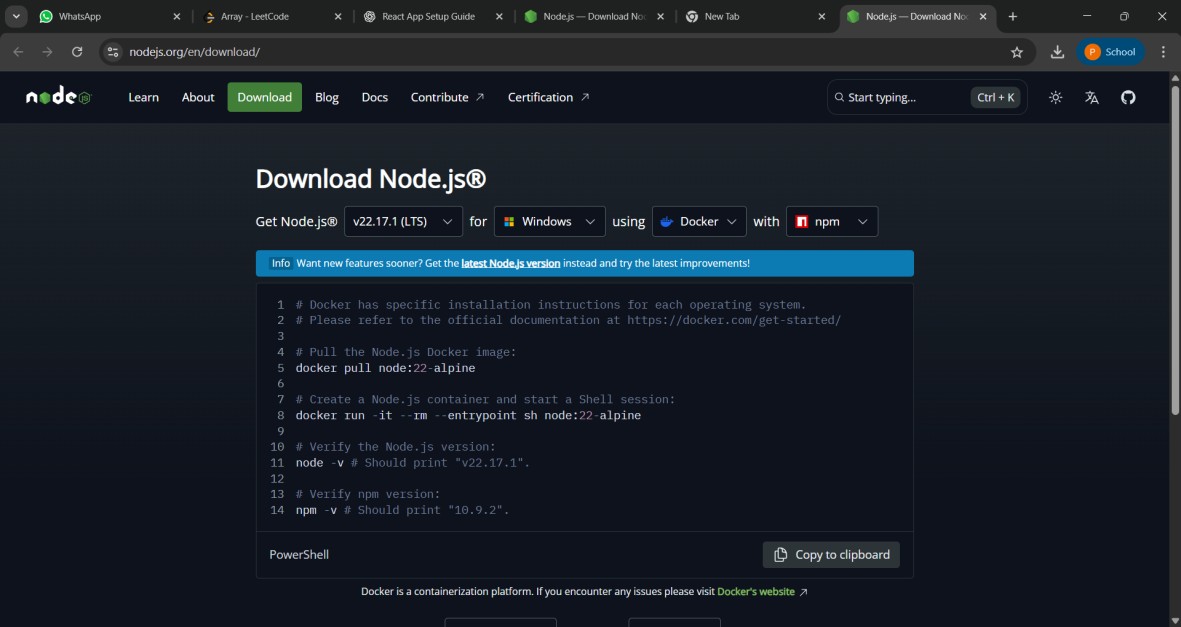
**1. ReactJS-HOL**

**Create and Run Your First React App**

**Prerequisites (Install Only Once) Step 1: Install Node.js and npm**

* **Visit:**  [**https://nodejs.org/en/download/**](https://nodejs.org/en/download/)
* **Download the LTS version (recommended for most users)**
* **Install it (npm comes bundled with Node.js)**

****

* **To confirm installation, open your terminal/command prompt:**

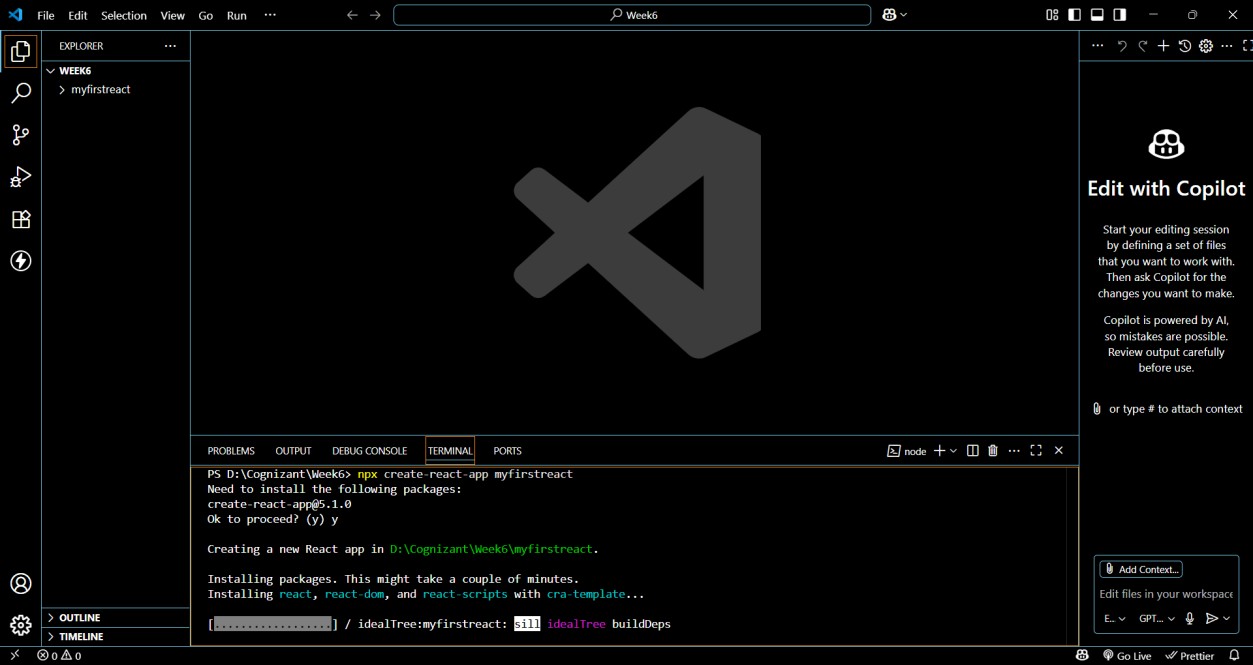
**node -v npm -v**

**React App Setup Step 2: Install create-react-app (Only Once)**

**npm install -g create-react-app**

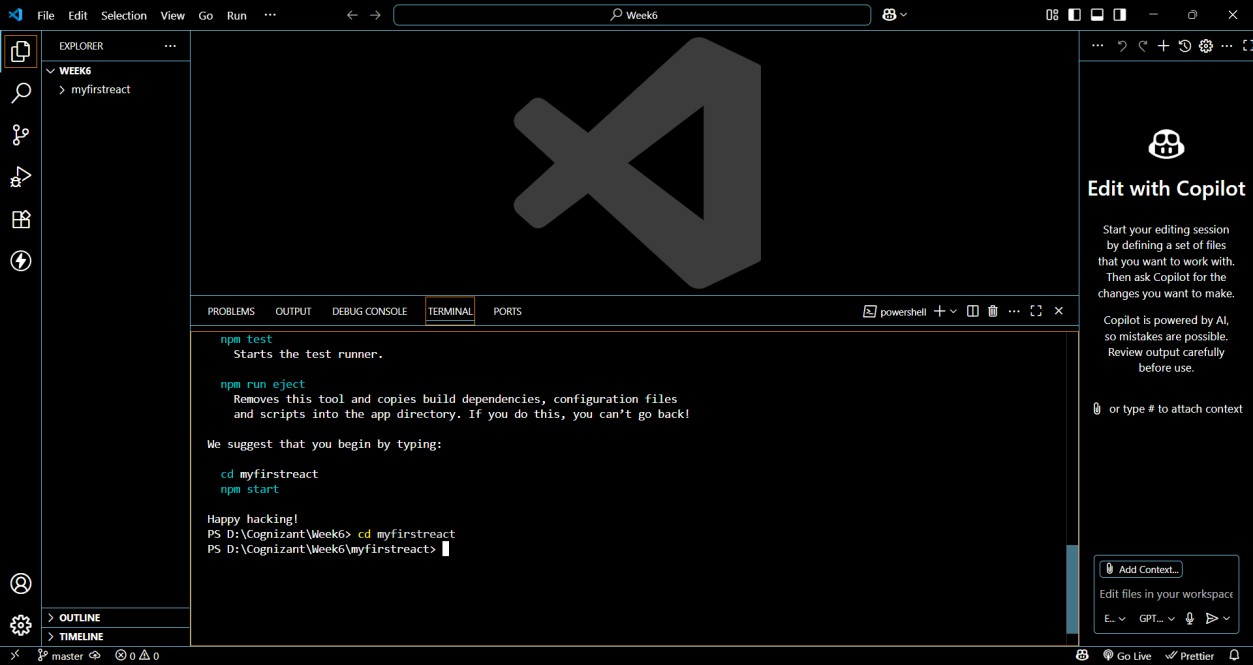
**Step 3: Create a new app named myfirstreact**

