**WEEK 6 ASSIGNMENT**

1. **ReactJS-HOL**

**Lab Title:**

Creating a React Application - “myfirstreact”

**Objective:**

To create a new React application using create-react-app and display the heading:  
“Welcome to the first session of React”

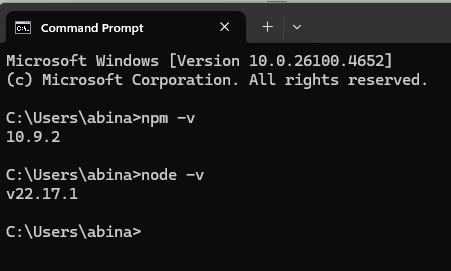
**Tools Used:**

- Node.js  
- npm (Node Package Manager)  
- React (via create-react-app)  
- VS Code (or any code editor)  
- Command Prompt (Terminal)

**Steps Followed:**

**Step 1: Installed Node.js and npm**

Installed from https://nodejs.org/en/download



**Step 2: Installed create-react-app globally**

Command:  
npm install -g create-react-app

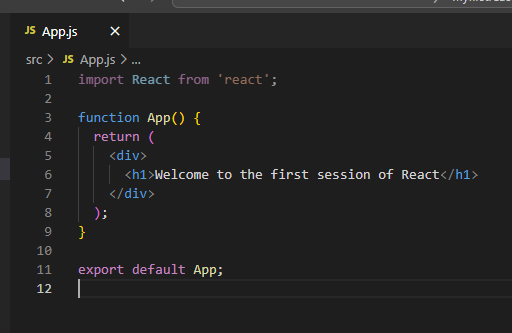
**Step 3: Created React project**

Command:  
npx create-react-app myfirstreact

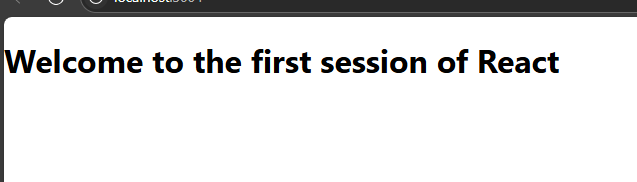
**Step 4: Ran the React App**

Commands:  
cd myfirstreact  
npm start

**Step 5: Modified App.js**



**Step 6: Final Output**



1. **ReactJS-HOL**

**Lab Title: Student Management Portal Using React Components**

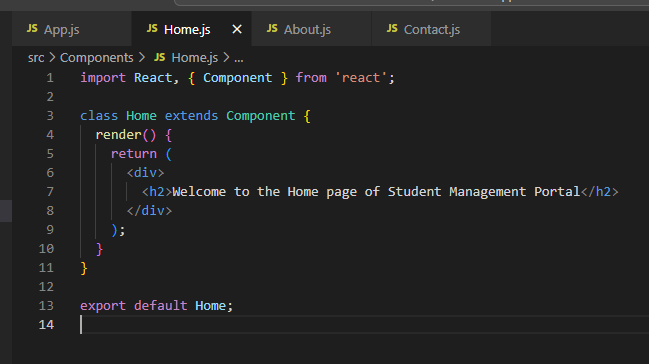
**Objective:**Create a React application named Student App with three separate components: Home, About, and Contact. Each component should display a unique welcome message. Integrate all three components in the main App.js

### ****Steps Followed:****

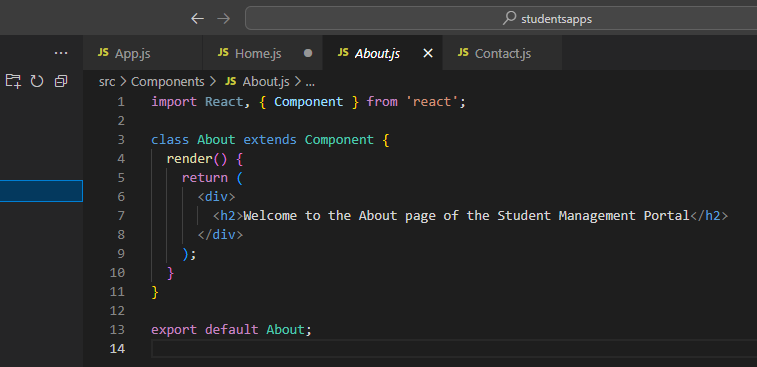
****Step 1:** Create a React project named** Student App**:**  
Open your terminal or Visual Studio Code terminal and run the following command:

****Step 2:** Create a folder named** Components **inside the** src **folder:**Inside the src directory of your project, create a new folder and name it Components.

