ctrl+N for a new file

            Name: NavBar

            Type: JavaScript

            -> CTRL+S to Save

2) pages folder

3) components folder

    -> (see screen 1)
```



Name	Date modified	Type
components	2/5/2021 5:09 PM	File folder
pages	2/11/2021 4:13 PM	File folder
App	2/5/2021 5:45 PM	CSS Source File
App	2/11/2021 4:33 PM	JavaScript Source ...
App.test	10/26/1985 3:15 AM	JavaScript Source ...
Cibus_Logo	2/1/2021 2:27 PM	PNG File
index	10/26/1985 3:15 AM	CSS Source File
index	2/5/2021 10:52 PM	JavaScript Source ...
logo	10/26/1985 3:15 AM	Microsoft Edge HT...
NavBar	2/11/2021 4:02 PM	JavaScript Source ...
reportWebVitals	10/26/1985 3:15 AM	JavaScript Source ...
setupTests	10/26/1985 3:15 AM	JavaScript Source ...

Screen 1

Navbar.js and App.js Updates:

1) Setup your Navbar.js file:

    -> (see screen 2)

2) Now first add your 'import' statements to pull the correct packages into your project.

    Note: Most of the import functions are built into React, so you won't need to add any npm
        pacakges in your terminal. You simply want to recall them in your source code.

    -> The main import statement for the navbar is:

```
import  {
  BrowserRouter as Router,
  Route,
  Switch,
} from 'react-router-dom';
```

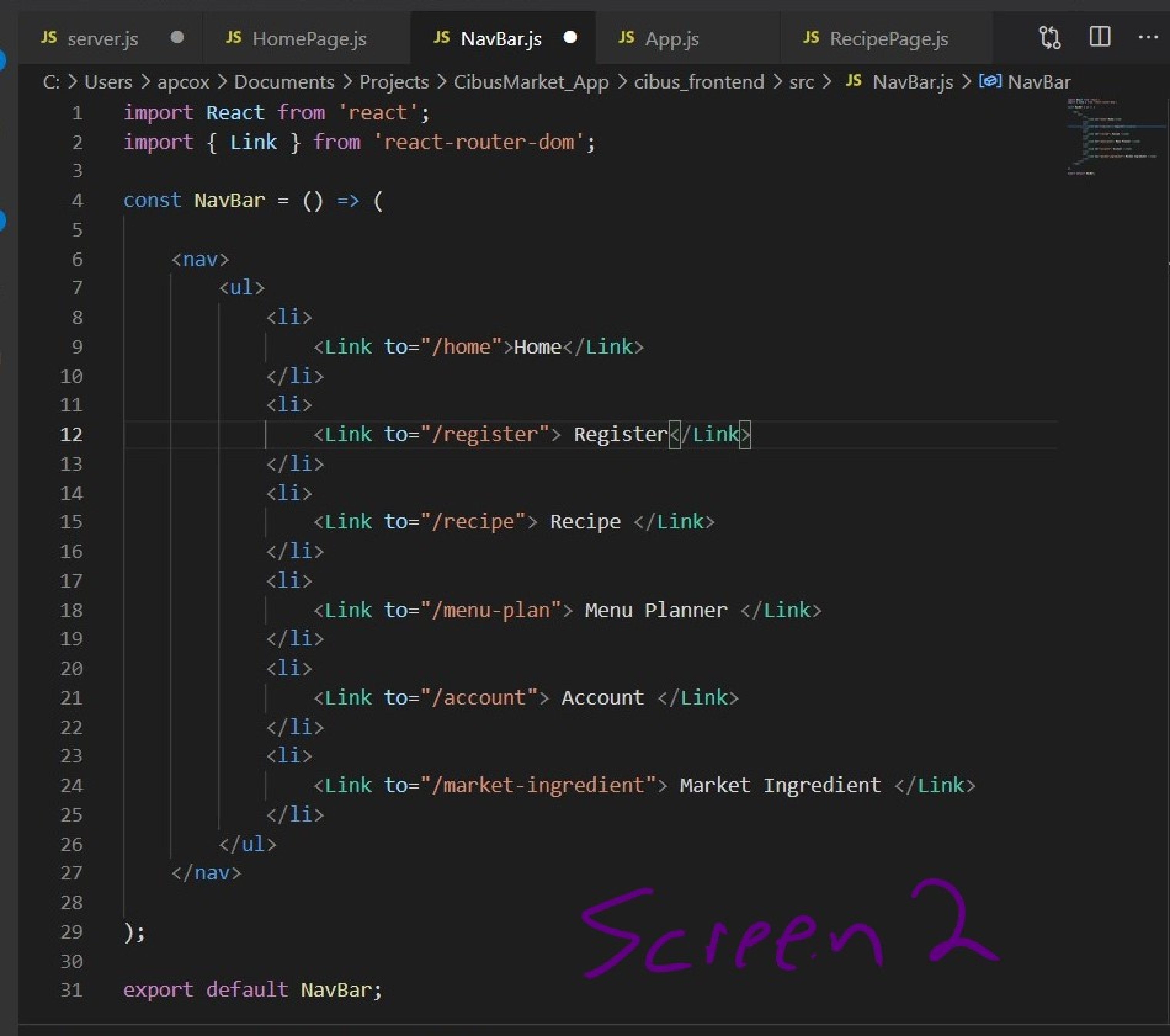
        ^ This call to the different system components will let your navbar respond accordingly.

    -> The general syntax for the Navbar element with the Route and Switch calls is:

```
<Router>
    <div className="App">
        <Navbar />
            < Switch>
                <Route path="/pathname" component={FileName}/>
                <Route path="/pathname1" component={FileName1}/>
                <Route path="/pathname2" component={FileName2}/>
                <Route component={NoFileName}/>
            </Switch>

</Router>
```

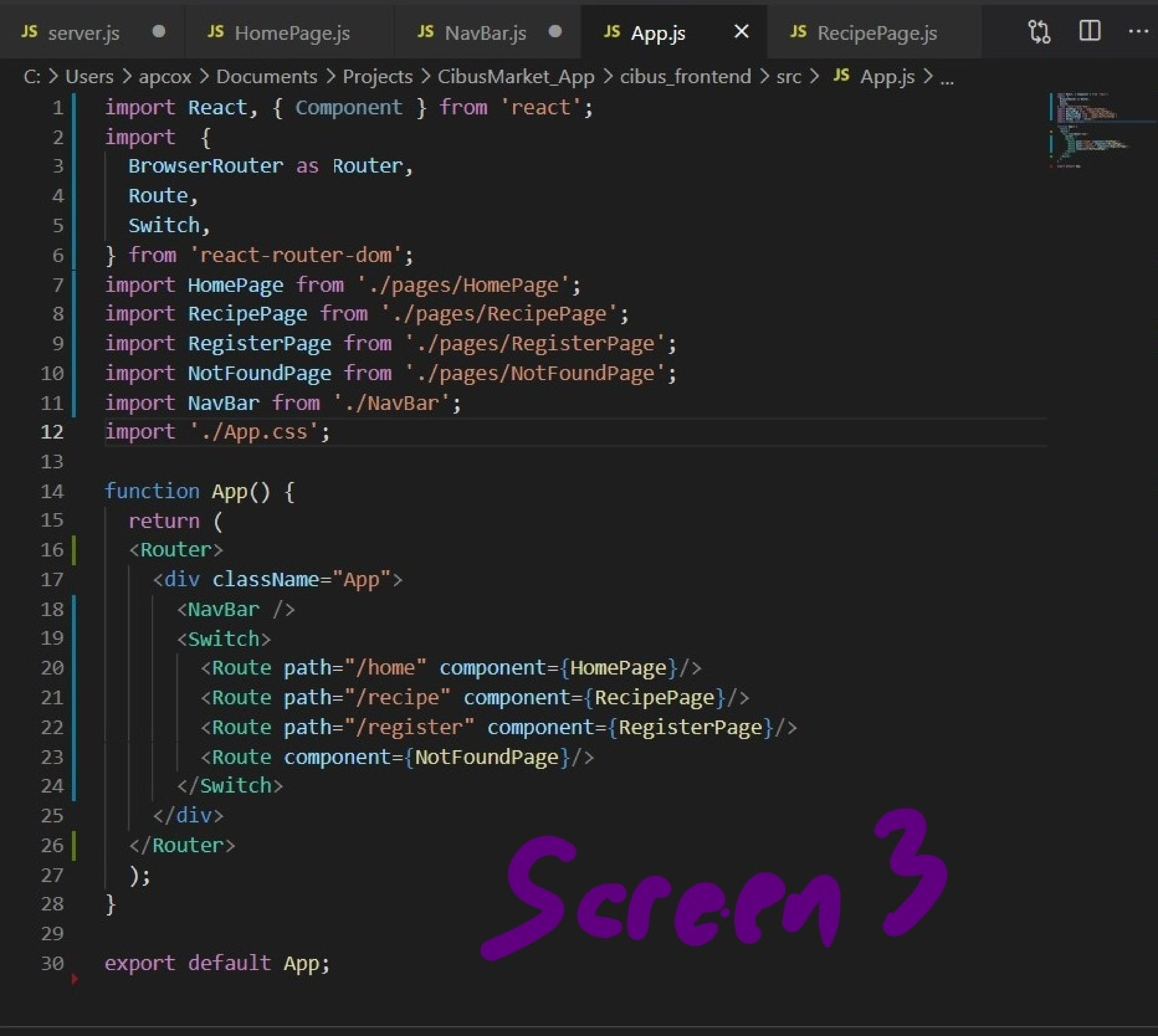