**npx create-react-app myfirstreact**

****

**Step 4: Move into the project folder**

**cd myfirstreact**

****

**Modify the App Content Step 5: Open App.js**

**Go to: src > App.js Delete all the existing code Paste this simple code:**

**import React from 'react';**

**function App() { return (**

**<div>**

**<h1>Welcome to the first session of React</h1>**

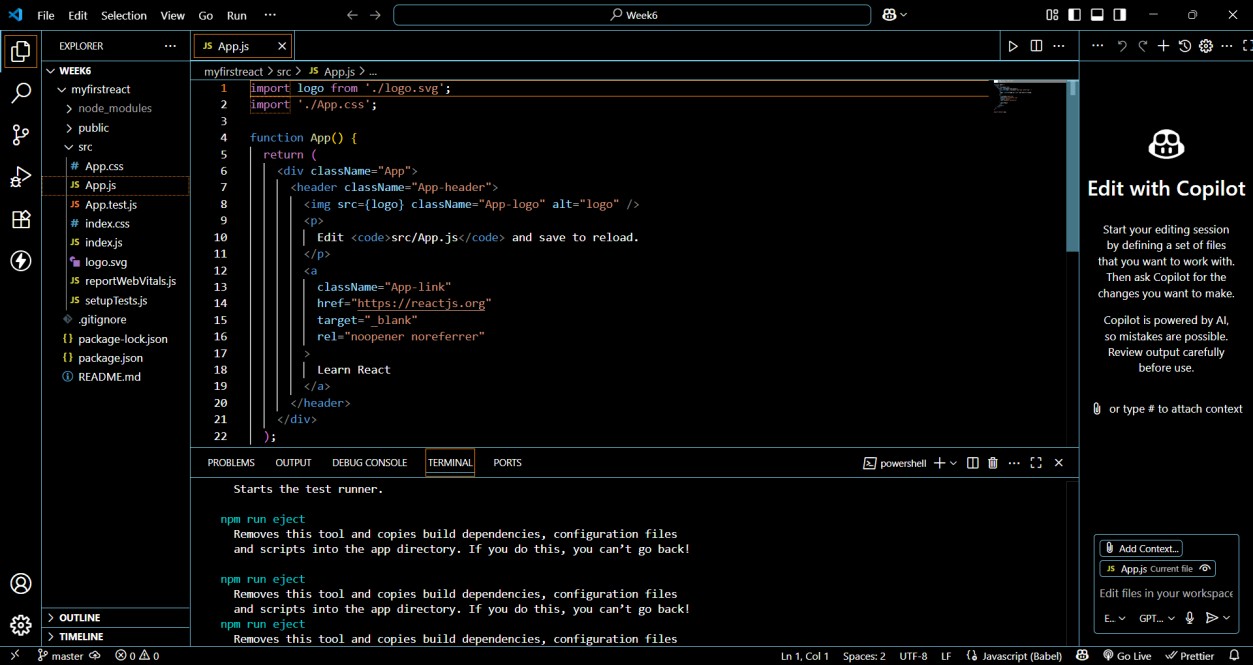
**</div>**

**);**

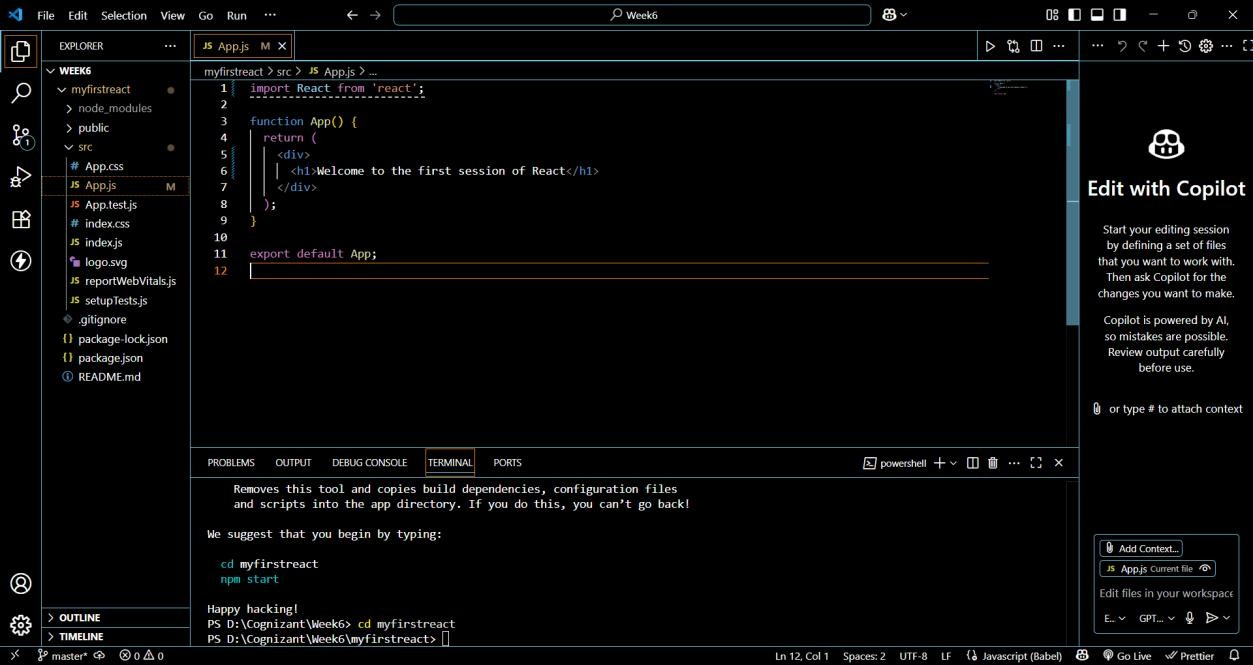
**}**

**export default App;**

**Existing App.js:**

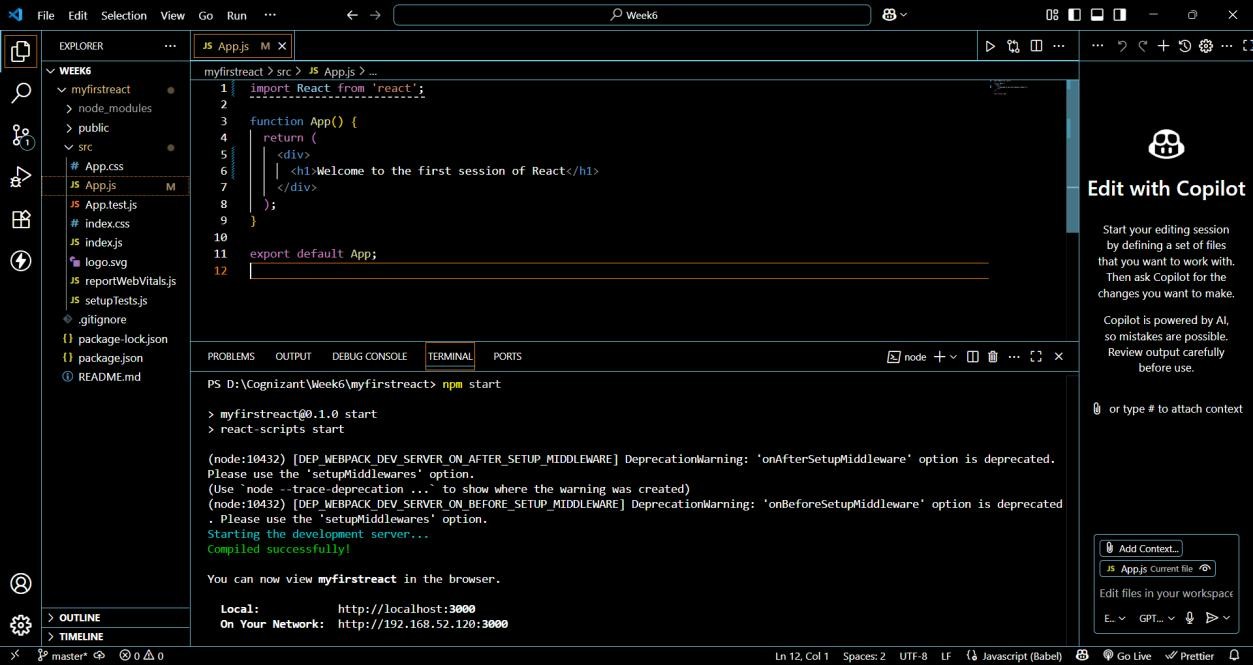
****

**Modified App.js:**

****

**Run the Application Step 7: Start the development server**

**In the terminal run: npm start**

****

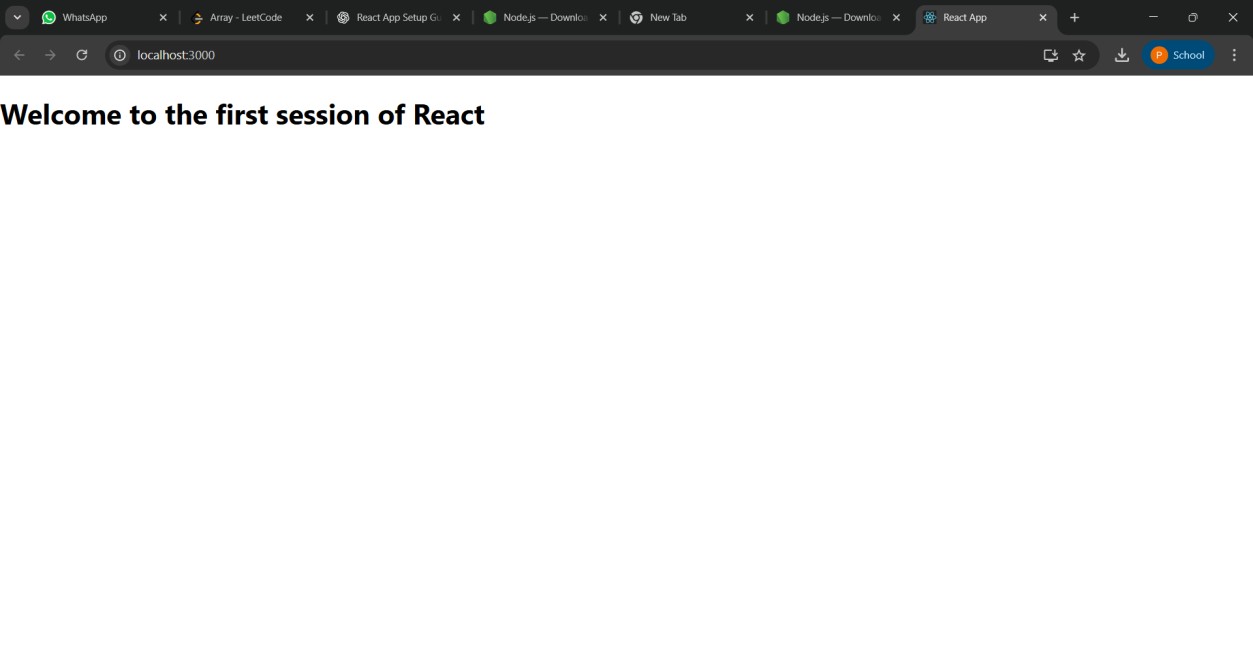
**Step 8: View in Browser**

**Open your browser and go to:**

**http://localhost:3000 You should see:**

**Welcome to the first session of React**

**Result:**

****

**2. ReactJS-HOL**

**Create StudentApp React App with 3 Components**

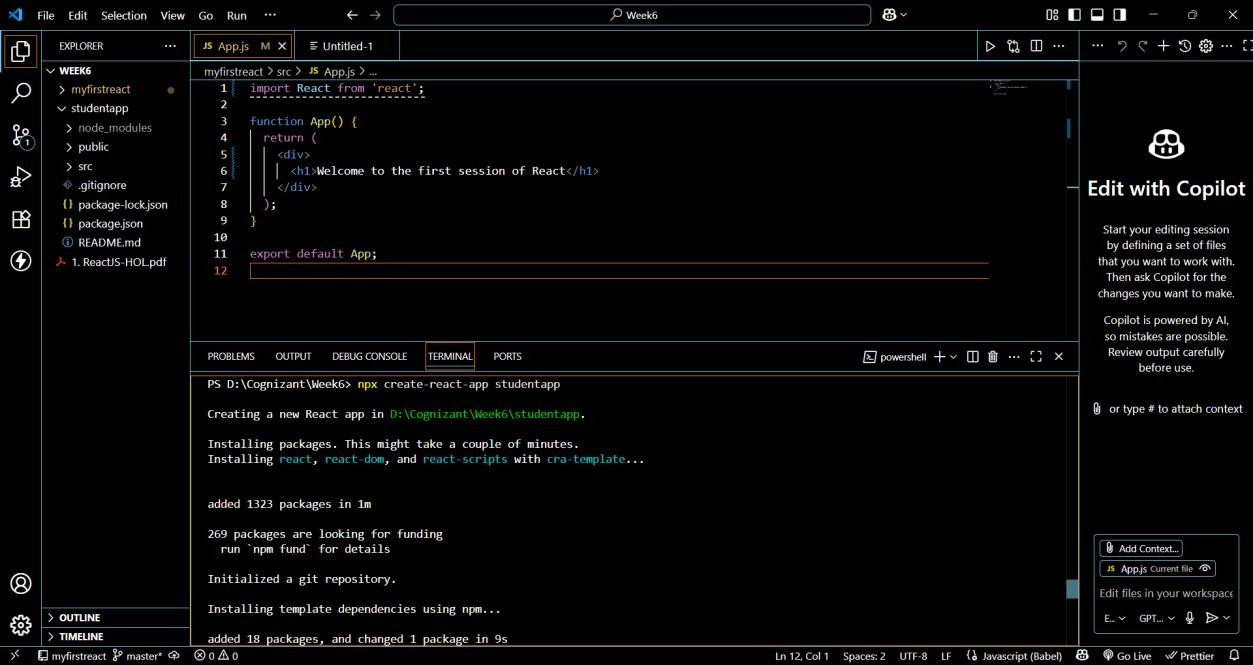
**Step 1: Open VS Code and Create React App**

1. **Open VS Code**
2. **Open a new folder where you want to create your app.**
3. **Open the terminal in VS Code:**