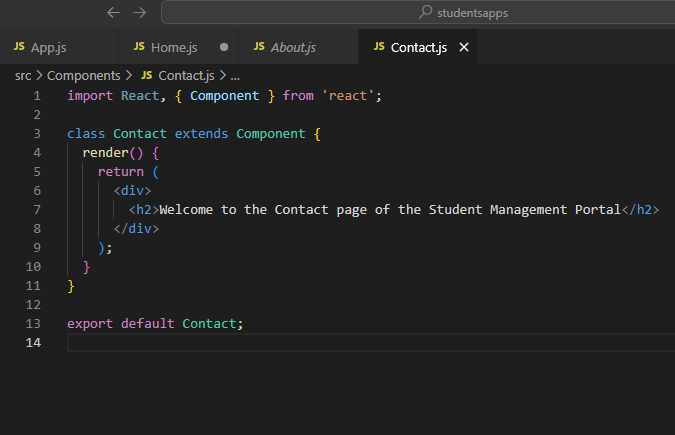
****Step 3:** Create** Home.js **component inside the** Components **folder:**



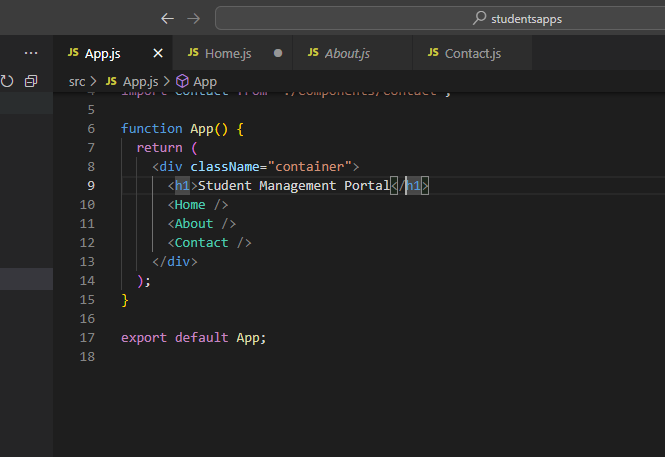
**Step 4:**  Create About.js component inside the same Components folder:



**Step 5:** Create Contact.js component in the same folder:

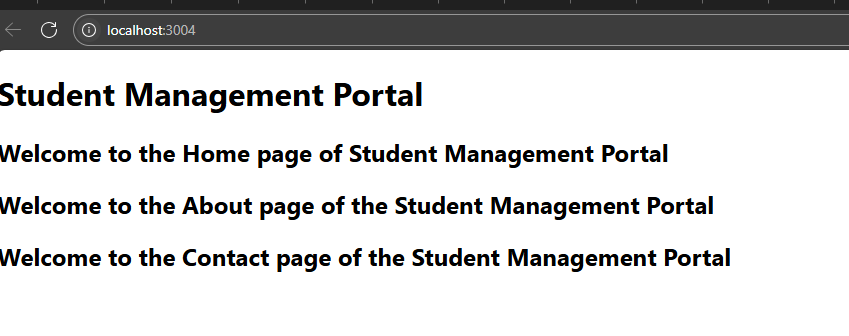


****Step 6:** Edit** App.js **to include and render the components:**  
Replace the content of src/App.js with the following:



**Step 7: Run the application:**  
Open terminal inside your project folder and run.

**Step 8: View the result in browser:**  
After successful compilation, open your browser and visit:



1. **ReactJS-HOL**

### ****Lab Title: Creating a Score Calculator App Using React Functional Component****

**Objective:**  
Build a React application to calculate and display the average score of a student using a functional component, accepting inputs like Name, School, Total Score, and Goal. Apply styling using an external CSS file.

