**Week 6 Solutions -REACT . JS**

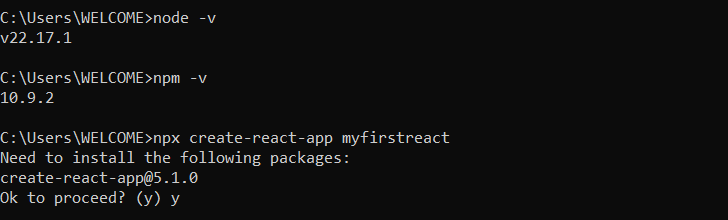
1. **ReactJS-HOL**

Create a new React Application with the name “myfirstreact”, Run the application to print “welcome to the first session of React” as heading of that page.

Step 1: To create a new React app, Install Nodejs and Npm from the following link:

<https://nodejs.org/en/download/>

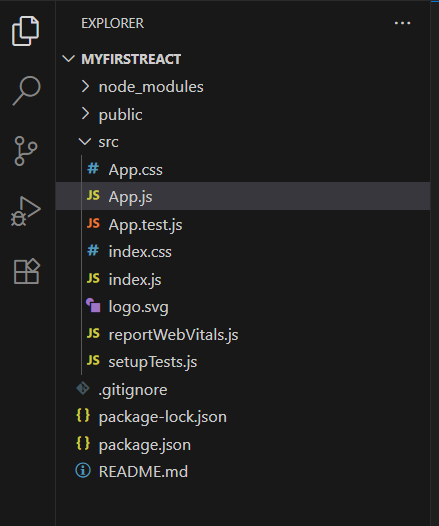
step 2: To create a React Application with the name of “myfirstreact”, type the following command:



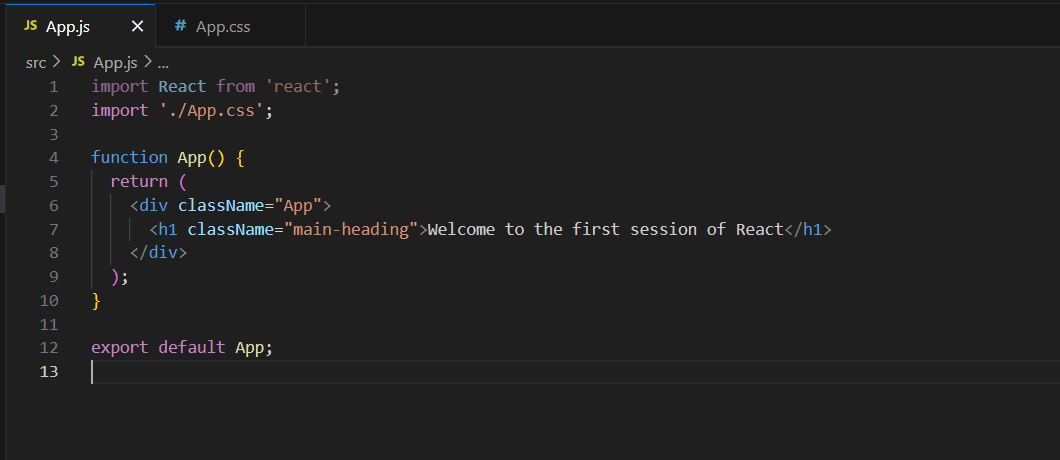
Step 3: Once the App is created, navigate into the folder of myfirstreact by typing the following command:



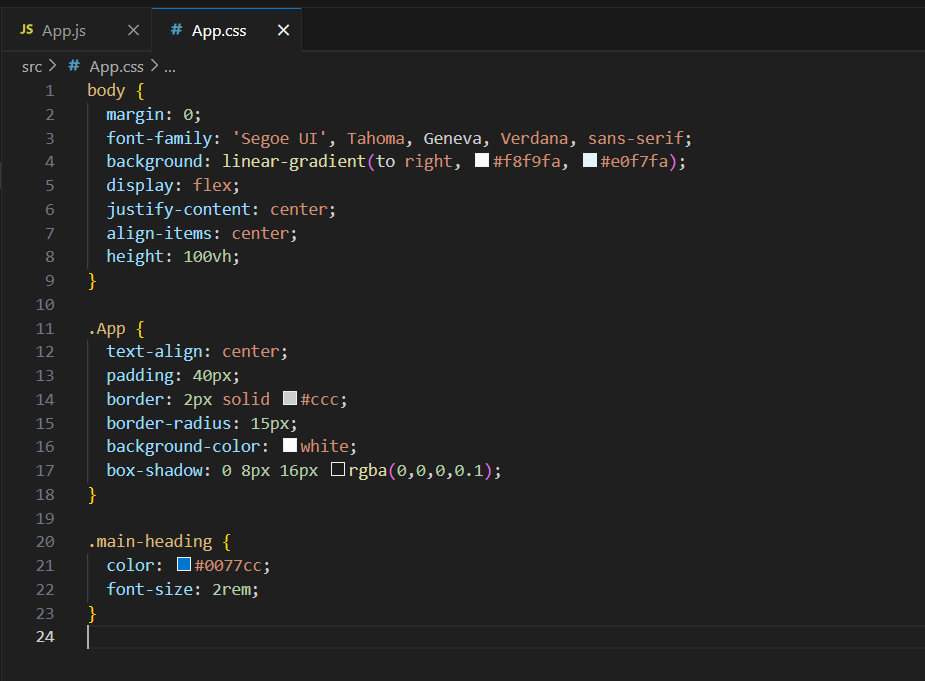
Step 4: Open the folder of myfirstreact in Visual Studio Code



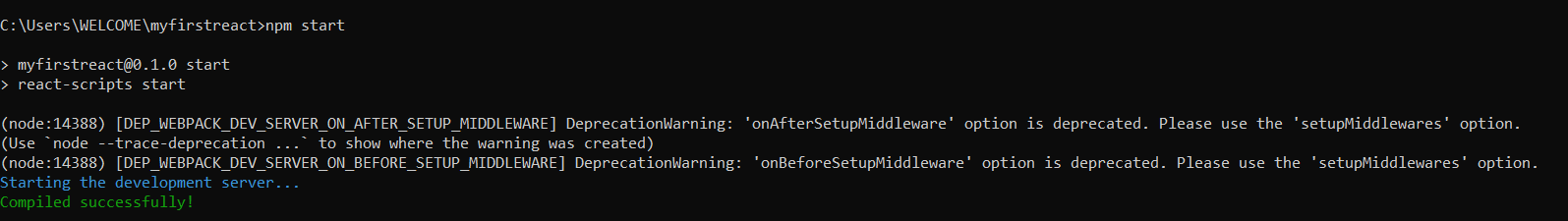
Step 5: Open the App.js file in Src Folder of myfirstreact. Update the Code:



Step 6: Open the App.css file in Src Folder of myfirstreact. Update the Code:

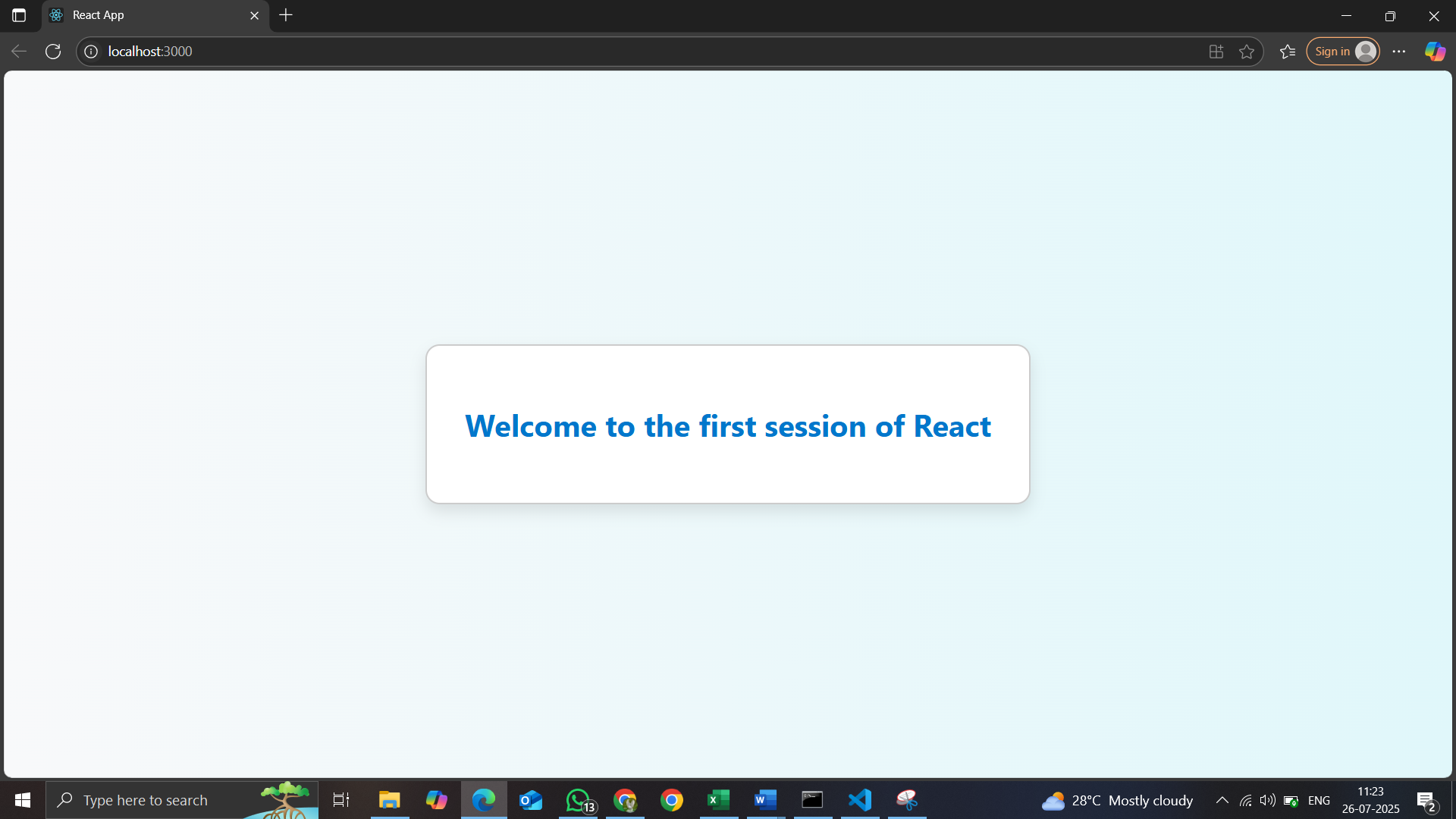


Step 7: Run the following command to execute the React application:



Step 8: Open a new browser window and type “localhost:3000” in the address bar

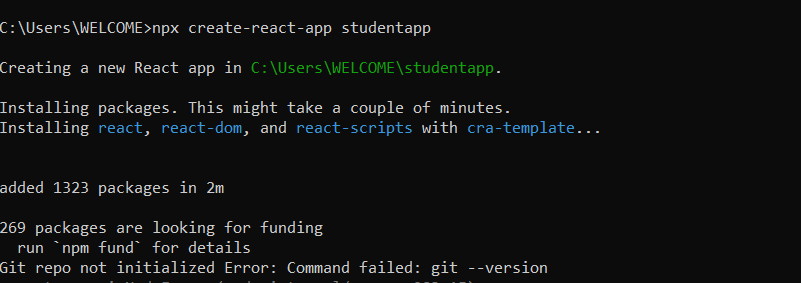
**Output:**



1. **ReactJS-HOL**

Create a react app for Student Management Portal named StudentApp and create a component named Home which will display the Message “Welcome to the Home page of Student Management Portal”. Create another component named About and display the Message “Welcome to the About page of the Student Management Portal”. Create a third component named Contact and display the Message “Welcome to the Contact page of the Student Management Portal”. Call all the three components.

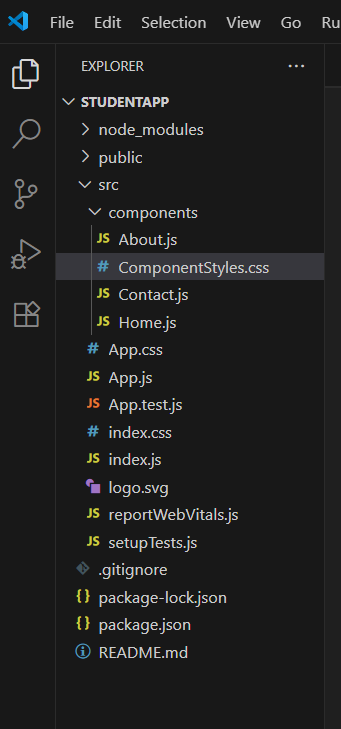
Step 1: To create a React Application with the name of “studentapp”, type the following command:



Step 2: Once the App is created, navigate into the folder of studentapp by typing the following command:



Step 3: Open the folder of studentapp in Visual Studio Code



Step 4: Create js files like Home,about,contact…

src/

│

├── components/

│ ├── Home.js

│ ├── About.js

│ └── Contact.js

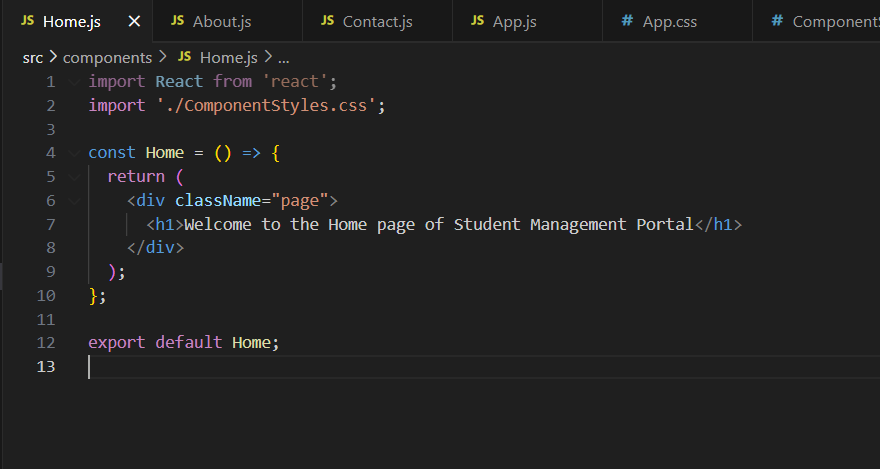
│

├── App.js

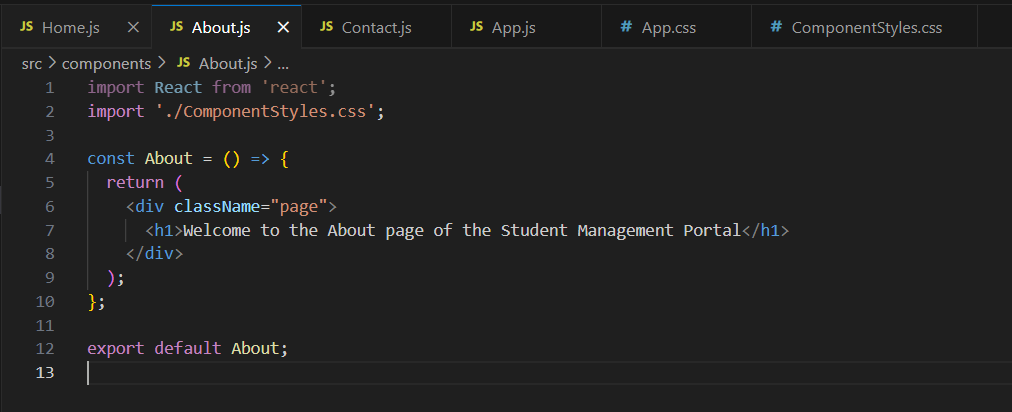
├── App.css

└── index.js

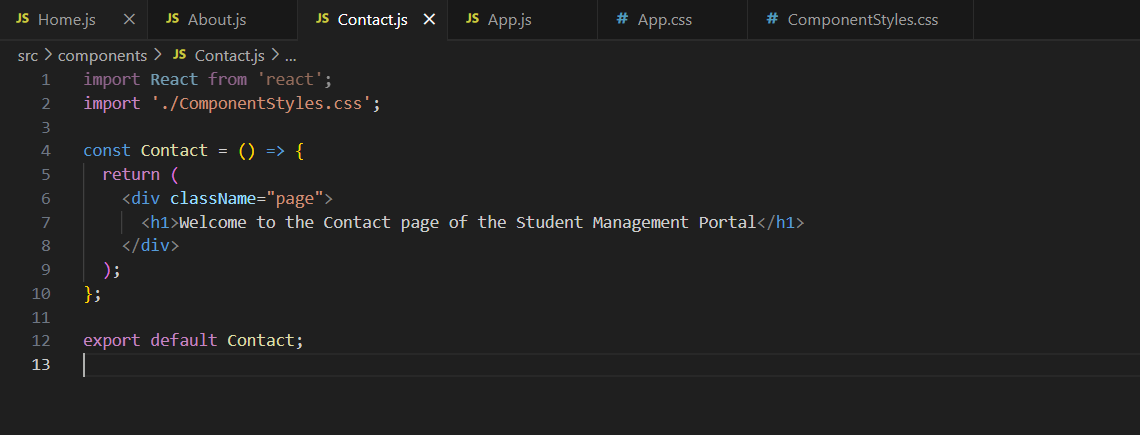
Step 5: Open the Home.js file in Src Folder of studentapp .Write the Code:



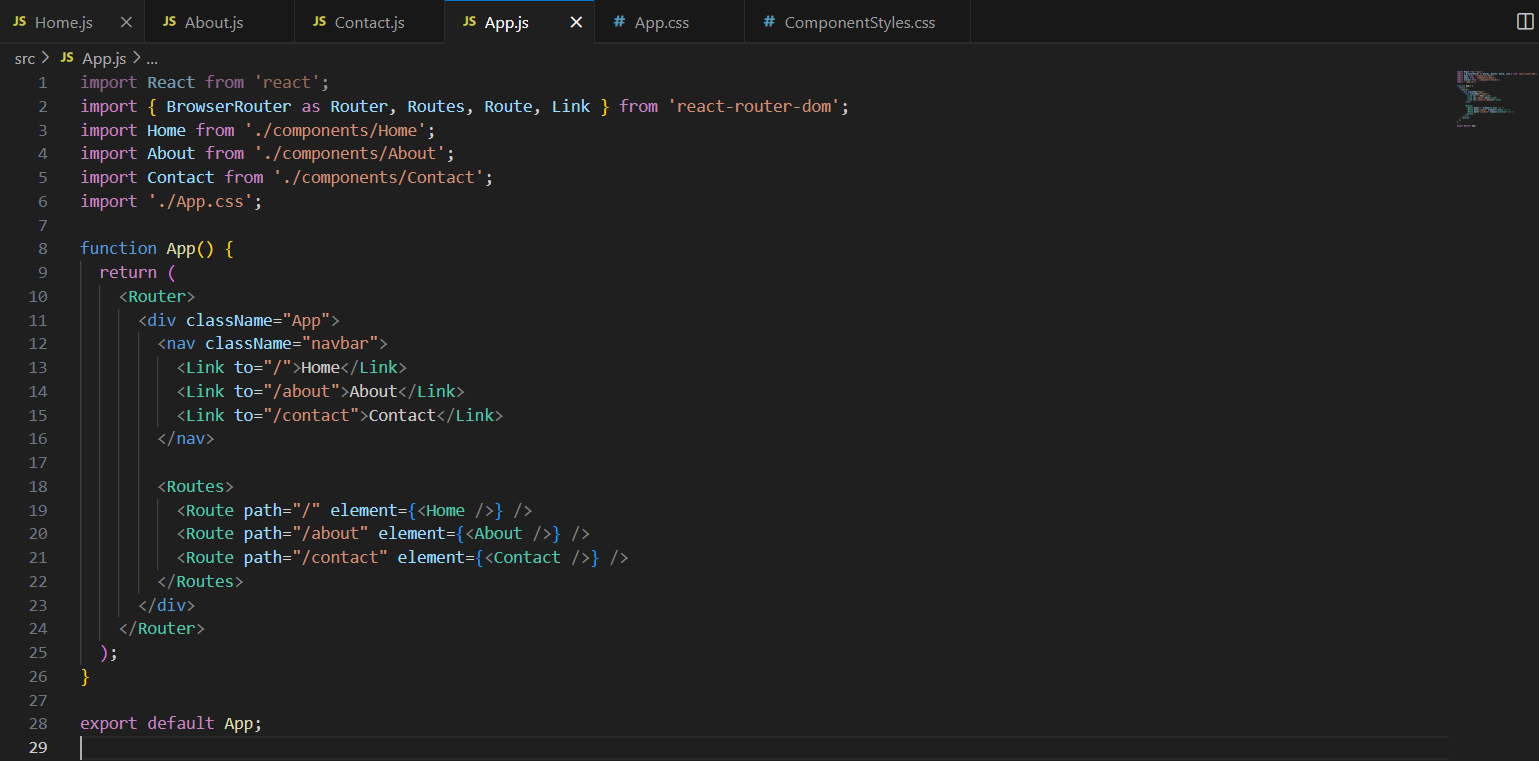
Step 6: Open the About.js file in Src Folder of studentapp .Write the Code.