**Terminal → New Terminal**

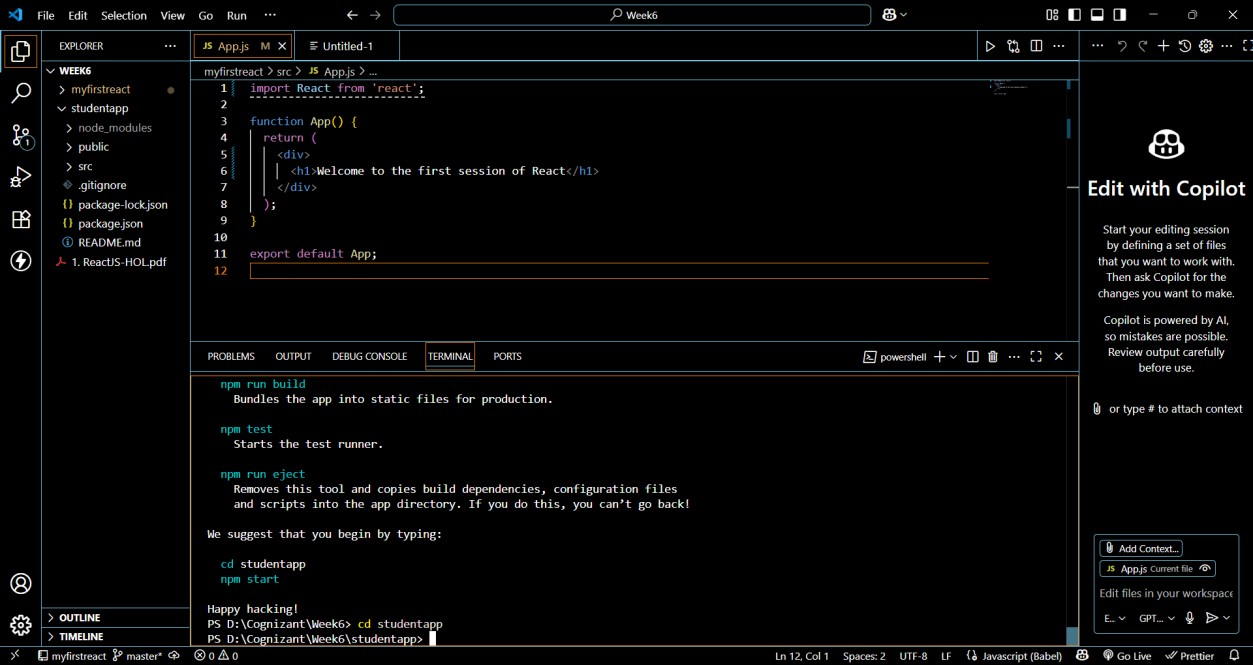
1. **Run this command:**

**npx create-react-app studentapp**

****

**Step 2: Move Into the Project Folder**

**cd studentapp**

****

**Step 3: Create a Folder for Components In the src folder:**

**1. Right-click → New Folder → Name it Components Step 4: Create Home.js Component (Class Component) In src/Components/Home.js, add:**

**import React, { Component } from 'react';**

**class Home extends Component {**

**constructor(props) { super(props);**

**}**

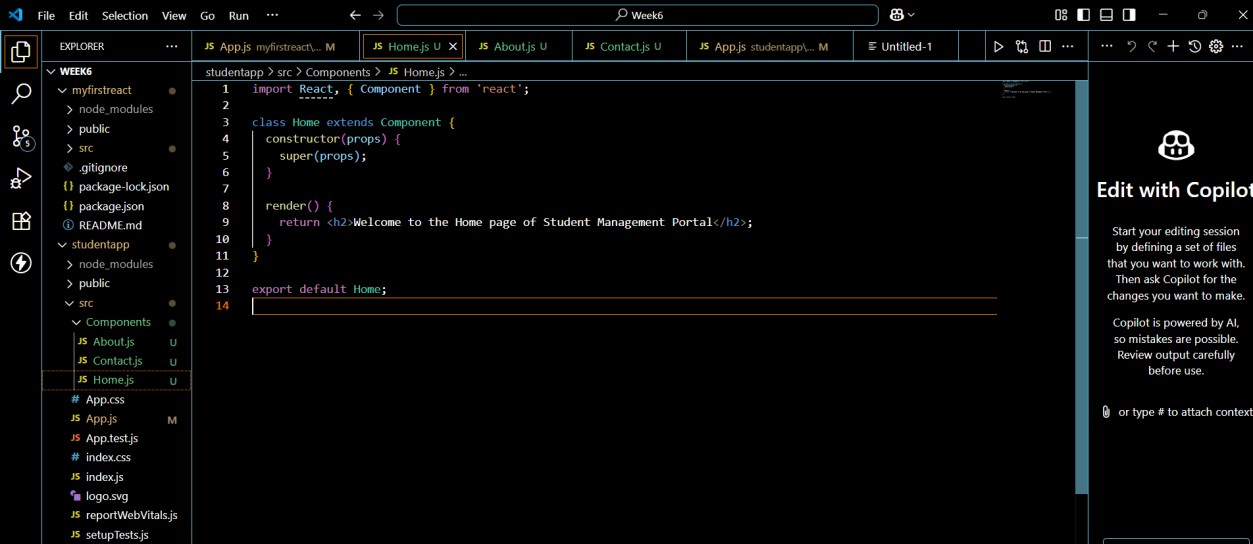
**render() {**

**return <h2>Welcome to the Home page of Student Management Portal</h2>;**

**}**

**}**

**export default Home;**

****

**Step 5: Create About.js Component (Class Component) In src/Components/About.js, add:**

**import React, { Component } from 'react';**

**class About extends Component {**

**constructor(props) { super(props);**

**}**

**render() {**

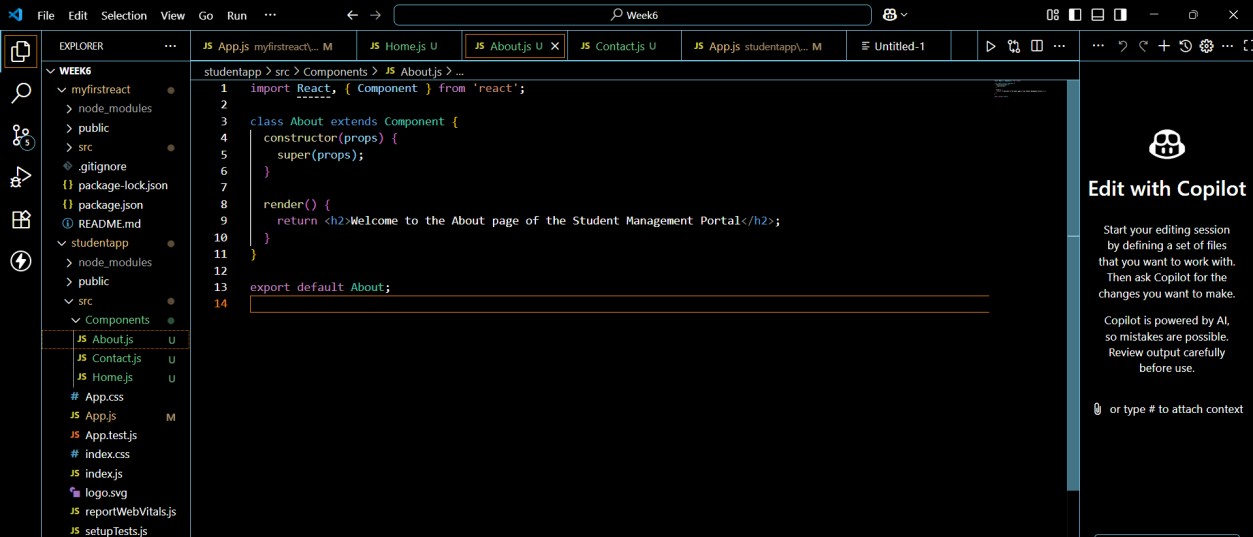
**return <h2>Welcome to the About page of the Student Management**