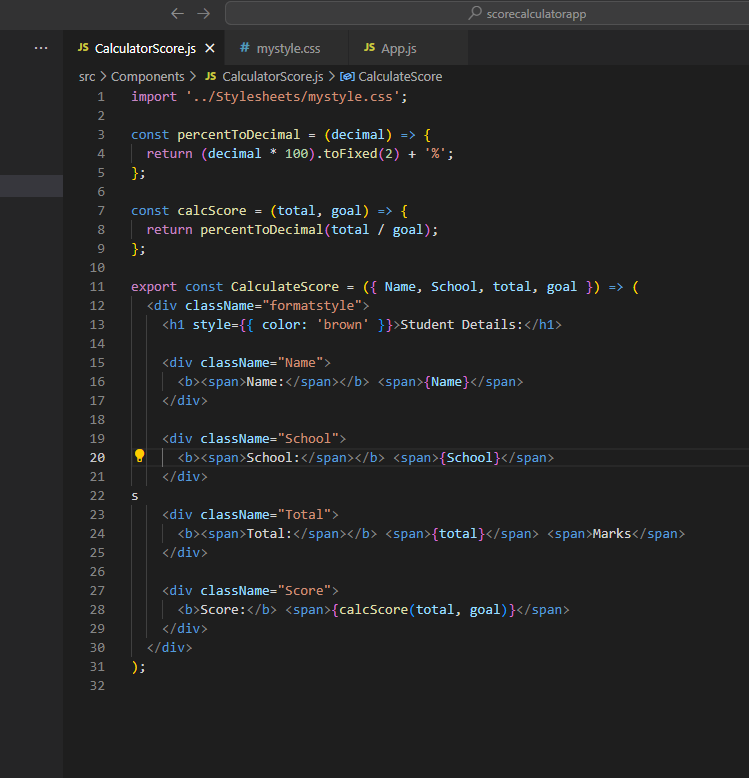
### ****Steps Followed:****

****Step 1:** Create a React project named** score calculator app**:  
Open Visual Studio Code terminal and type the following command.**

****Step 2:** Navigate to the project directory.**

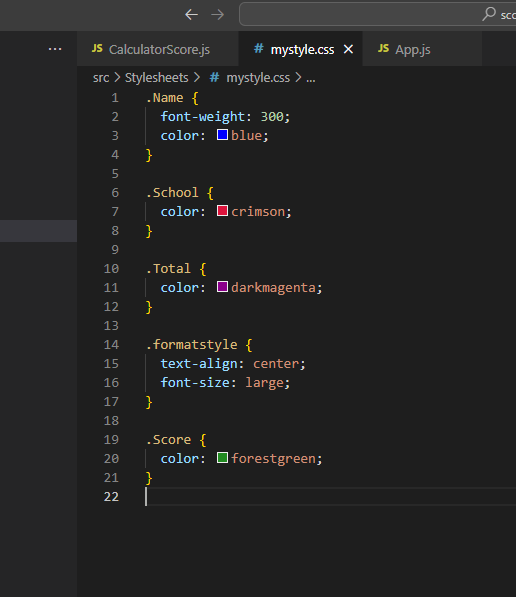
****Step 3:** Inside the** src **folder, create a new folder named** Components**:**