    Note: (The Switch function is just like using a switch statement in other languages.
            Basically, to evaluate cases specified within its boundaries.)

    -> (see screen 3)

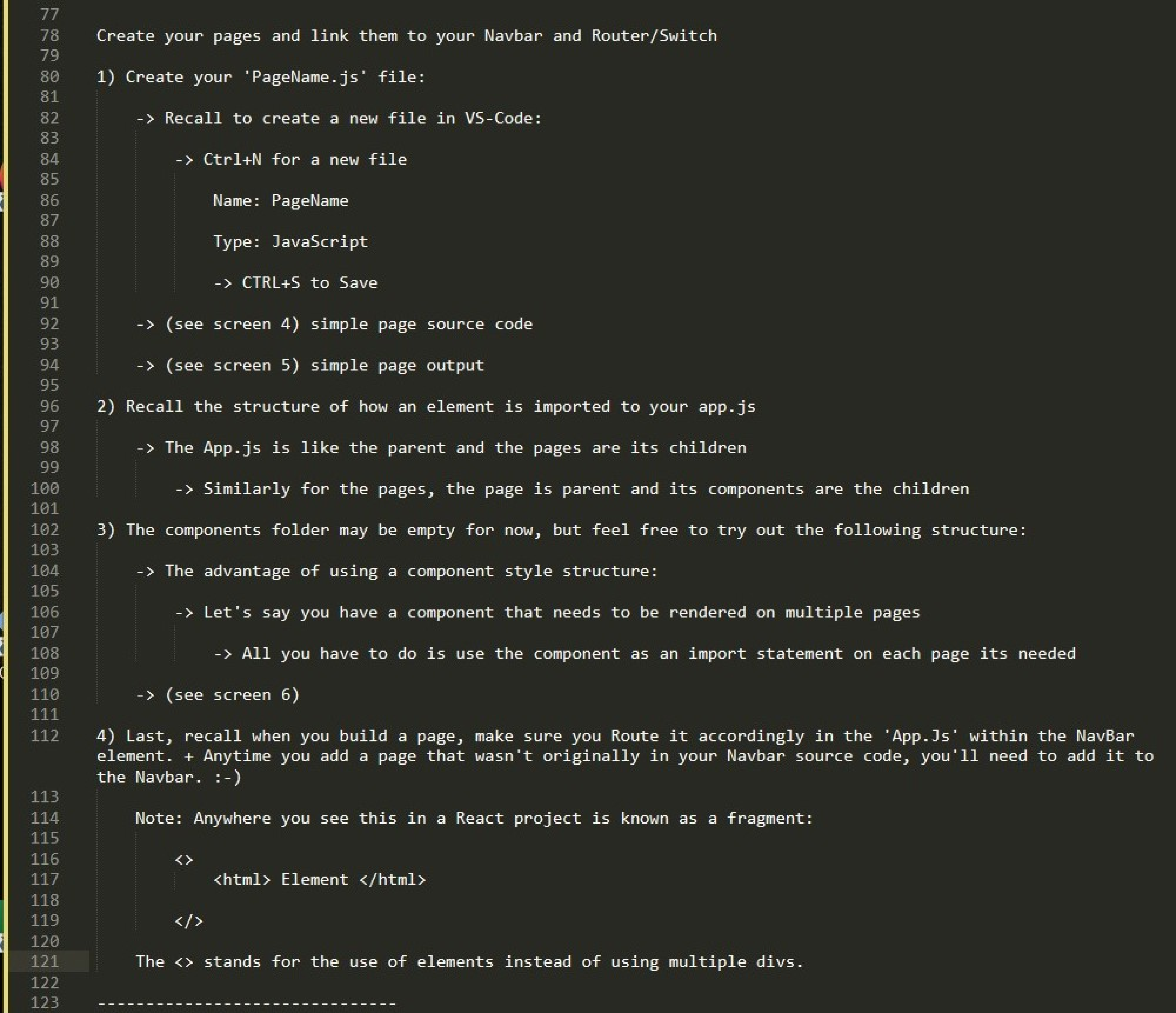

server.js
HomePage.js
NavBar.js
App.js
RecipePage.js
C: > Users > apcox > Documents > Projects > CibusMarket_App > cibus_frontend > src > NavBar.js > NavBar