**Portal</h2>;**

**}**

**}**

**export default About;**

****

**Step 6: Create Contact.js Component (Class Component) In src/Components/Contact.js, add:**

**import React, { Component } from 'react';**

**class Contact extends Component {**

**constructor(props) { super(props);**

**}**

**render() {**

**return <h2>Welcome to the Contact page of the Student Management**

**Portal</h2>;**

**}**

**}**

**export default Contact;**

****

**Step 7: Edit App.js to Use the Components In src/App.js, replace everything with:**

**import React from 'react';**

**import Home from './Components/Home'; import About from './Components/About';**

**import Contact from './Components/Contact';**

**function App() { return (**

**<div>**

**<h1>Student Management Portal</h1>**

**<Home />**

**<About />**

**<Contact />**

**</div>**

**);**

**}**

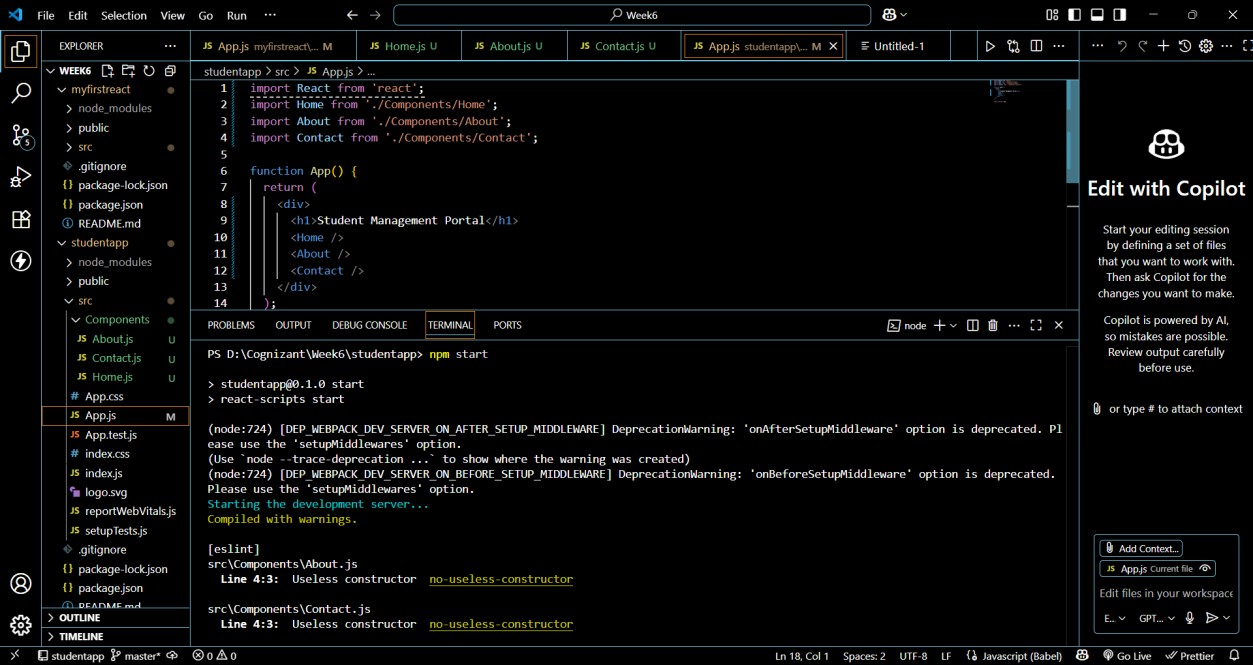
**export default App;**

****

**Step 8: Run the React App**

**In the terminal, inside the StudentApp folder, run:**

**npm start**

****

**Step 9: View Output in Browser Open a browser and visit:**

**http://localhost:3000**

**You should see:**

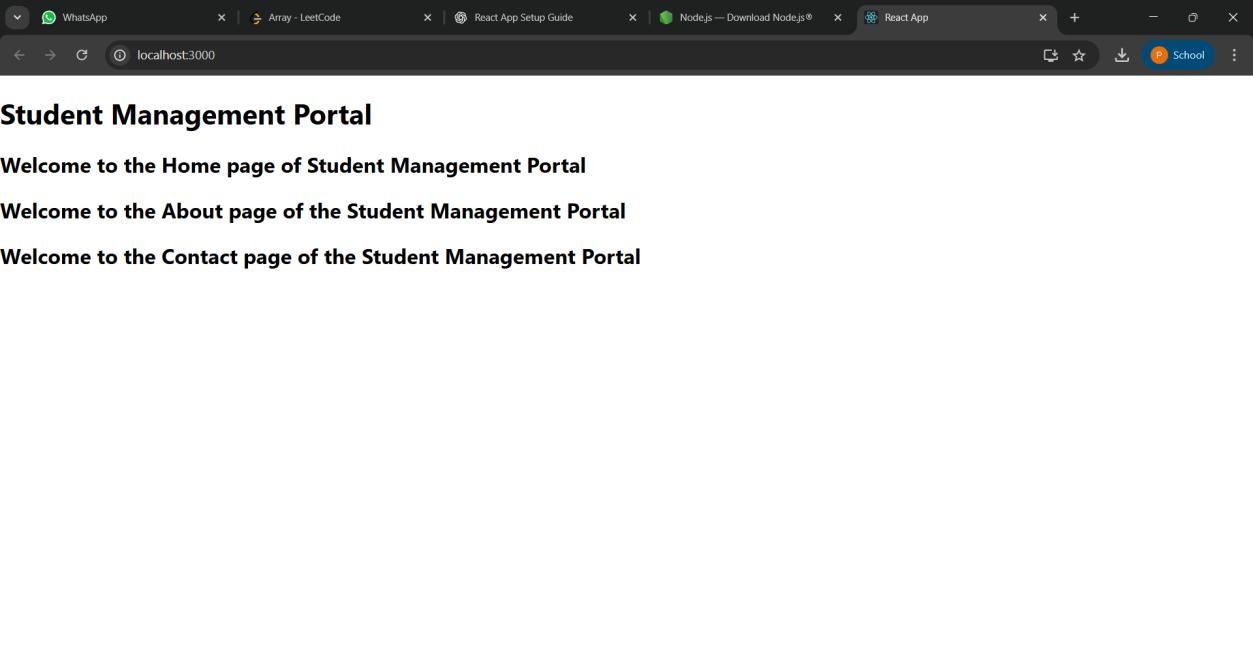
**Student Management Portal**

**Welcome to the Home page of Student Management Portal**

**Welcome to the About page of the Student Management Portal**

**Welcome to the Contact page of the Student Management Portal**

**Result:**

****

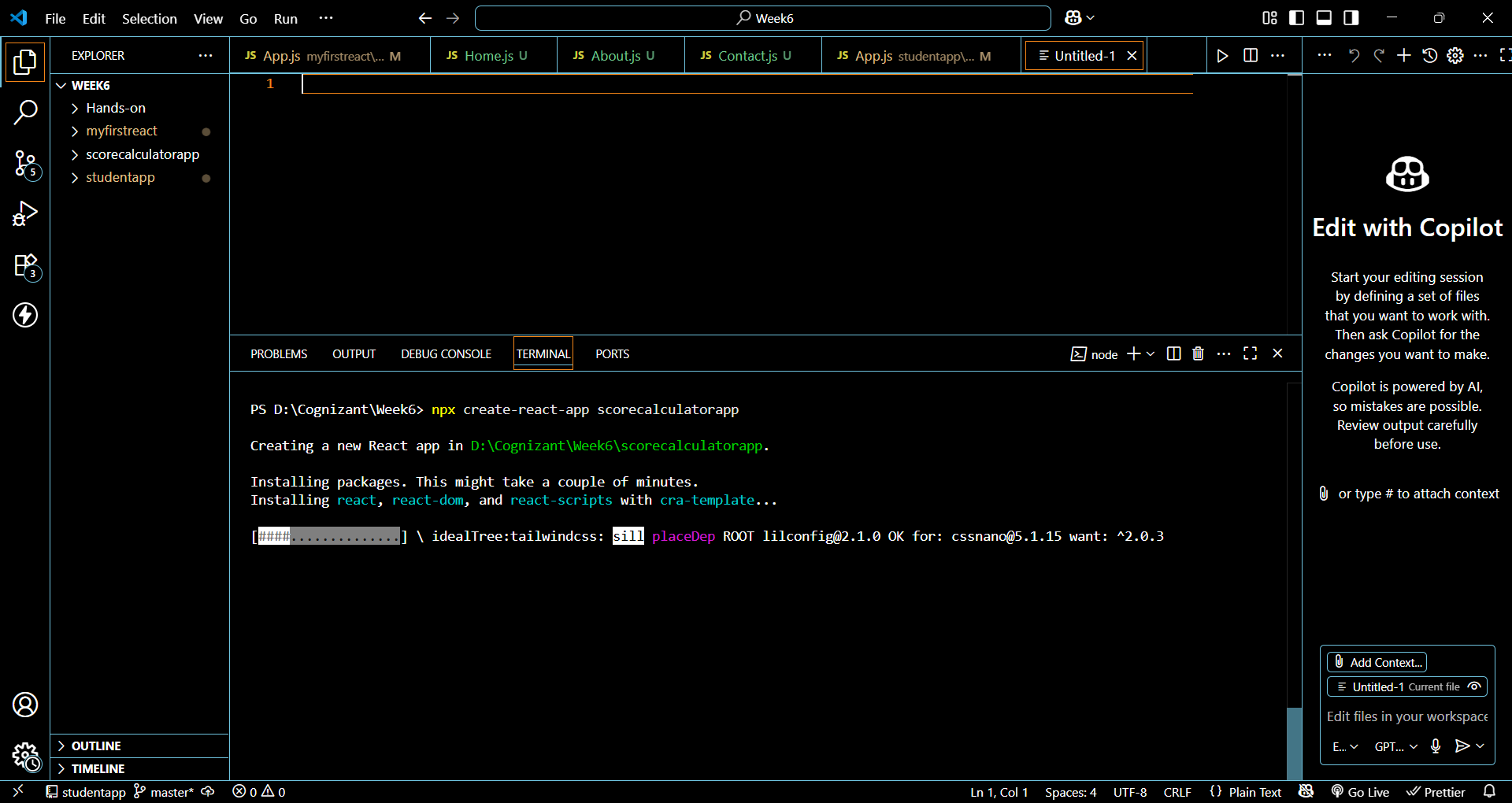
**3. ReactJS-HOL**

**Create scorecalculatorapp with a Functional Component**

**Step 1: Create the React App**

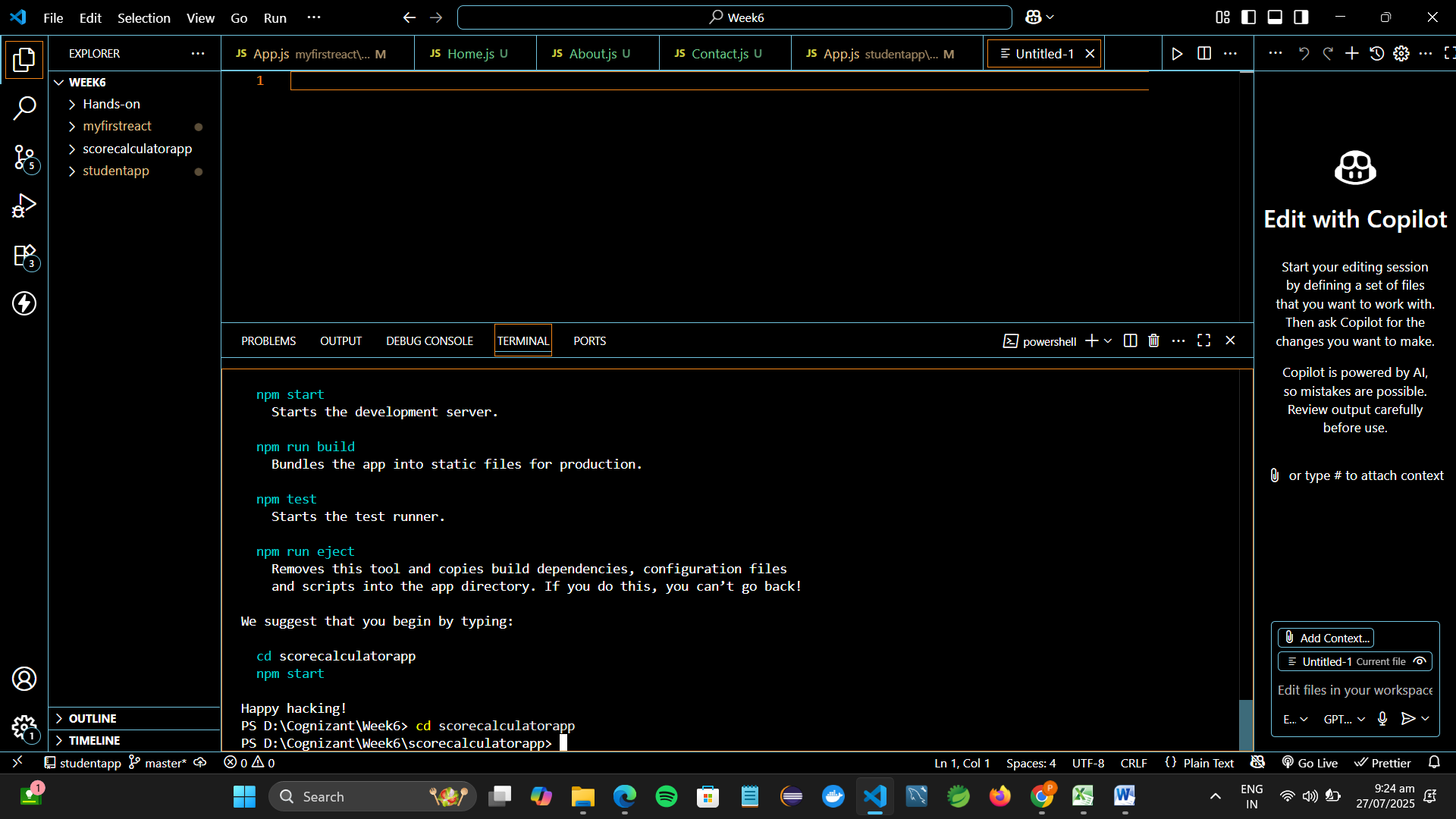
1. **Open VS Code**
2. **Open a new folder for the project**
3. **Open Terminal → Terminal → New Terminal**
4. **Run:**