**Step 4 :** Type the following code inside Calculate Score.js:

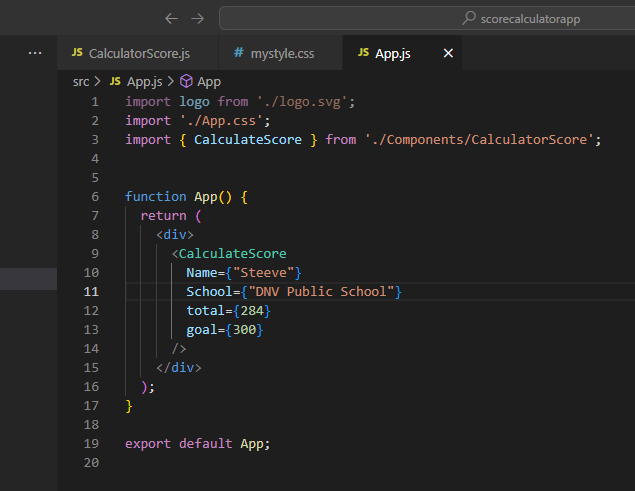


****Step 5:** Inside** src**, create a folder named** Stylesheets**:**  
Inside this folder, create a file .

**Step 6:** Add the following CSS styles in my style.css:

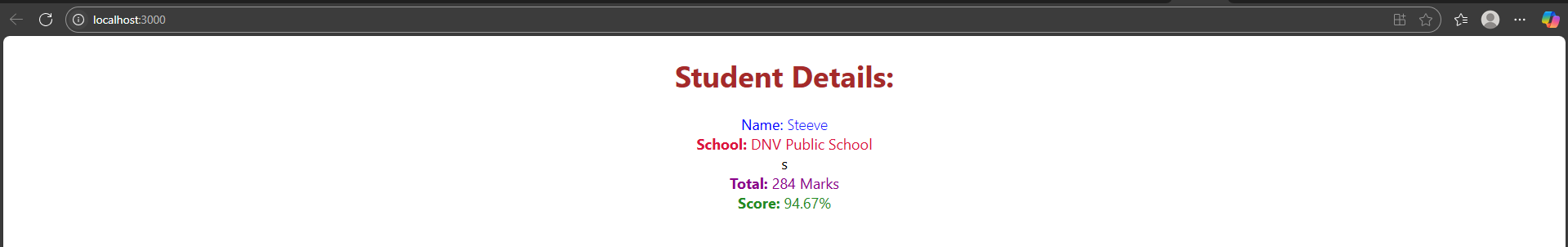


**Step 7:** Edit App.js to call the Calculate Score component:



**Step 8 :** Start the application using the following command in terminal:

****Step 9 :** Open your browser and go to:**



1. **React JS-HOL**

**Lab Title: Working with Component Life cycle Methods in React**

**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

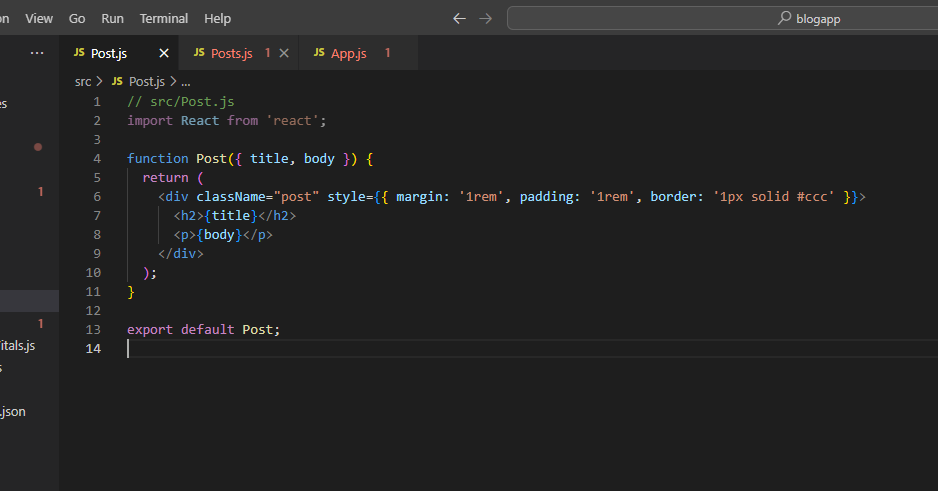
Estimated time to complete this lab: 60 minutes.