```
import React from 'react';
import { Link } from 'react-router-dom';

const NavBar = () => (

    <nav>
        <ul>
            <li>
                <Link to="/home">Home</Link>
            </li>
            <li>
                <Link to="/register"> Register</Link>
            </li>
            <li>
                <Link to="/recipe"> Recipe </Link>
            </li>
            <li>
                <Link to="/menu-plan"> Menu Planner </Link>
            </li>
            <li>
                <Link to="/account"> Account </Link>
            </li>
            <li>
                <Link to="/market-ingredient"> Market Ingredient </Link>
            </li>
        </ul>
    </nav>

);

export default NavBar;
```

Screen 2


server.js
HomePage.js
NavBar.js
App.js
RecipePage.js
C: > Users > apcox > Documents > Projects > CibusMarket_App > cibus_frontend > src > App.js > ...
Screen 3

```javascript
import React, { Component } from 'react';
import  {
  BrowserRouter as Router,
  Route,
  Switch,
} from 'react-router-dom';
import HomePage from './pages/HomePage';
import RecipePage from './pages/RecipePage';
import RegisterPage from './pages/RegisterPage';
import NotFoundPage from './pages/NotFoundPage';
import NavBar from './NavBar';
import './App.css';

function App() {
  return (
  <Router>
    <div className="App">
      <NavBar />
      <Switch>
        <Route path="/home" component={HomePage}/>
        <Route path="/recipe" component={RecipePage}/>
        <Route path="/register" component={RegisterPage}/>
        <Route component={NotFoundPage}/>
      </Switch>
    </div>
  </Router>
  );
}

export default App;
```



```
Create your pages and link them to your Navbar and Router/Switch

1) Create your 'PageName.js' file:

    -> Recall to create a new file in VS-Code:

        -> Ctrl+N for a new file

            Name: PageName

            Type: JavaScript

            -> CTRL+S to Save

    -> (see screen 4) simple page source code

    -> (see screen 5) simple page output

2) Recall the structure of how an element is imported to your app.js

    -> The App.js is like the parent and the pages are its children

        -> Similarly for the pages, the page is parent and its components are the children

3) The components folder may be empty for now, but feel free to try out the following structure:

    -> The advantage of using a component style structure:

        -> Let's say you have a component that needs to be rendered on multiple pages

            -> All you have to do is use the component as an import statement on each page its needed

    -> (see screen 6)

4) Last, recall when you build a page, make sure you Route it accordingly in the 'App.Js' within the NavBar element. + Anytime you add a page that wasn't originally in your Navbar source code, you'll need to add it to the Navbar. :-)

    Note: Anywhere you see this in a React project is known as a fragment:

        <>
            <html> Element </html>

        </>

    The <> stands for the use of elements instead of using multiple divs.

-------------------------------
```

server.js
NavBar.js
App.js
HomePage.js
NotFoundPage.js
RecipeList.js
RegisterPage.js
C: > Users > apcox > Documents > Projects > CibusMarket_App > cibus_frontend > src > pages > NotFoundPage.js
C: > Users > apcox > Documents > Projects > CibusMarket_App > cibus_frontend > src > pages > RegisterPage.js

```
import React from 'react';

const NotFoundPage = () => (

    <h1>404: Page Not Found </h1>
);

export default NotFoundPage;
```

Screen 4

```
import React from 'react';

const RegisterPage = () => (

        <div>
            <h4> Test return element </h4>
        </div>

);

export default RegisterPage;
```

localhost:3000/register

- Home
- Register
- Recipe
- Menu Planner
- Account
- Market Ingredient

Test return element