**npx create-react-app scorecalculatorapp**

****

**Step 2: Move Into the Project Directory**

**cd scorecalculatorapp**

****

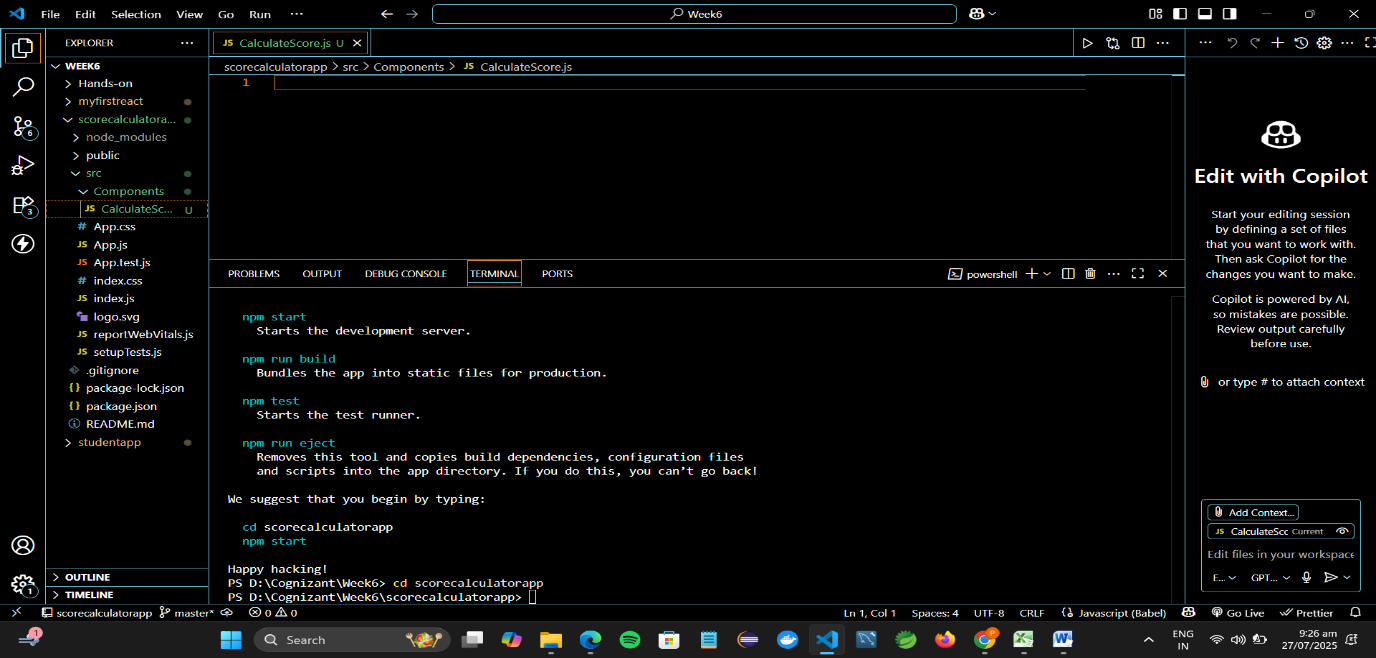
**Step 3: Create Functional Component CalculateScore**

1. **Inside src, create a folder named:**

**Components**

1. **Inside Components, create a file:**

**CalculateScore.js**

****

1. **Add the following code to CalculateScore.js:**

**import React from 'react';**

**import '../Stylesheets/mystyle.css';**

**function CalculateScore() {**

**const name = "Pooja";**

**const school = "SKCET";**

**const total = 450;**

**const goal = 500;**

**const average = (total / goal) \* 100;**

**return (**

**<div className="score-container">**

**<h2>Student Score Calculator</h2>**

**<p><strong>Name:</strong> {name}</p>**

**<p><strong>School:</strong> {school}</p>**

**<p><strong>Total Marks:</strong> {total}</p>**

**<p><strong>Goal:</strong> {goal}</p>**

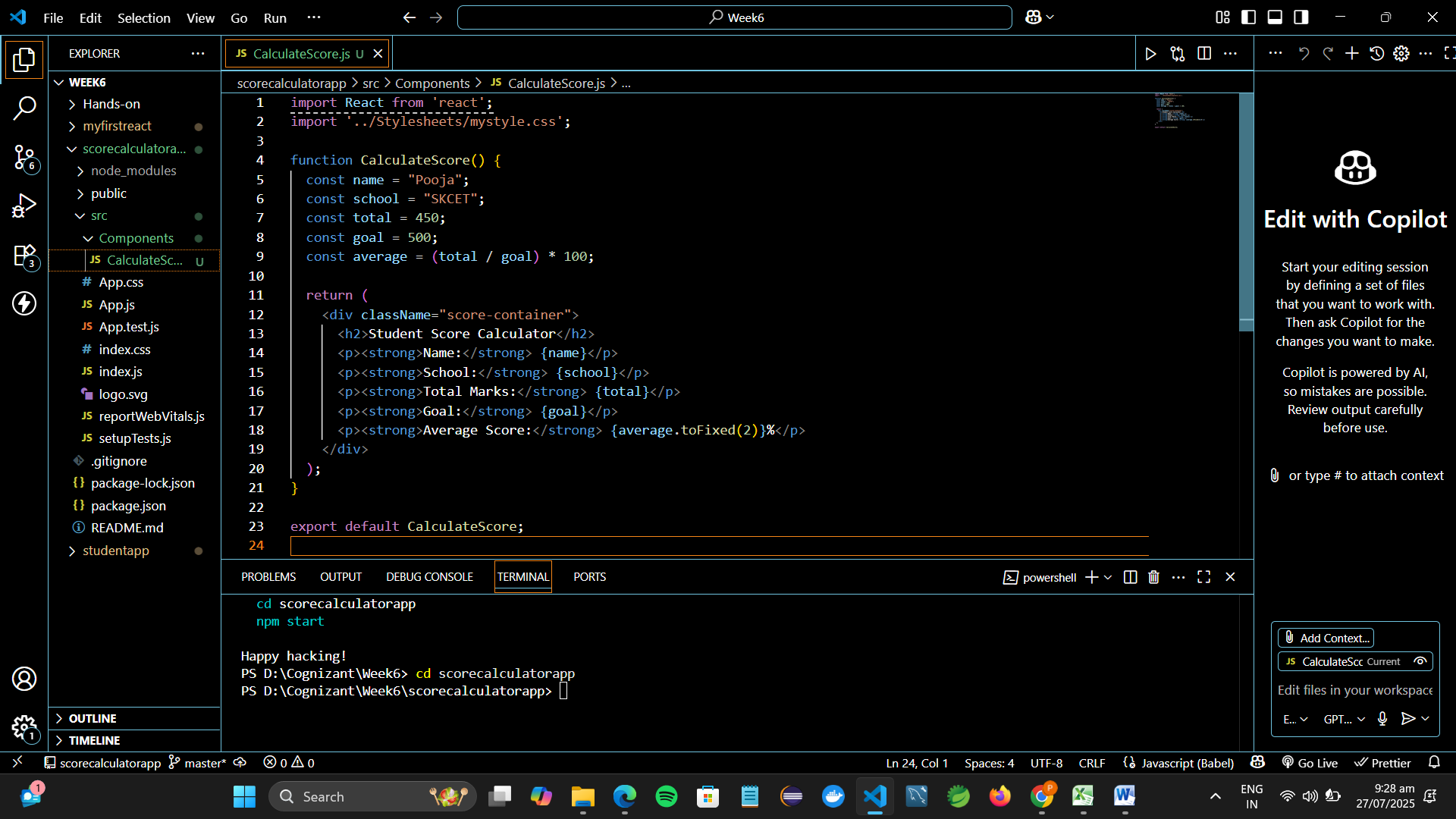
**<p><strong>Average Score:</strong> {average.toFixed(2)}%</p>**

**</div>**

**);**

**}**

**export default CalculateScore;**

****

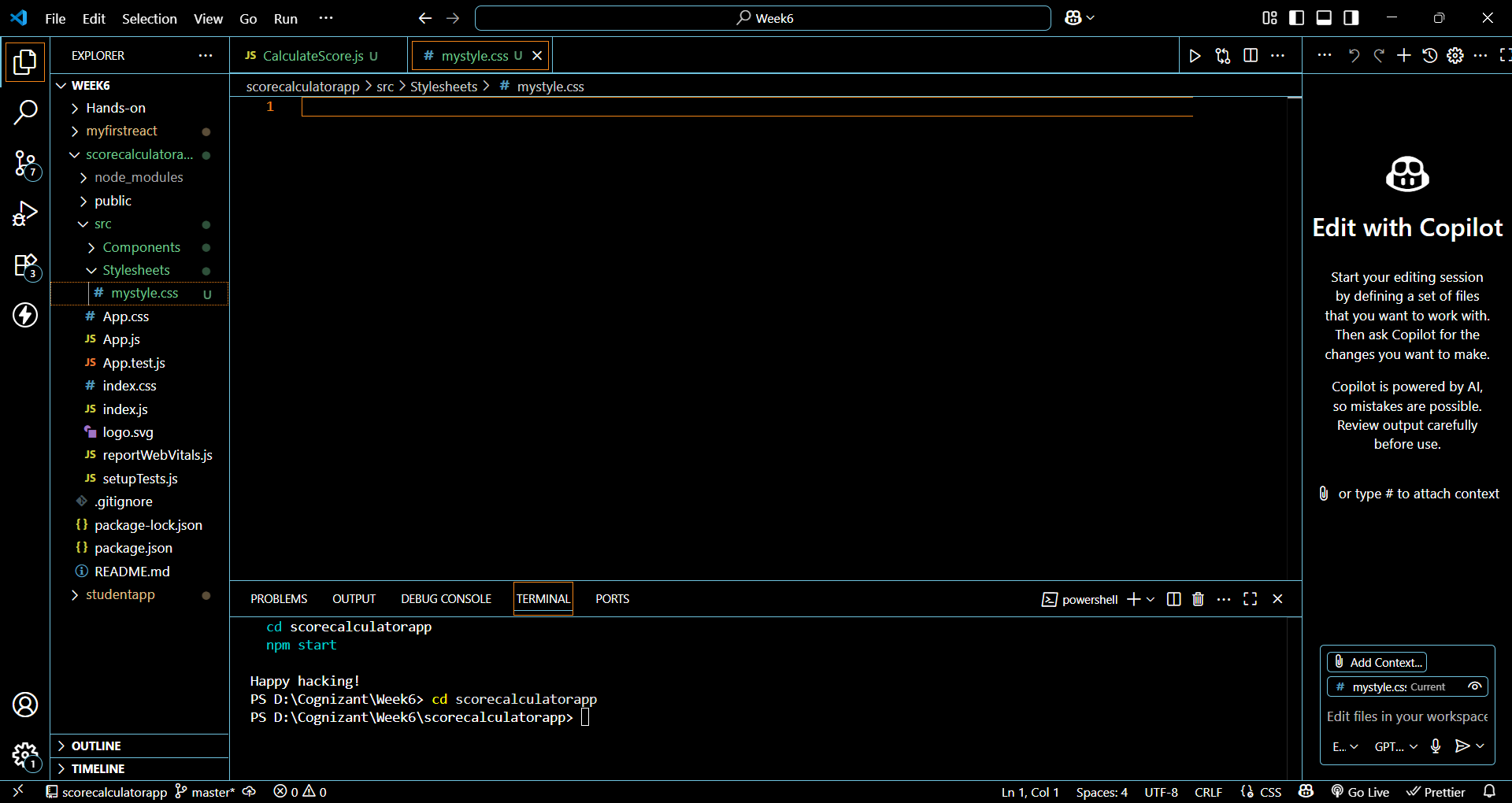
**Step 4: Add Styling with mystyle.css**