## Steps Followed:

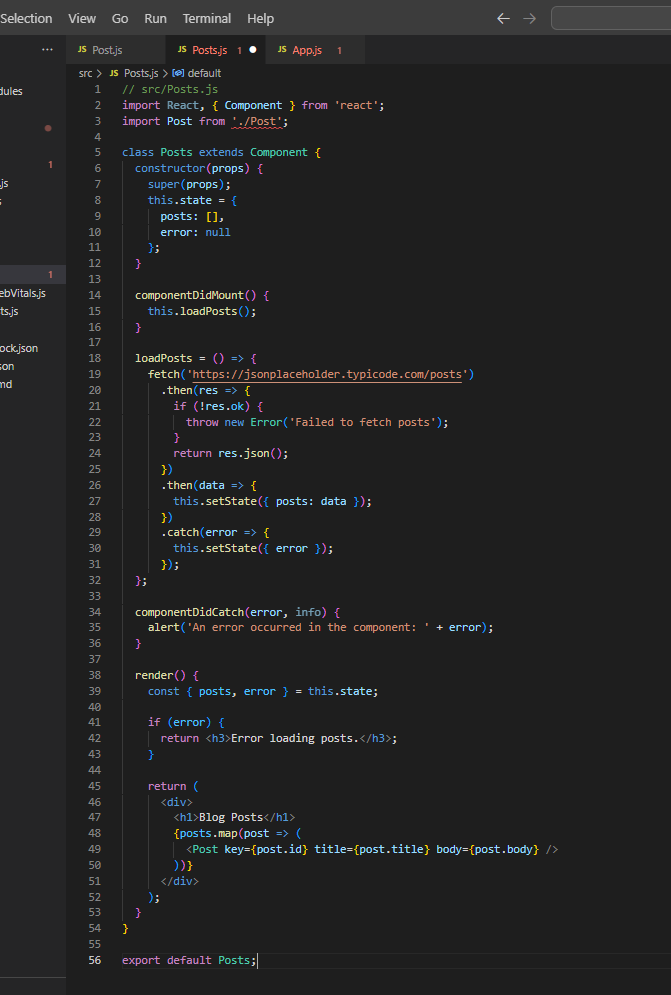
**Step 1**: Create a new react application using create-react-app tool with the name as “blog app”.

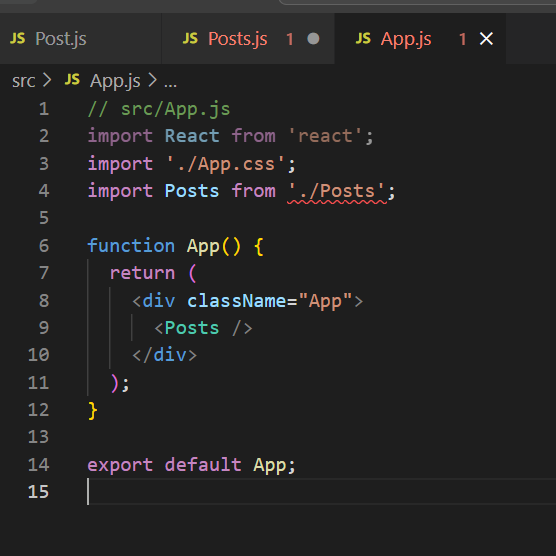
**Step 2:** Open the application using VS Code.