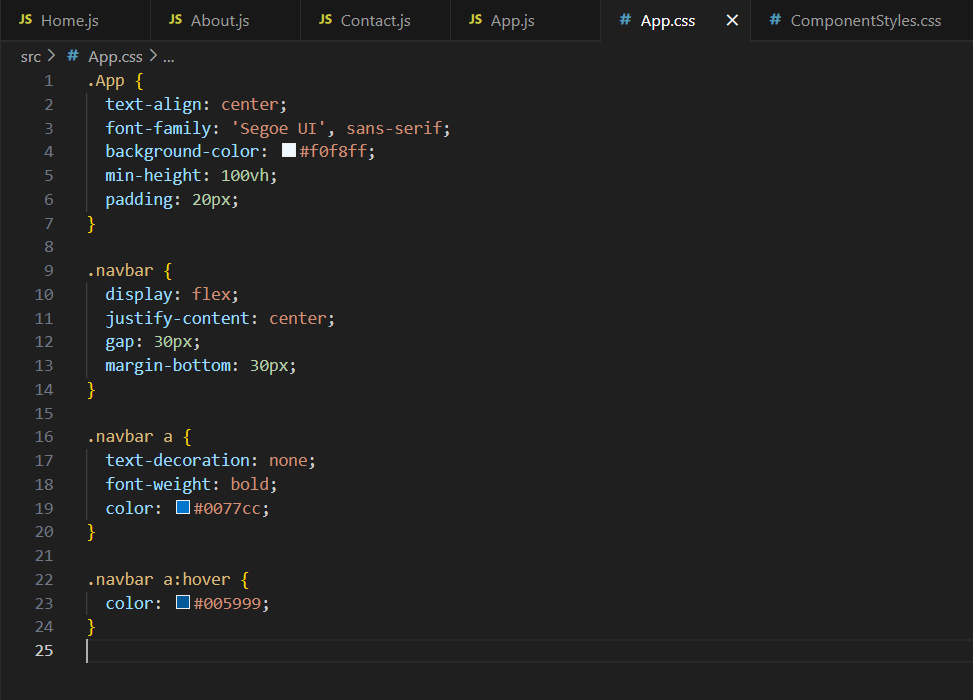
Step 7: Open the Contact.js file in Src Folder of studentapp .Write the Code



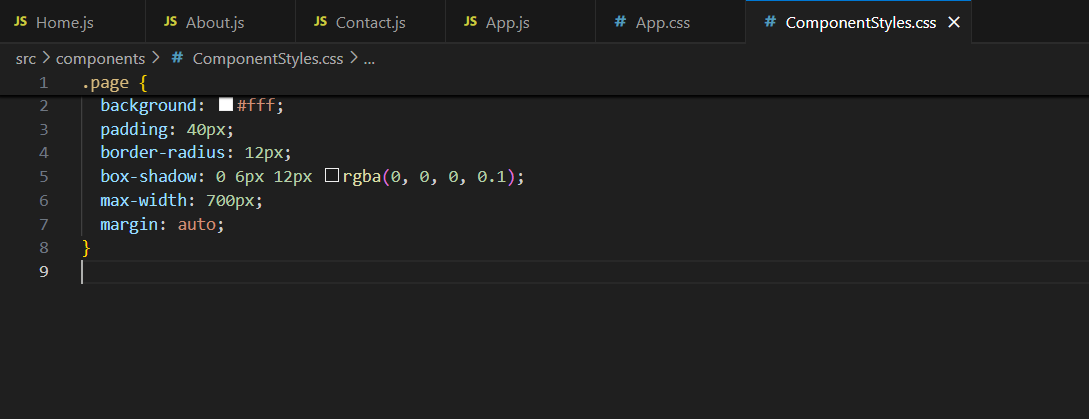
Step 8: Open the App.js file in Src Folder of studentapp.Update the Code:



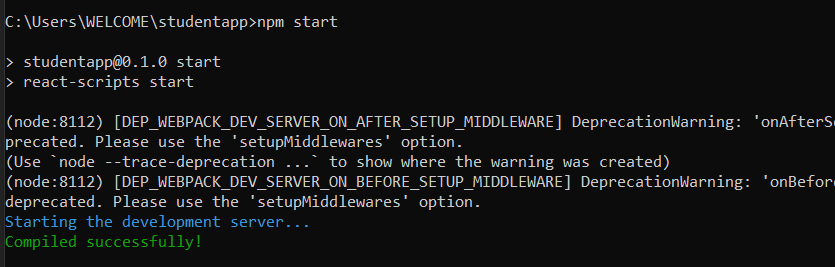
Step 9: Open the App.css file in Src Folder of studentapp. Update the Code:



Step 10: Open the ComponentStyles.css file in Src Folder of studentapp. write the Code:



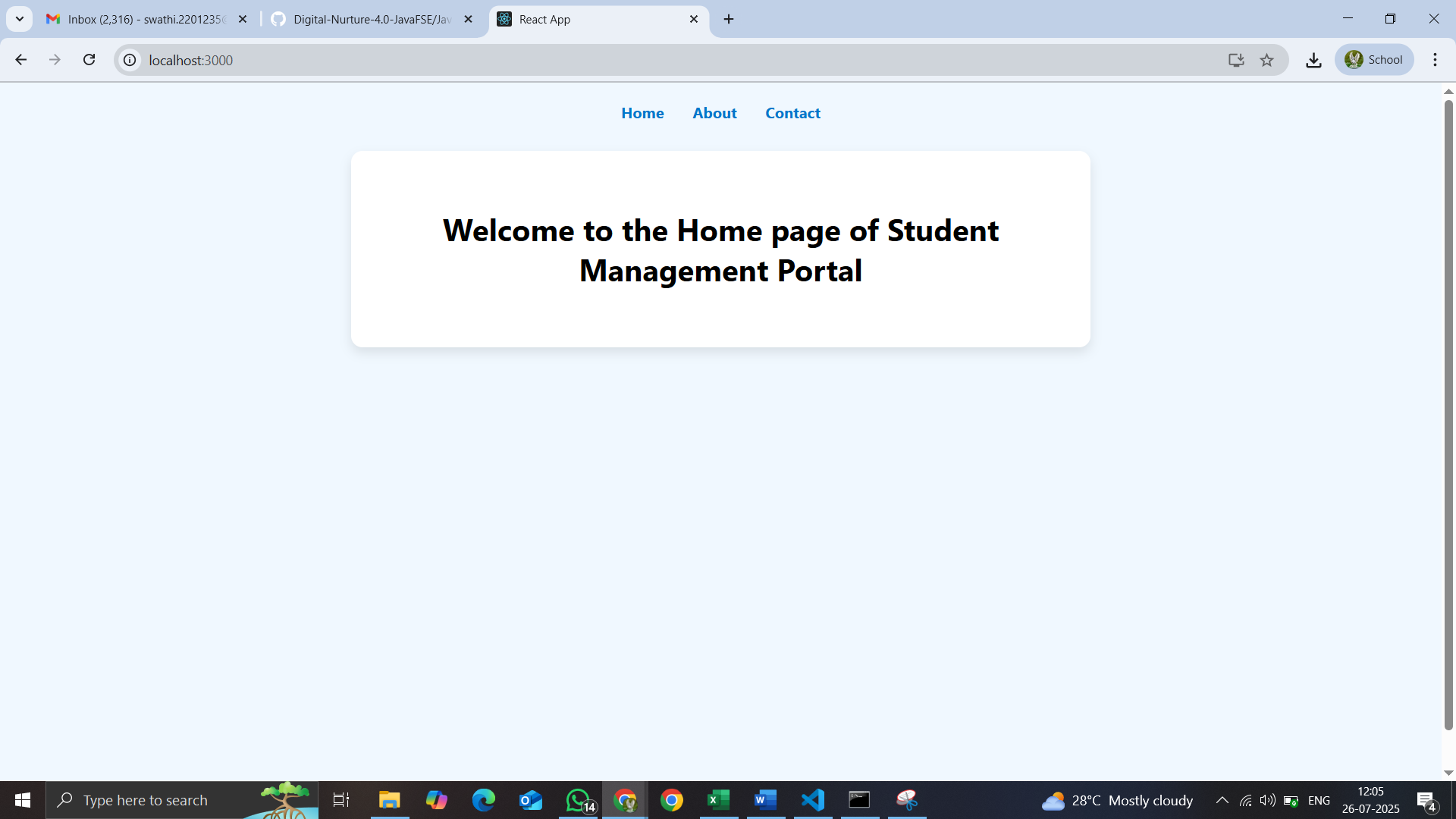
Step 11: Run the following command to execute the React application:



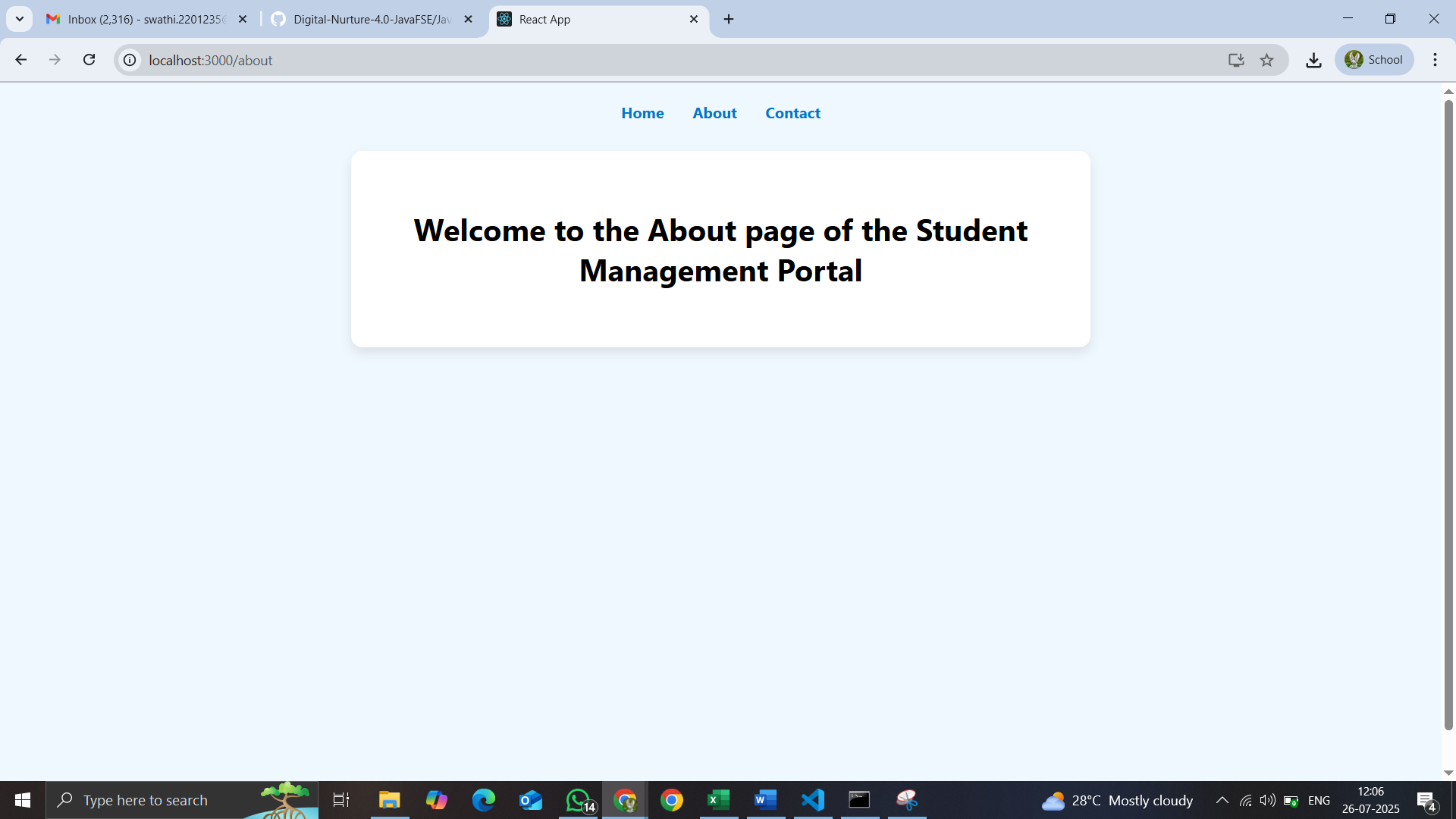
Step 12: Open a new browser window and type “localhost:3000” in the address bar

**Output:**

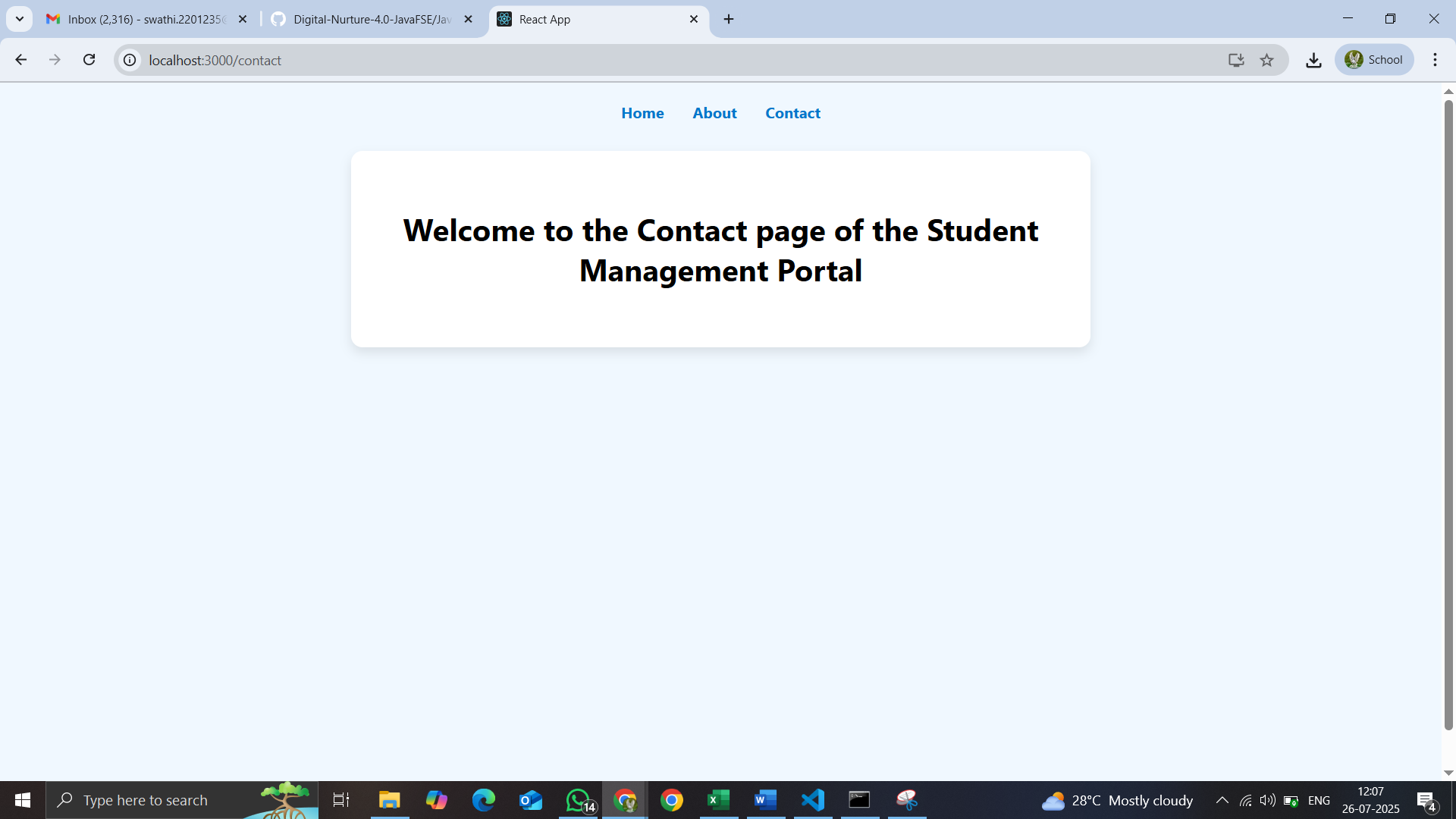
**Home:**



**About:**



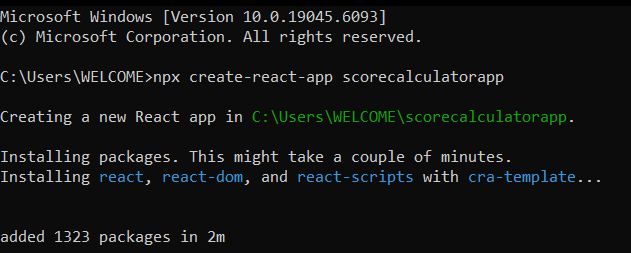
**Contact:**



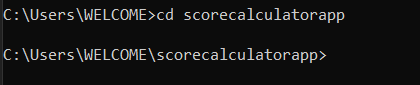
1. **ReactJS-HOL**

Create a react app for Student Management Portal named scorecalculatorapp and create a function component named “CalculateScore” which will accept Name, School, Total and goal in order to calculate the average score of a student and display the same.