1. **Inside src, create a folder:**

**Stylesheets**

1. **Inside Stylesheets, create a file:**

**mystyle.css**

****

1. **Add some styles:**

**.score-container {**

**background-color: #f5f5f5;**

**border: 2px solid #4caf50;**

**padding: 20px;**

**width: 400px;**

**margin: 40px auto;**

**border-radius: 10px;**

**text-align: left;**

**font-family: 'Arial', sans-serif;**

**}**

**.score-container h2 {**

**color: #4caf50;**

**text-align: center;**

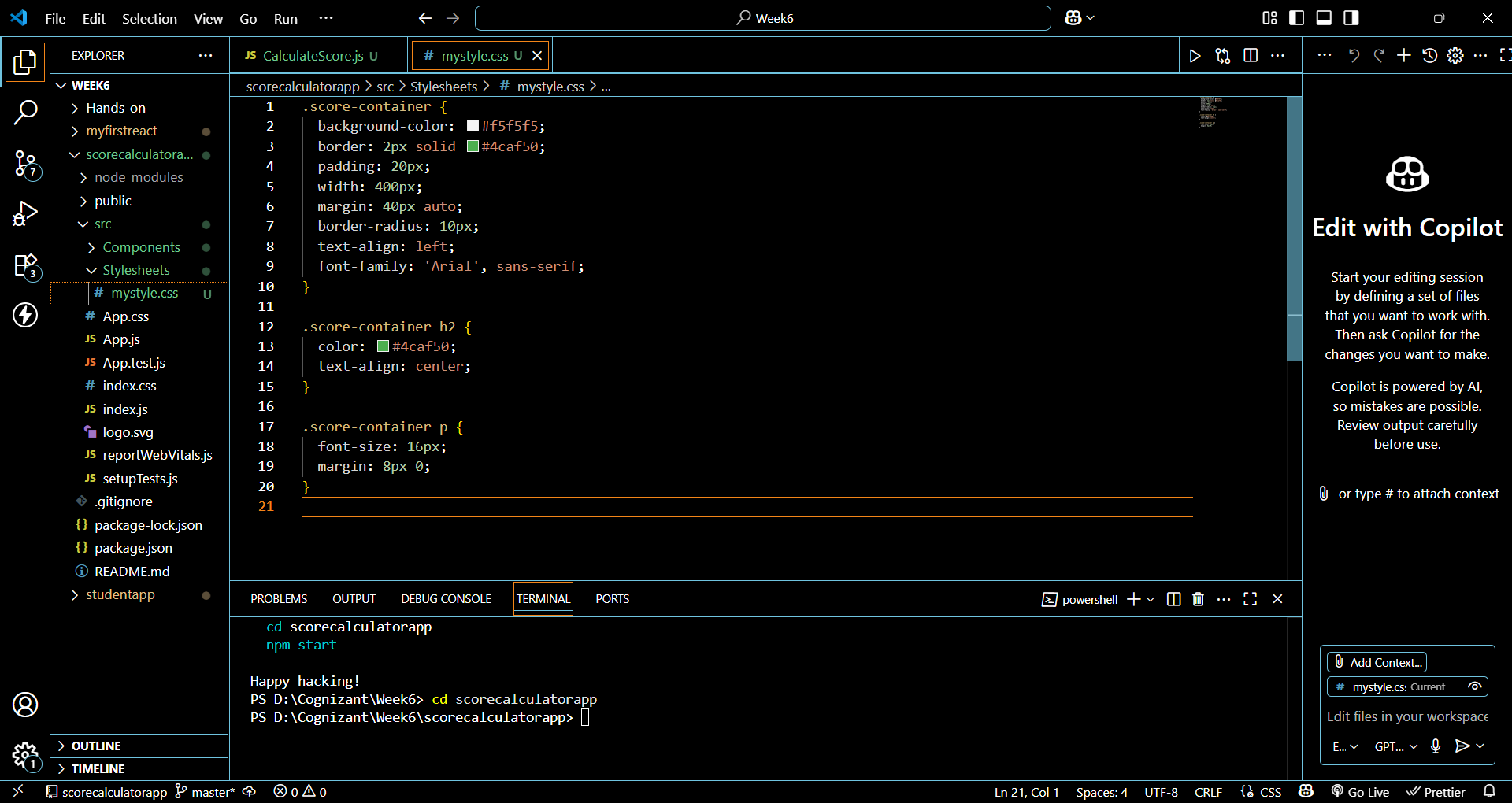
**}**

**.score-container p {**

**font-size: 16px;**

**margin: 8px 0;**

**}**

****

**Step 5: Edit App.js to Render the Component**

**In src/App.js, replace everything with:**

**import React from 'react';**

**import CalculateScore from './Components/CalculateScore';**

**function App() {**

**return (**

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

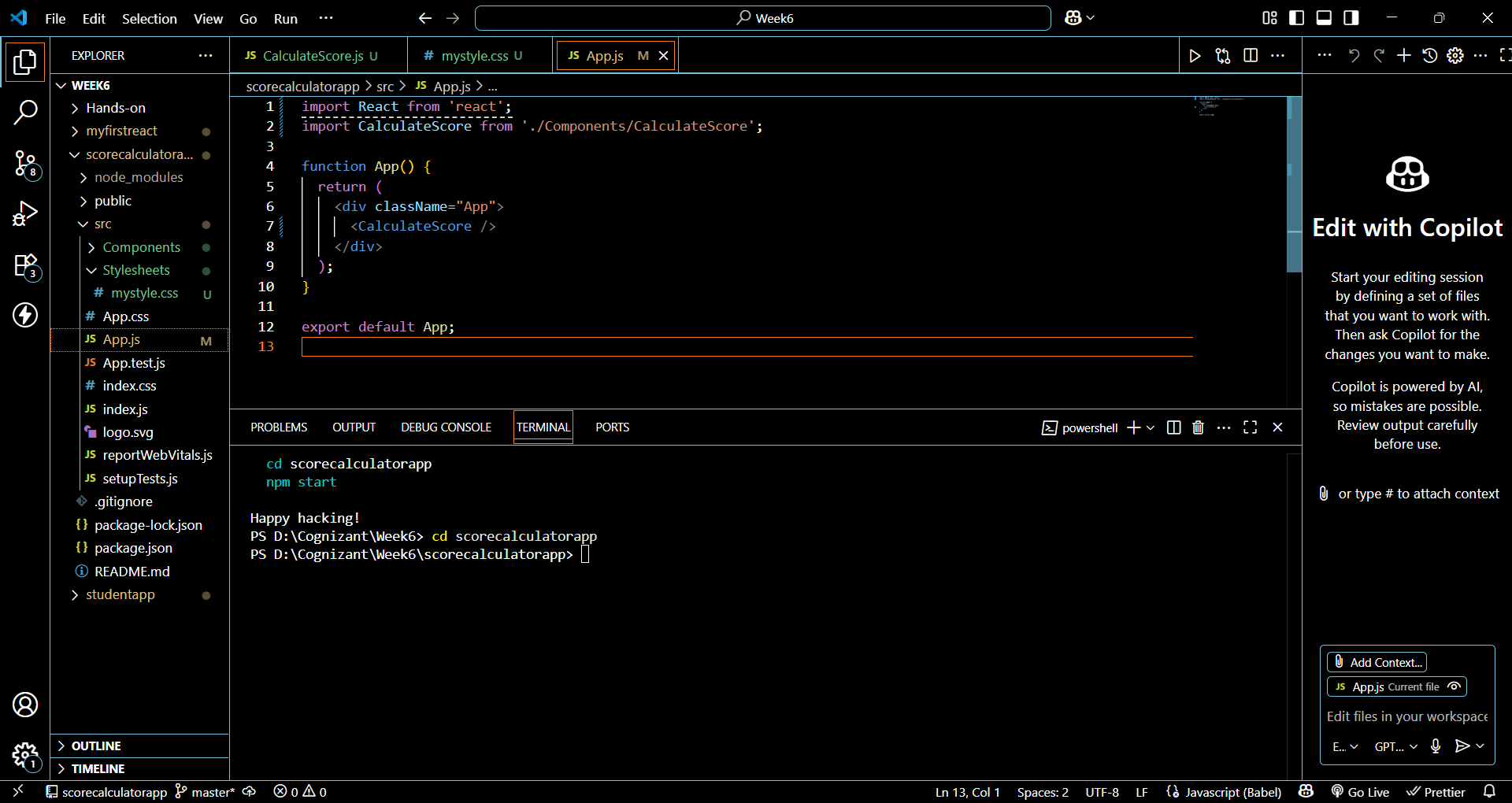
**<CalculateScore />**

**</div>**

**);**

**}**

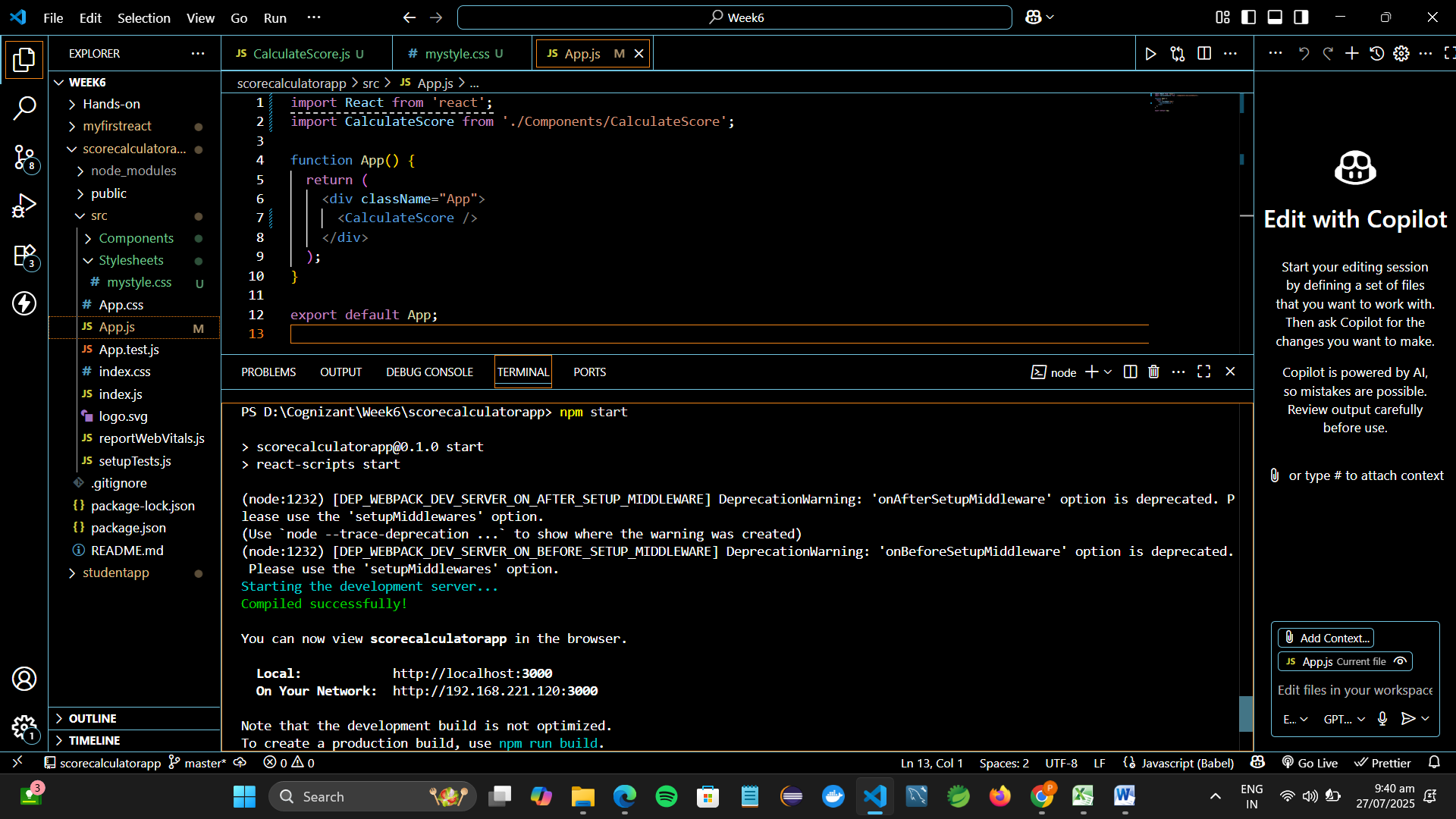
**export default App;**

****

**Step 6: Run the App**

**In terminal (inside scorecalculatorapp folder), run:**

**npm start**

****

**Step 7: View in Browser**

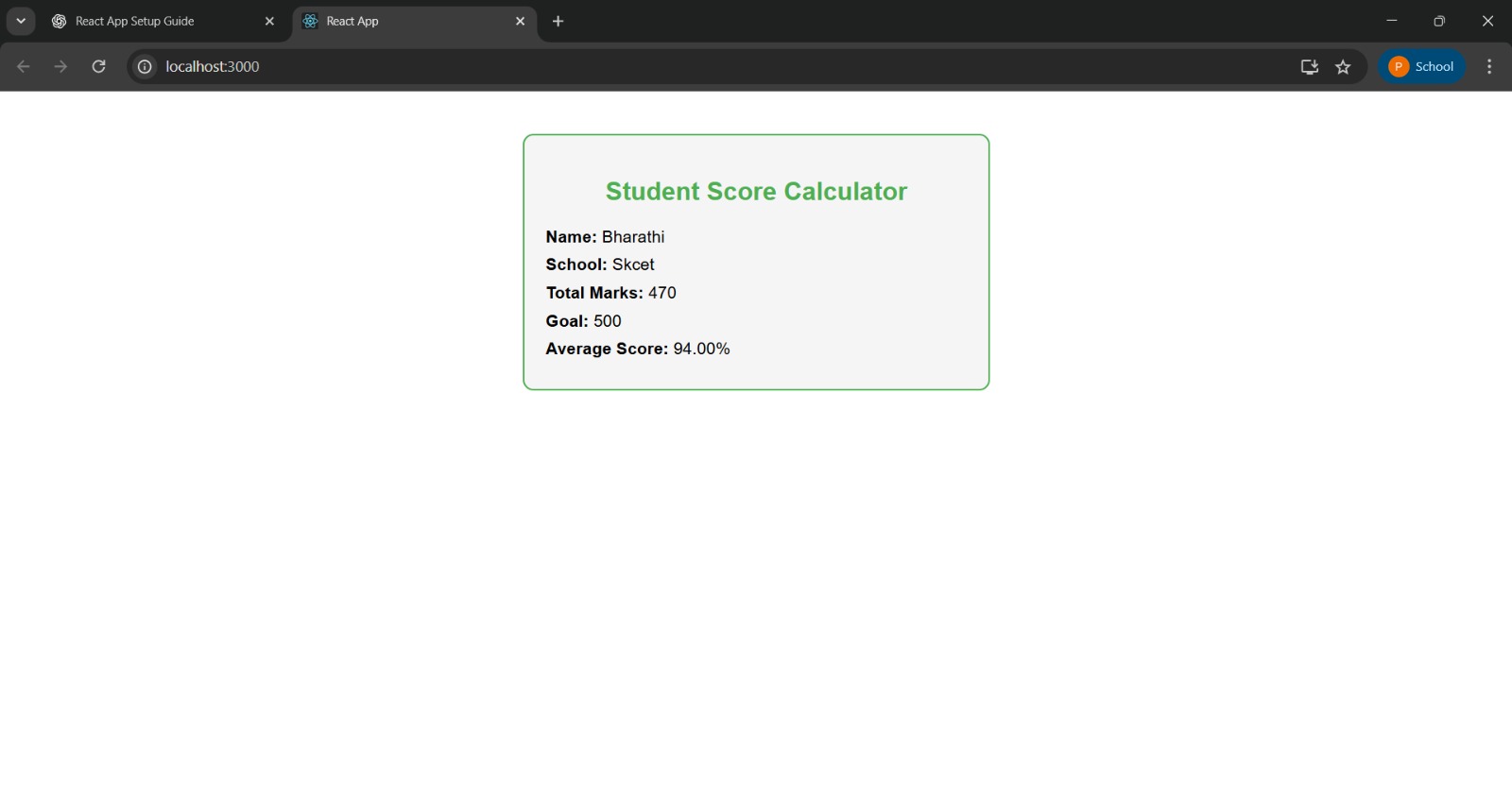
**Open a browser and visit:**

**http://localhost:3000**

**You should see a styled box showing:**

* **Student name**
* **School name**
* **Total marks**
* **Goal**
* **Calculated average in %**

**Result:**



**4.Objectives**

* Explain the need and Benefits of component life cycle
* Identify various life cycle hook methods
* List the sequence of steps in rendering a component

In this hands-on lab, you will learn how to:

* Implement componentDidMount() hook
* Implementing componentDidCatch() life cycle hook.

**Prerequisites**

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

**Notes**

Estimated time to complete this lab: **60 minutes.**

1. Create a new react application using *create-react-app* tool with the name as “blogapp”
2. Open the application using VS Code
3. Create a new file named as **Post.js** in **src folder** with following properties



*Figure 2: Post class*

1. Create a new class based component named as **Posts** inside **Posts.js** file



*Figure 3: Posts Component*

1. Initialize the component with a list of Post in state of the component using the constructor