```
<!DOCTYPE html>
<html lang="en">
  ▼<head> == $0
    <meta charset="utf-8">
```

element.style {
}

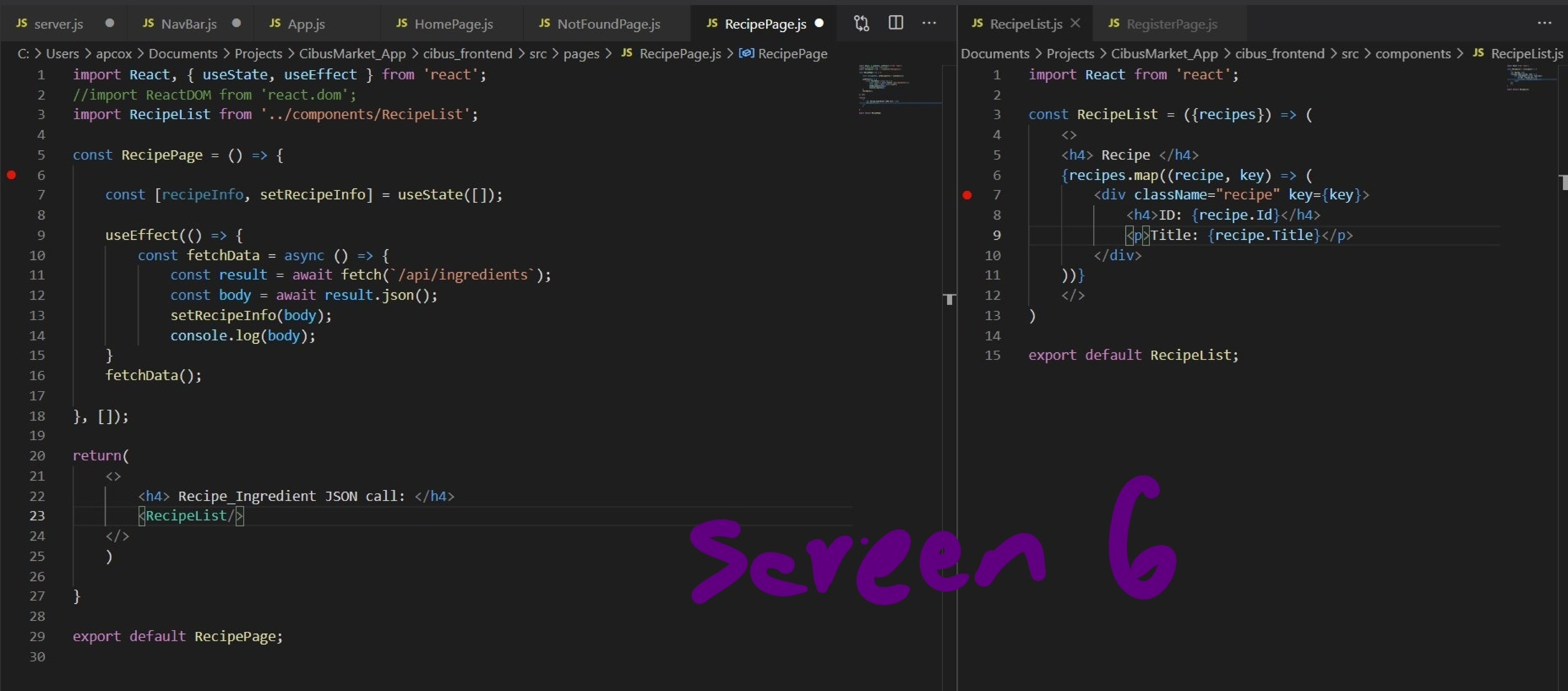
localhost:3000/market-ingredient

- Home
- Register
- Recipe
- Menu Planner
- Account
- Market Ingredient

404: Page Not Found

Screen 5

Cibus Market
localhost:3001/api/recipe
Introducing JSX – React
default page syntax react js
Java Switch
Apps
Gmail
YouTube
Collision Login
Key shortcuts - Spo...
DU-Paycor
Home | myDU
Dominican Universi...
Design toolkits and...
Adobe XD — 30 Tip...
Elements
Console
Sources
Network
Styles
Computed
Layout
Event Listeners
DOM Breakpoints
Properties
Accessibility
Filter
:hov
.cls
server.js
master*
JavaScript
Type here to search
5:09 PM
2/11/2021


server.js
NavBar.js
App.js
HomePage.js
NotFoundPage.js
RecipePage.js
RecipeList.js
RegisterPage.js
C: > Users > apcox > Documents > Projects > CibusMarket_App > cibus_frontend > src > pages > RecipePage.js > RecipePage
Documents > Projects > CibusMarket_App > cibus_frontend > src > components > RecipeList.js

```
import React, { useState, useEffect } from 'react';
//import ReactDOM from 'react.dom';
import RecipeList from '../components/RecipeList';

const RecipePage = () => {

    const [recipeInfo, setRecipeInfo] = useState([]);

    useEffect(() => {
        const fetchData = async () => {
            const result = await fetch(`/api/ingredients`);
            const body = await result.json();
            setRecipeInfo(body);
            console.log(body);
        }
        fetchData();

    }, []);

    return(
        <>
            <h4> Recipe_Ingredient JSON call: </h4>
            <RecipeList/>
        </>
        )

}

export default RecipePage;
```

```
import React from 'react';

const RecipeList = ({recipes}) => (
    <>
    <h4> Recipe </h4>
    {recipes.map((recipe, key) => (
        <div className="recipe" key={key}>
            <h4>ID: {recipe.Id}</h4>
            <p>Title: {recipe.Title}</p>
        </div>
    ))}
    </>
)

export default RecipeList;
```