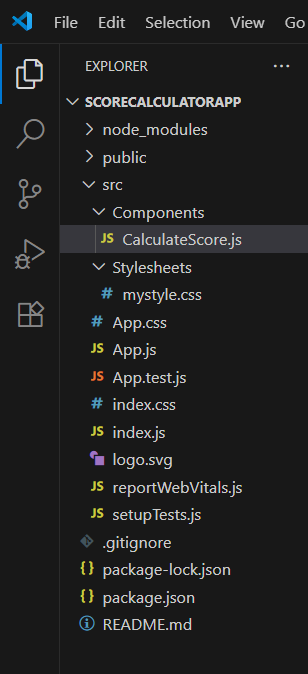
Step 1: To create a React Application with the name of “scorecalculatorapp”, type the following command:



Step 2: Once the App is created, navigate into the folder of scorecalculatorapp by typing the following command:



Step 3: Open the folder of scorecalculatorapp in Visual Studio Code



Step 4: Create js and css files CalculateScore.js in Components and mystyle.css in Stylesheets…

src/

│

├── Stylesheets/

│ ├── mystyle.css

├── Components/

│ ├── CalculateScore.js

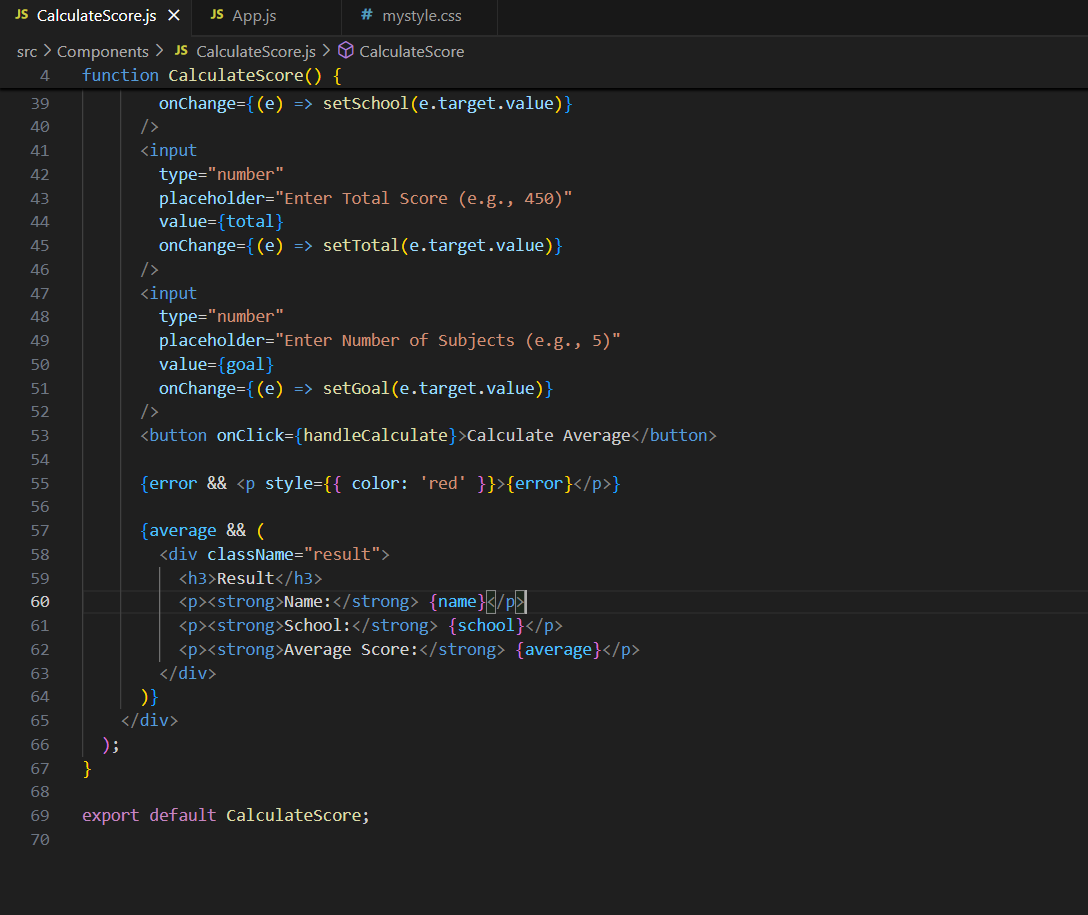
│

├── App.js

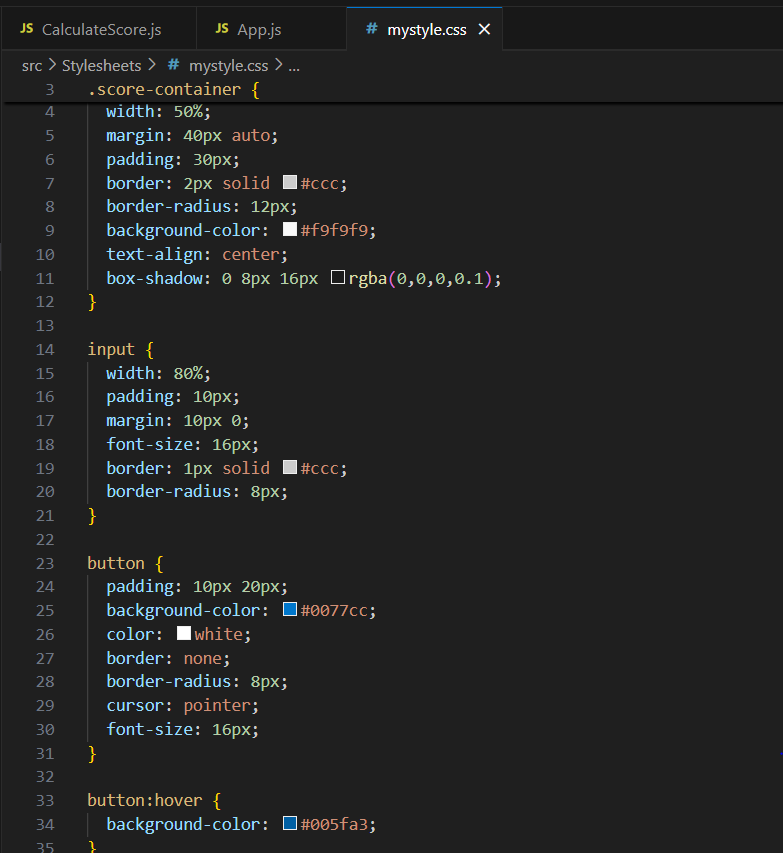
├── App.css

└── index.js

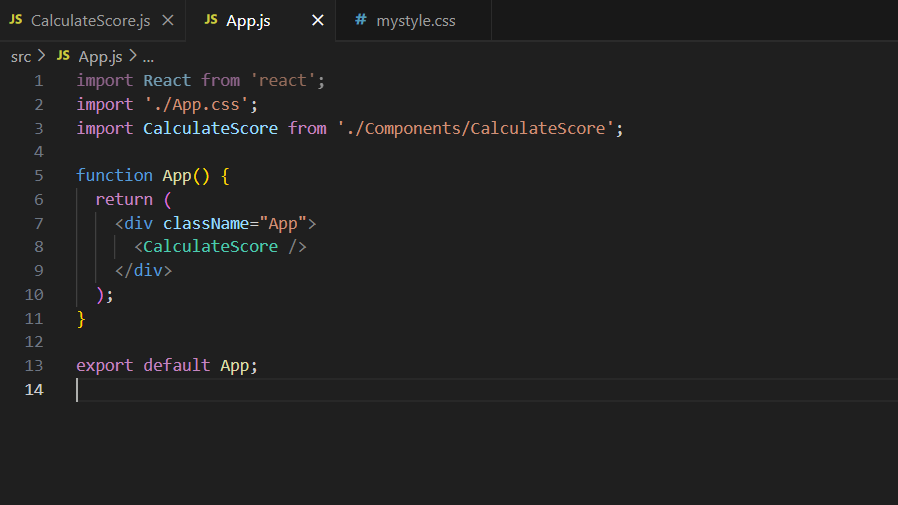
Step 5: Open the CalculateScore.js file in Src Folder of scorecalculatorapp.Write the Code:



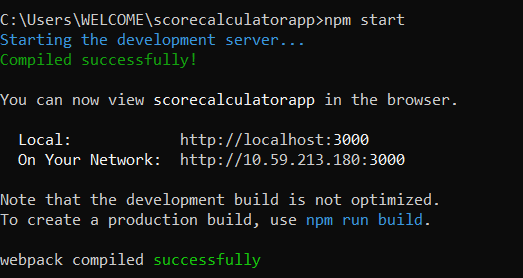
Step 6: Open the mystyle.css file in Src Folder of scorecalculatorapp .Write the Code.