**Step 3 :** Create a new file named as Post.js in src folder with following properties (Figure 1: Post class).



**Step 4:** Create a new class based component named as Posts inside Posts.js file (Figure 2: Posts Component).





**Step 5 :** Initialize the component with a list of Post in state of the component using the constructor.

**Step 6:** 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 .

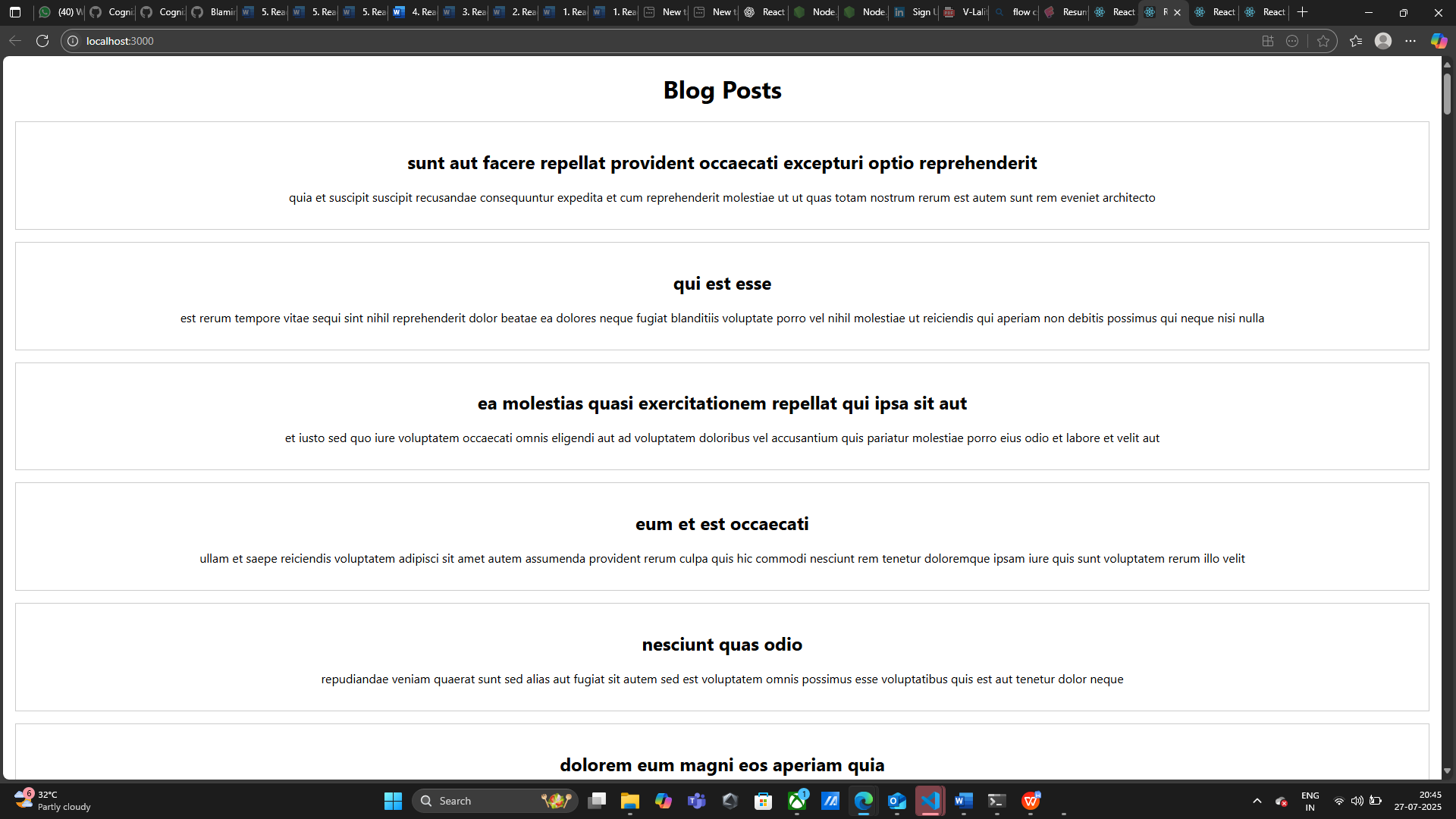
**Step 7:** Implement the componentDidMount() hook to make calls to loadPosts() which will fetch the posts .

**Step 8:** Implement the render() which will display the title and post of posts in HTML page using heading and paragraphs respectively .

**Step 9:** Define a componentDidCatch() method which will be responsible for displaying any error happening in the component as alert messages .

**Step 10:** Add the Posts component to App component.

**Step 11:** Build and Run the application using npm start command.



1. **ReactJS-HOL**

**Lab Title: Cohort Tracker with Conditional Styling in React**

**Objective:**• Create a React component to display cohort details.  
• Apply CSS Modules for scoped and conditional styling.  
• Render cohorts row by row in individual styled boxes.