2. Create a new method in component with the name as **loadPosts()** which will be responsible for using Fetch API and assign it to the component state created earlier. To get the posts use the url (<https://jsonplaceholder.typicode.com/posts>)



*Figure 4: loadPosts() method*

1. Implement the **componentDidMount()** hook to make calls to **loadPosts()** which will fetch the posts



*Figure 5: componentDidMount() hook*

1. Implement the **render()** which will display the title and post of posts in html page using heading and paragraphs respectively.



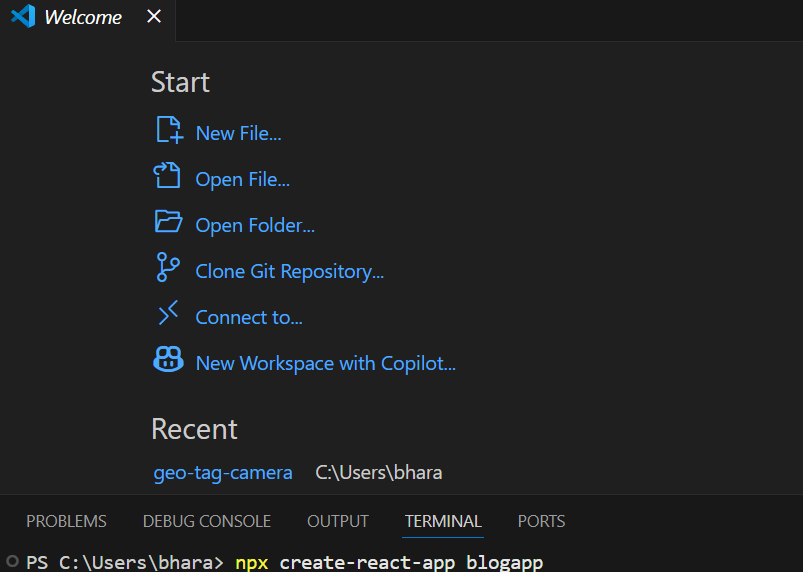
*Figure 6: render() method*

1. Define a **componentDidCatch()** method which will be responsible for displaying any error happing in the component as alert messages.



*Figure 7: componentDidCatch() hook*

1. Add the Posts component to App component.
2. Build and Run the application using *npm start* command.

**Create a New React Application**

**Create the Post.js File**

In the src folder, create a new file named Post.js and add the following code:

class Post {

constructor(id, title, body) {

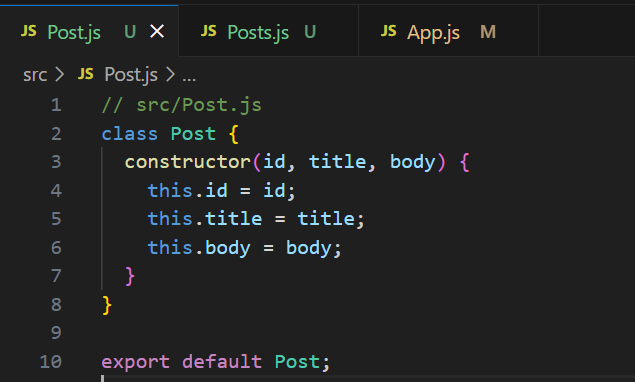
this.id = id;

this.title = title;

this.body = body;

}

}

export default Post;

**Create the Posts Component**

In the src folder, create another file named Posts.js and create a class-based component:

import React, { Component } from 'react';

import Post from './Post';

class Posts extends Component {

  constructor(props) {

    super(props);

    this.state = {

      posts: [],

      error: null

    };

  }

  loadPosts = async () => {

    try {

      const response = await fetch('https://jsonplaceholder.typicode.com/posts');

      const data = await response.json();

      const posts = data.map(p => new Post(p.id, p.title, p.body));

      this.setState({ posts });

    } catch (error) {

      this.setState({ error });

    }

  }

  componentDidMount() {

    this.loadPosts();

  }

  componentDidCatch(error, info) {

    alert("Something went wrong: " + error.toString());

  }

  render() {

    return (

      <div>

        <h1>All Posts</h1>

        {this.state.posts.map(post => (

          <div key={post.id}>

            <h3>{post.title}</h3>

            <p>{post.body}</p>

          </div>

        ))}

      </div>

    );

  }

}

export default Posts;



**App.js**

import React from 'react';

import './App.css';

import Posts from './Posts';

function App() {

return (

<div className="App">

<Posts />

</div>

);