Step 7: Open the App.js file in Src Folder of scorecalculatorapp .Write the Code

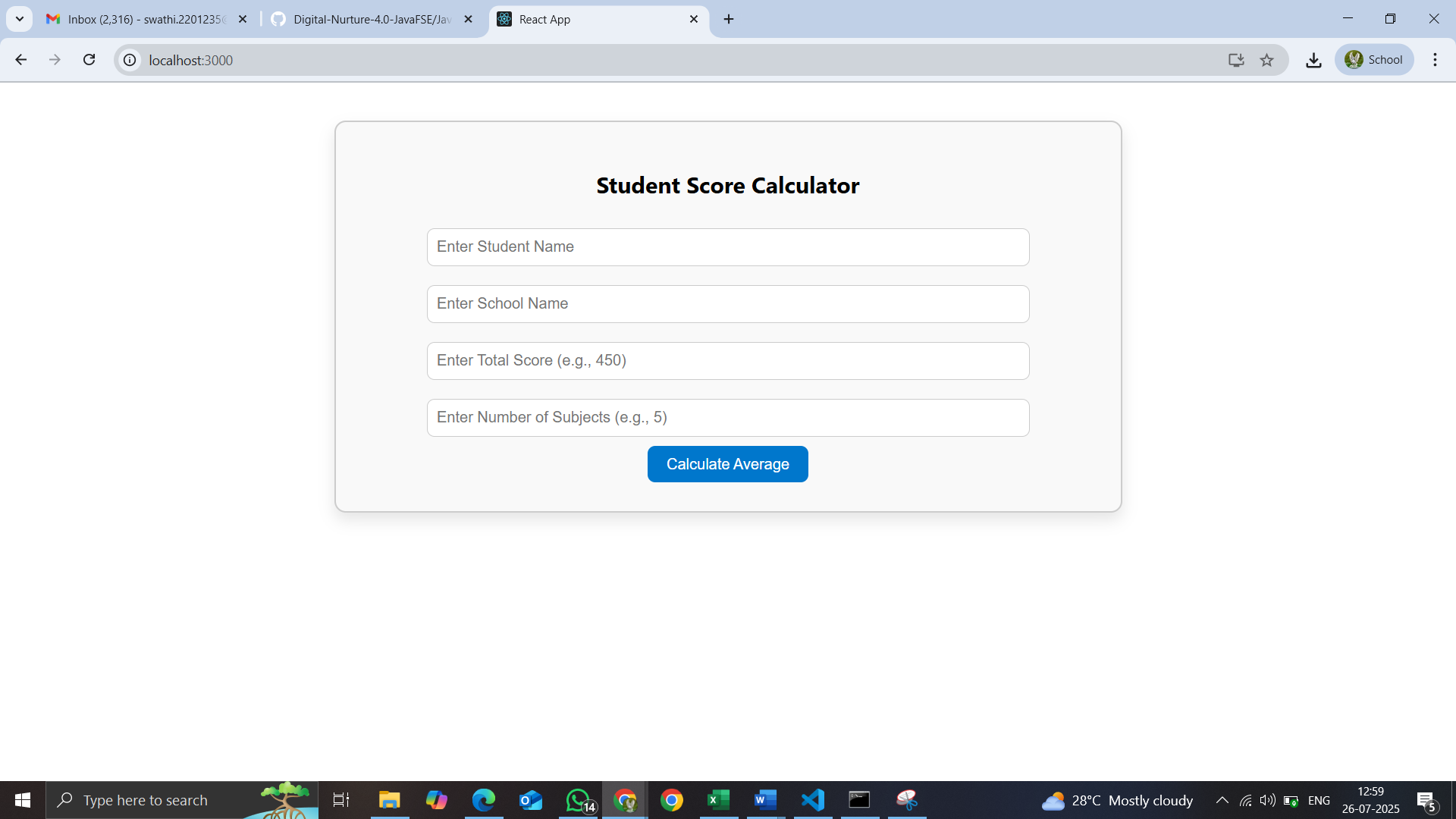


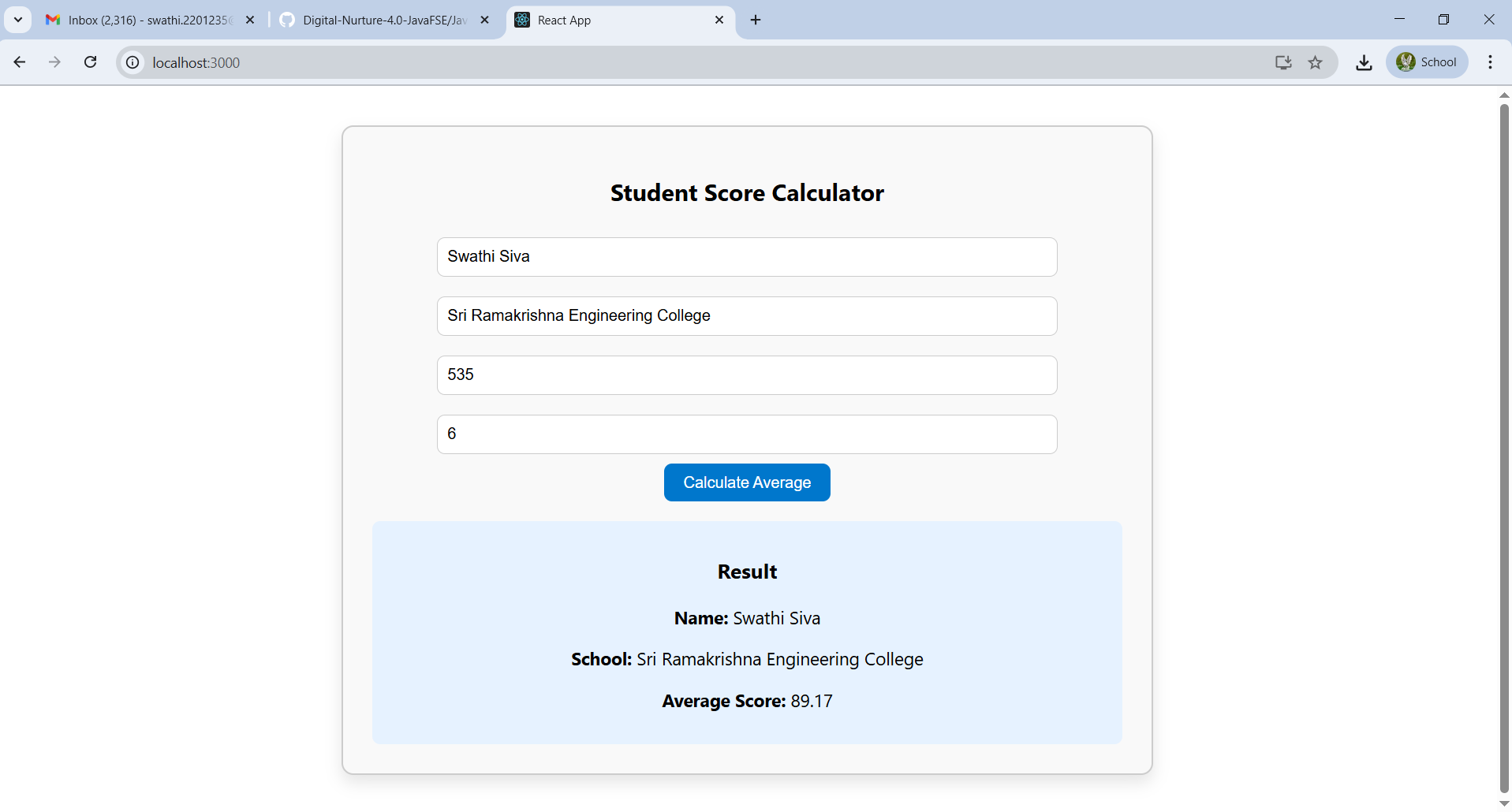
Step 8: Run the following command to execute the React application:



Step 9: Open a new browser window and type “localhost:3000” in the address bar

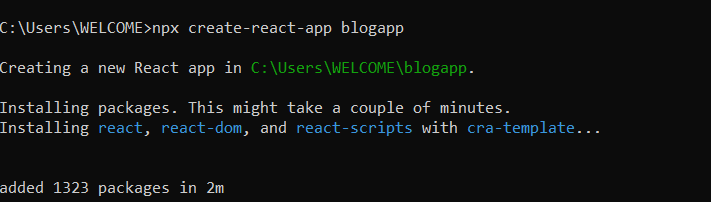
**Output:**



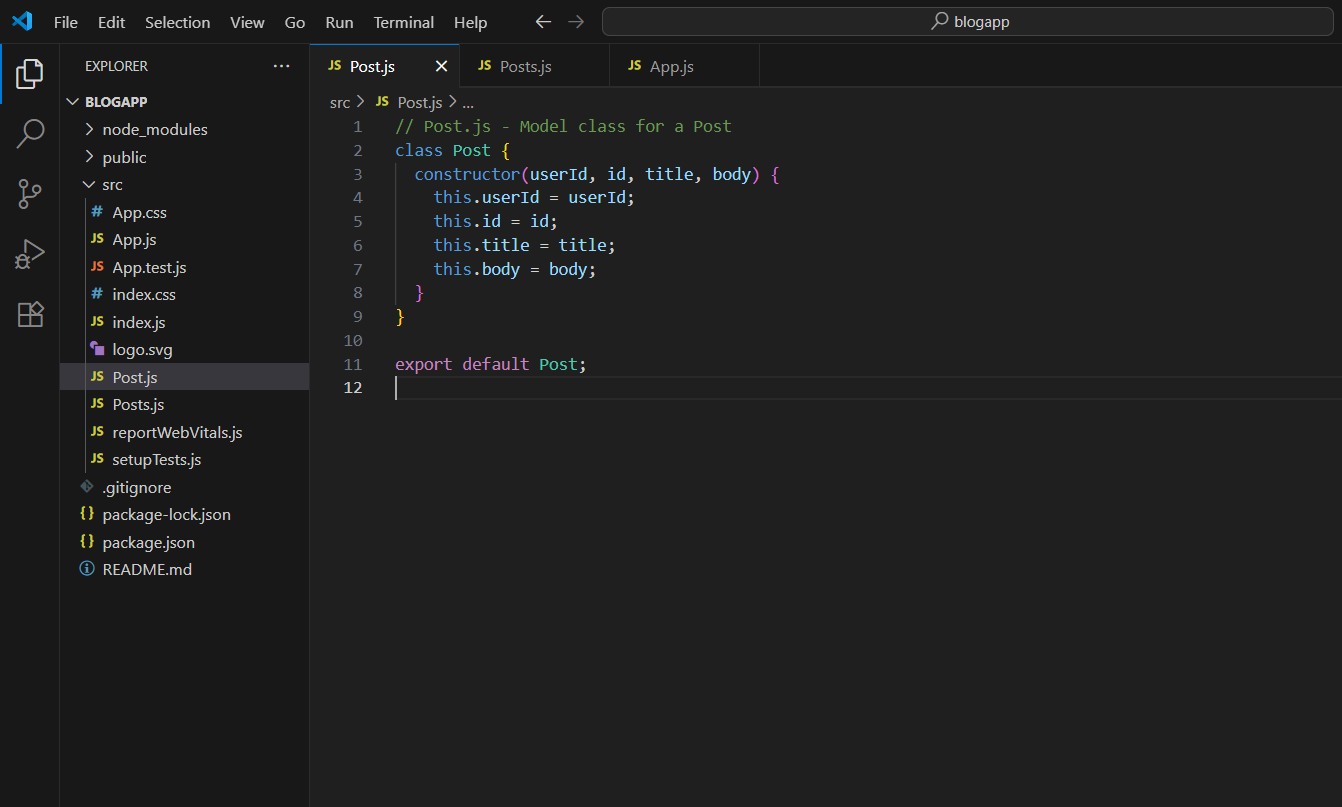


1. **ReactJS-HOL**

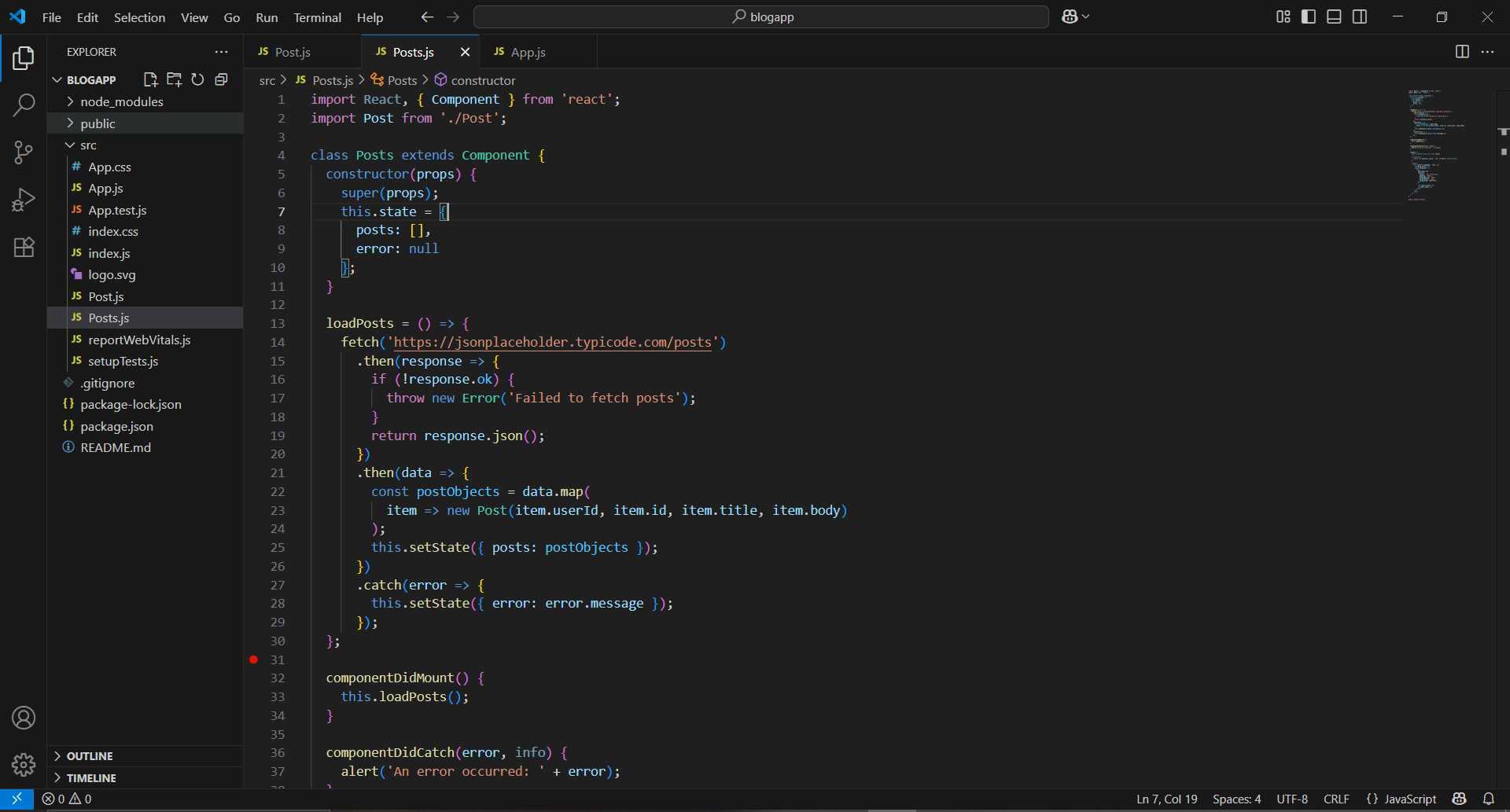
Step 1: Create a new react application using *create-react-app* tool with the name as “blogapp”



Step 2: Create a new file named as Post.js in src folder with following properties

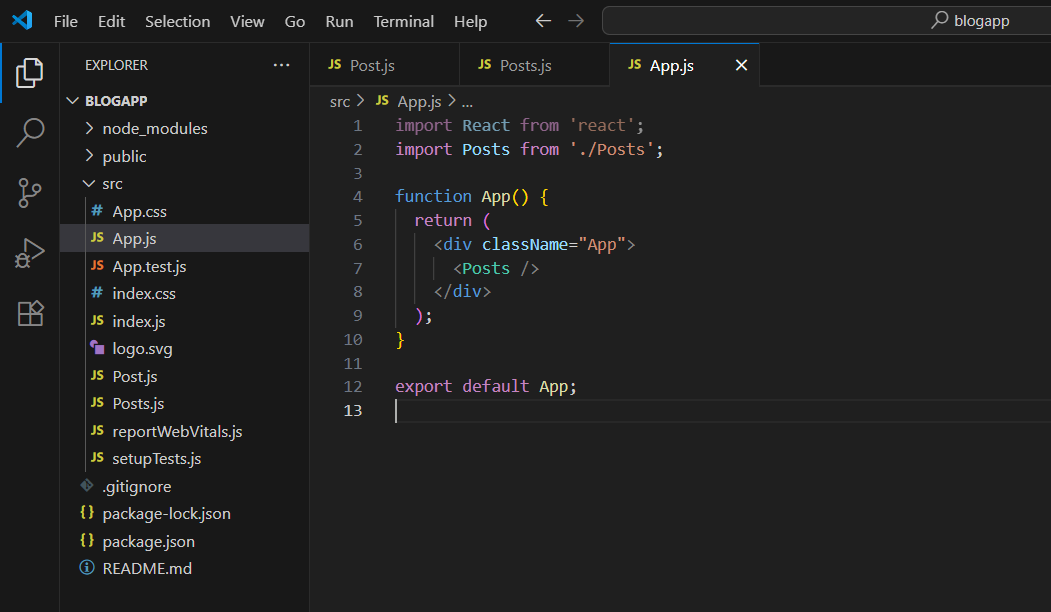


Step 3: Create a new file named as Posts.js in src folder with following properties



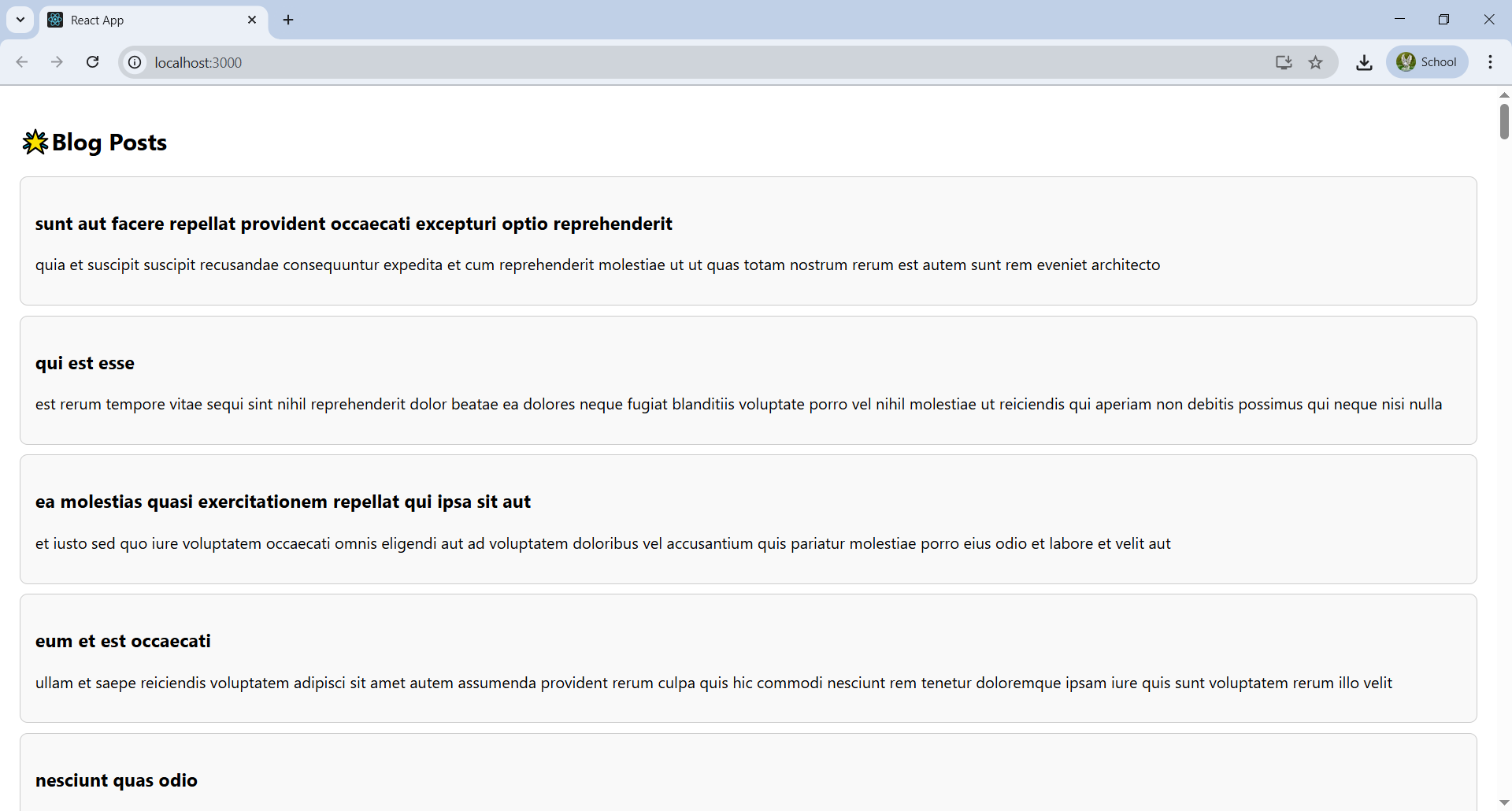
Step 4: Create a new method in component with the name as loadPosts() which will be responsible for using Fetch API and assign it to the component state created earlier. To get the posts use the url (<https://jsonplaceholder.typicode.com/posts>)

Step 5: update the file named as app.js in src folder with following properties



*Step 6:* Build and Run the application using *npm start* command.