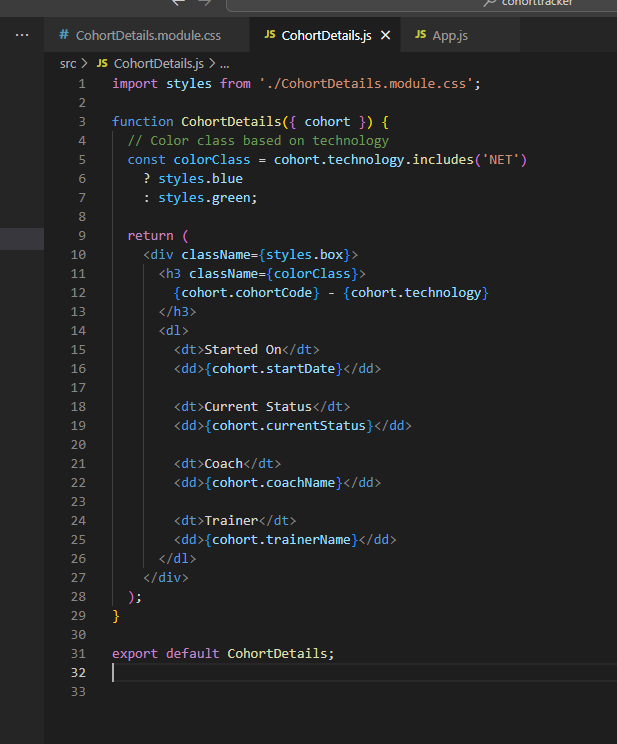
**Steps Followed:**

**Step 1:** Create a React project using the command:  
  **npx create-react-app cohorttracker**

**Step 2:** Open the project in Visual Studio Code.

**Step 3:** Under the 'src' folder, create a new file named 'CohortDetails.js'.

**Step 4:**  Add the following code to 'CohortDetails.js':

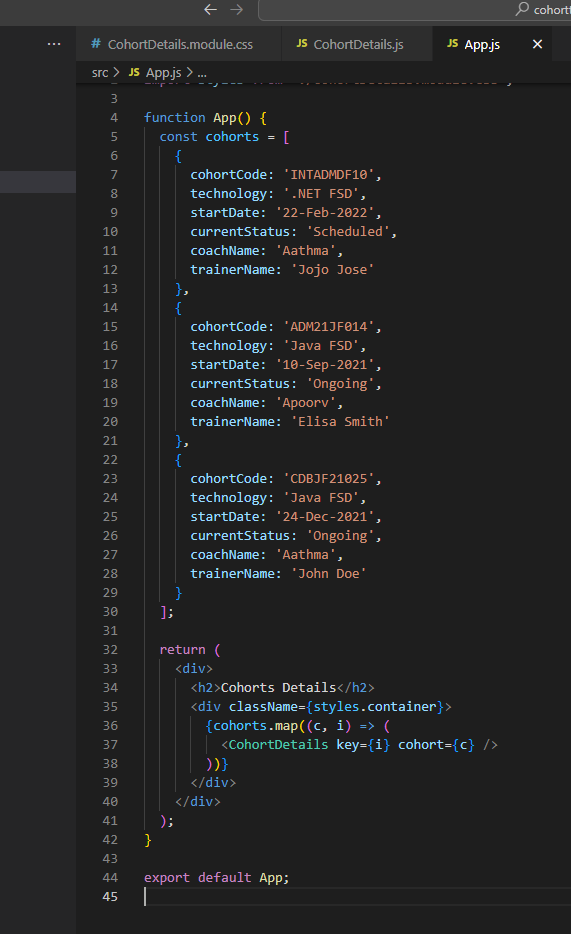


**Step 5:**  Under 'src', create a CSS Module file named 'CohortDetails.module.css'.

**Step 6:** Add the following styles to 'CohortDetails.module.css':



**Step 7:** In 'App.js', import and use multiple 'CohortDetails' components by passing cohort objects as props.



**Step 8:** Run the application using the command: npm start

**Step 9:** Open browser and go to: http://localhost:3000 to view the styled cohort cards.