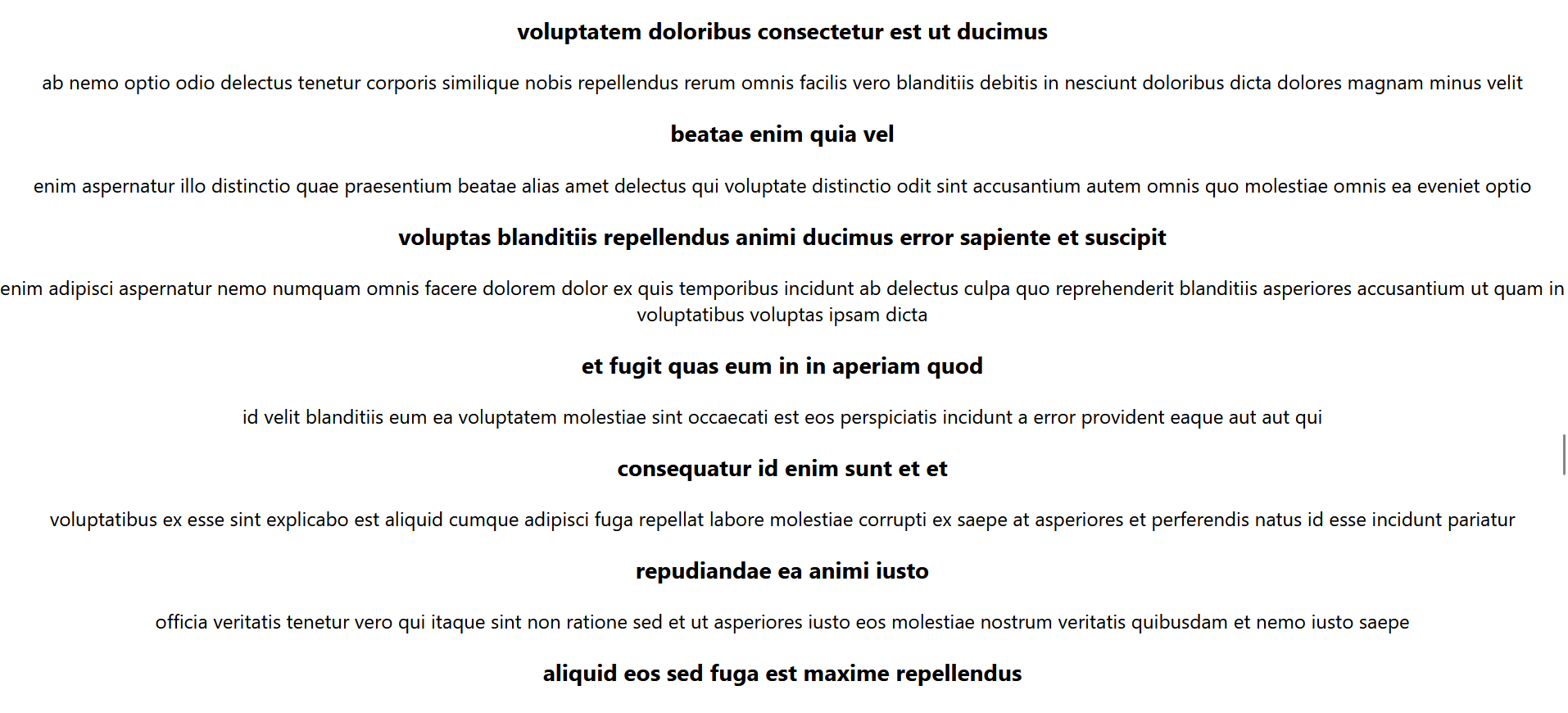
}

export default App;

**Run the Application:**

npm start

**OUTPUT:**



**5.Objectives**

* Understanding the need for styling react component
* Working with CSS Module and inline styles

In this hands-on lab, you will learn how to:

* Style a react component
* Define styles using the CSS Module
* Apply styles to components using className and style properties

**Prerequisites**

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

**Notes**

Estimated time to complete this lab: **30 minutes.**

My Academy team at Cognizant want to create a dashboard containing the details of ongoing and completed cohorts. A react application is created which displays the detail of the cohorts using react component. You are assigned the task of styling these react components.

Download and build the attached react application.



1. Unzip the react application in a folder
2. Open command prompt and switch to the react application folder
3. Restore the node packages using the following commands



*Figure 1: Restore packages*

1. Open the application using VS Code
2. Create a new CSS Module in a file called “CohortDetails.module.css”
3. Define a css class with the name as “box” with following properties

*Width = 300px;*

*Display = inline block;*

*Overall 10px margin*

*Top and bottom padding as 10px*

*Left and right padding as 20px*

*1 px border in black color*

*A border radius of 10px*

1. Define a css style for html <dt> element using tag selector. Set the font weight to 500.
2. Open the cohort details component and import the CSS Module
3. Apply the box class to the container div
4. Define the style for <h3> element to use “green” color font when cohort status is “ongoing” and “blue” color in all other scenarios.
5. Final result should look similar to the below image



1. **Create a New React Application**

Use the following command in your terminal to create a new React app named blogapp:

cd \*\*\*

npm install

**2.Open the Project in VS Code**

Code .

**3.Create CSS Module**

Inside src/, create a new file:

**4.CohortDetails.module.css**

.box {

width: 300px;

display: inline-block;

margin: 10px;

padding: 10px 20px;

border: 1px solid black;

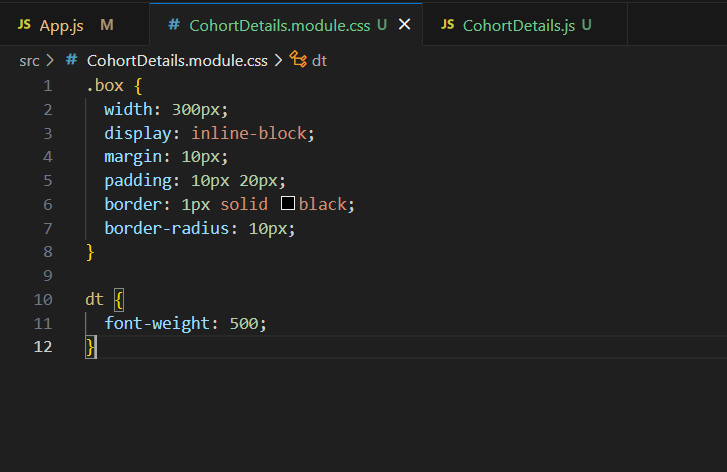
border-radius: 10px;

}

dt {

font-weight: 500;

}



**4.CohortDetails.js**

import React from 'react';

import styles from './CohortDetails.module.css';

const CohortDetails = ({ cohort }) => {

const headingColor = cohort.status.toLowerCase() === 'ongoing' ? 'green' : 'blue';

return (

<div className={styles.box}>

<h3 style={{ color: headingColor }}>{cohort.name}</h3>

<dl>

<dt>Started On</dt>

<dd>{cohort.startDate}</dd>

<dt>Current Status</dt>

<dd>{cohort.status}</dd>

<dt>Coach</dt>

<dd>{cohort.coach}</dd>

<dt>Trainer</dt>

<dd>{cohort.trainer}</dd>

</dl>

</div>

);

};

export default CohortDetails;

**5. App.js**

import React from 'react';

import CohortDetails from './CohortDetails';

const data = [

{

name: 'INTADMDF10 -.NET FSD',

startDate: '22-Feb-2022',

status: 'Scheduled',

coach: 'Aathma',

trainer: 'Jojo Jose',

},

{

name: 'ADM21JF014 - Java FSD',

startDate: '10-Sep-2021',

status: 'Ongoing',

coach: 'Apoorv',

trainer: 'Elisa Smith',

},

{

name: 'CDBJF21025 - Java FSD',

startDate: '24-Dec-2021',

status: 'Ongoing',

coach: 'Aathma',

trainer: 'John Doe',

},

];

function App() {

return (

<div style={{ border: '1px solid black', padding: '20px' }}>

<h2>Cohorts Details</h2>

{data.map((cohort, index) => (

<CohortDetails key={index} cohort={cohort} />

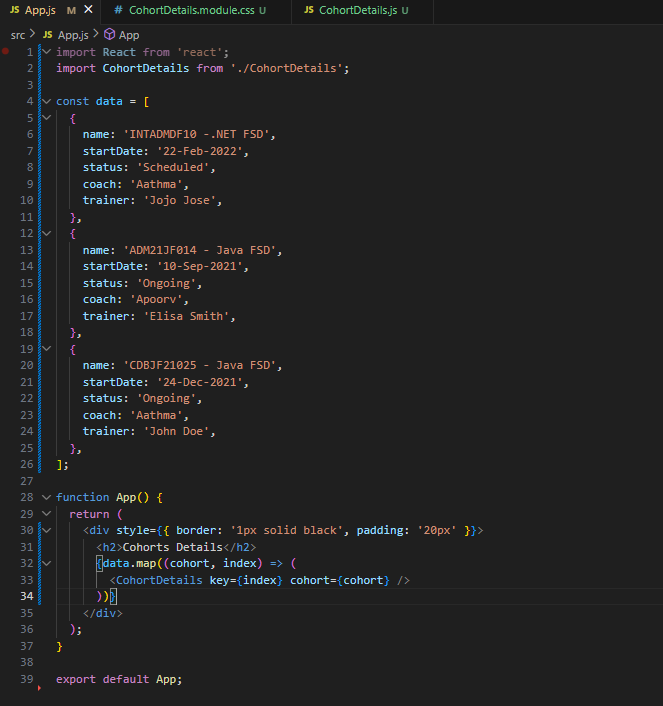
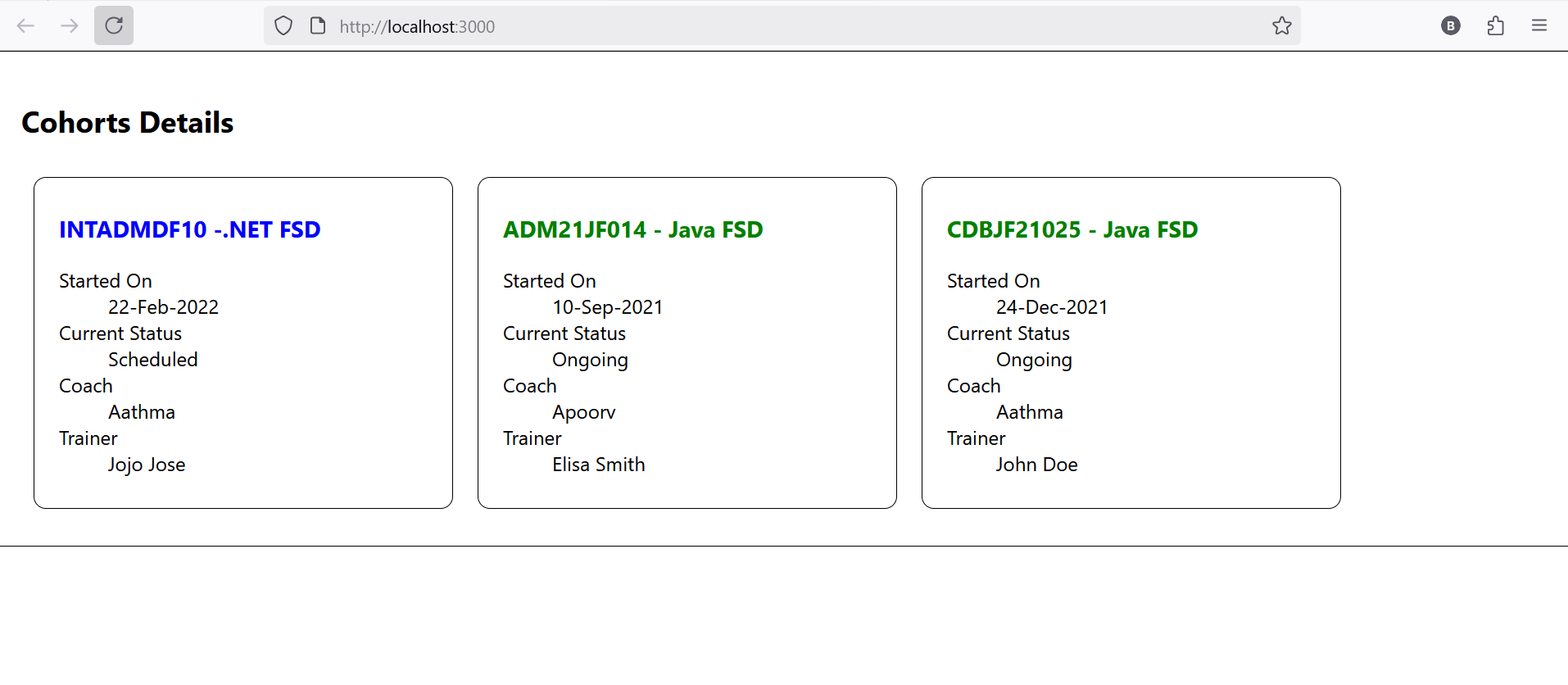
))}

</div>

);

}

export default App;

**OUPUT:**