**OUTPUT:**

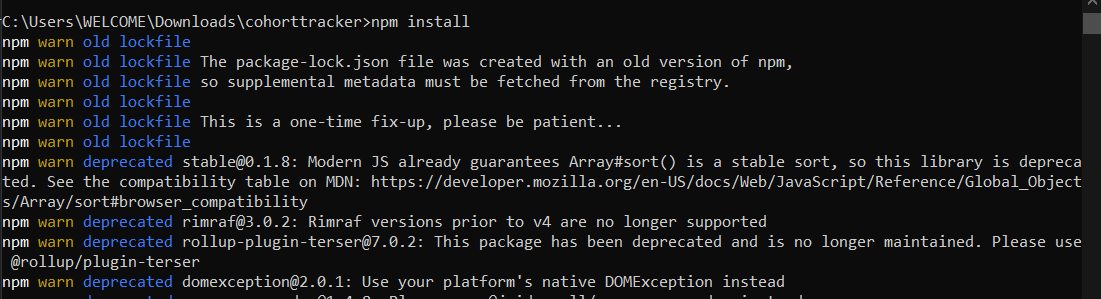


1. **ReactJS-HOL**

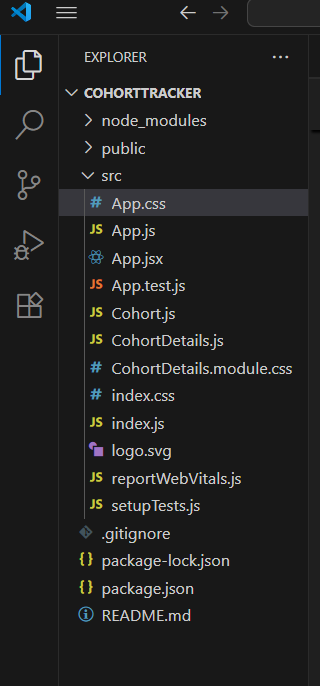
My Academy team at Cognizant want to create a dashboard containing the details of ongoing and completed cohorts. A react application is created which displays the detail of the cohorts using react component. You are assigned the task of styling these react components.

Step 1: Extract the zip of React Application with the name of “cohorttracker”.

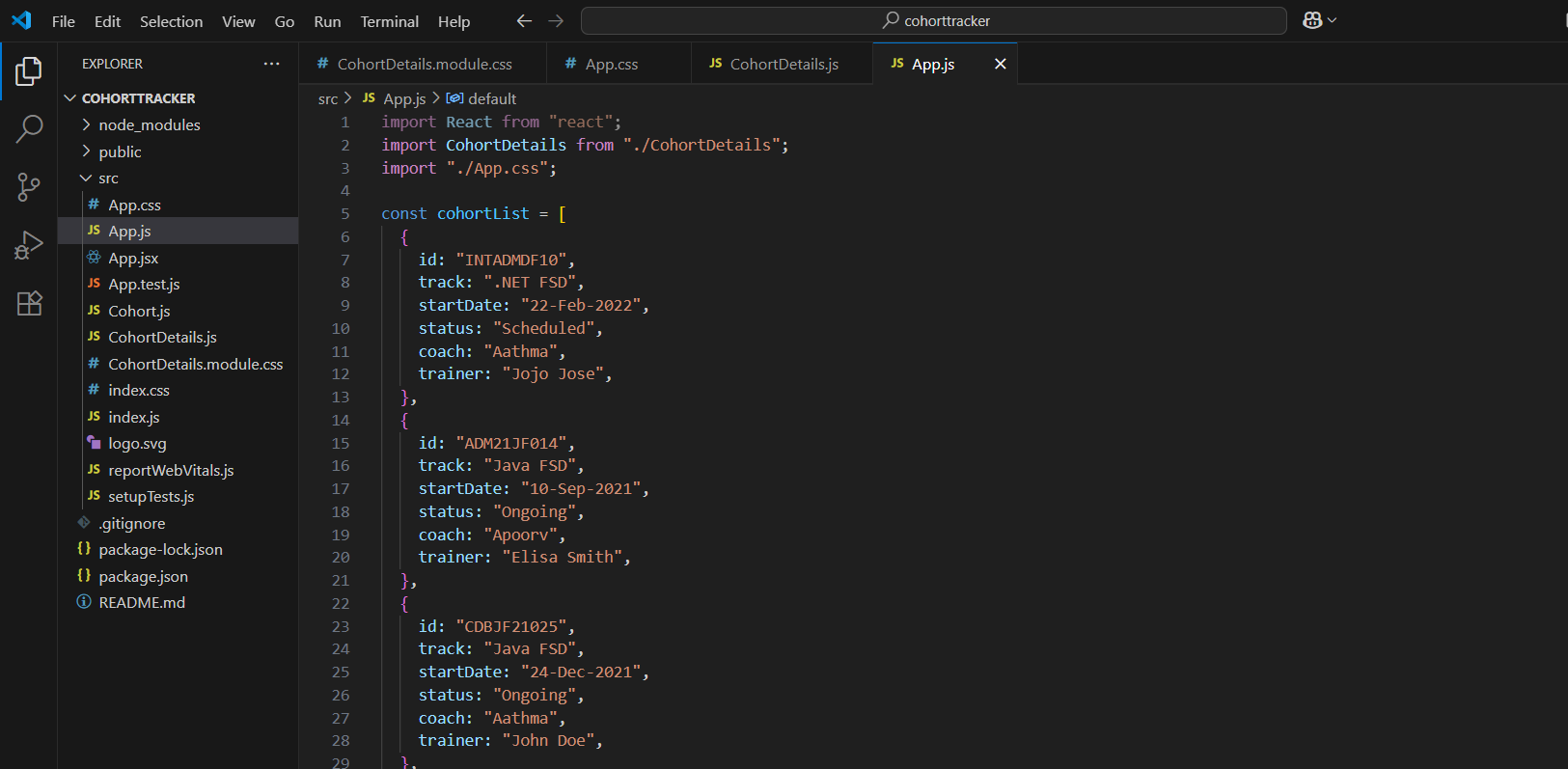
Step 2: Install the required pacakages .



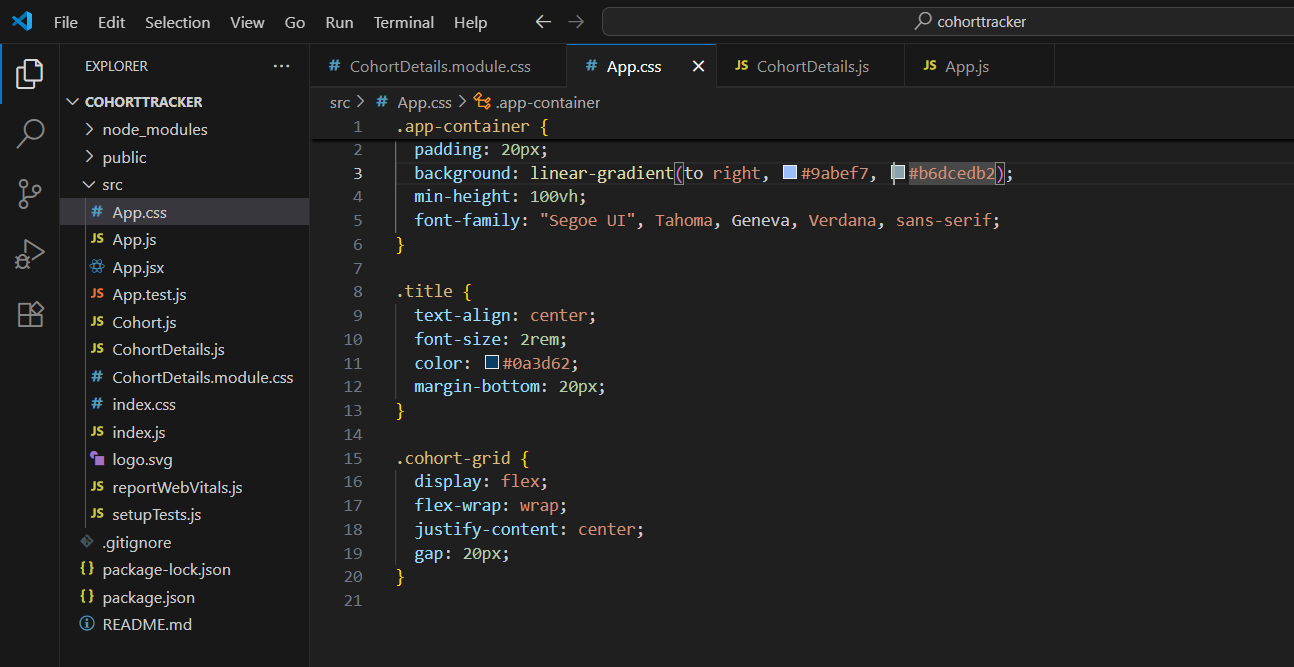
Step 3: Open the folder of cohorttracker in Visual Studio Code



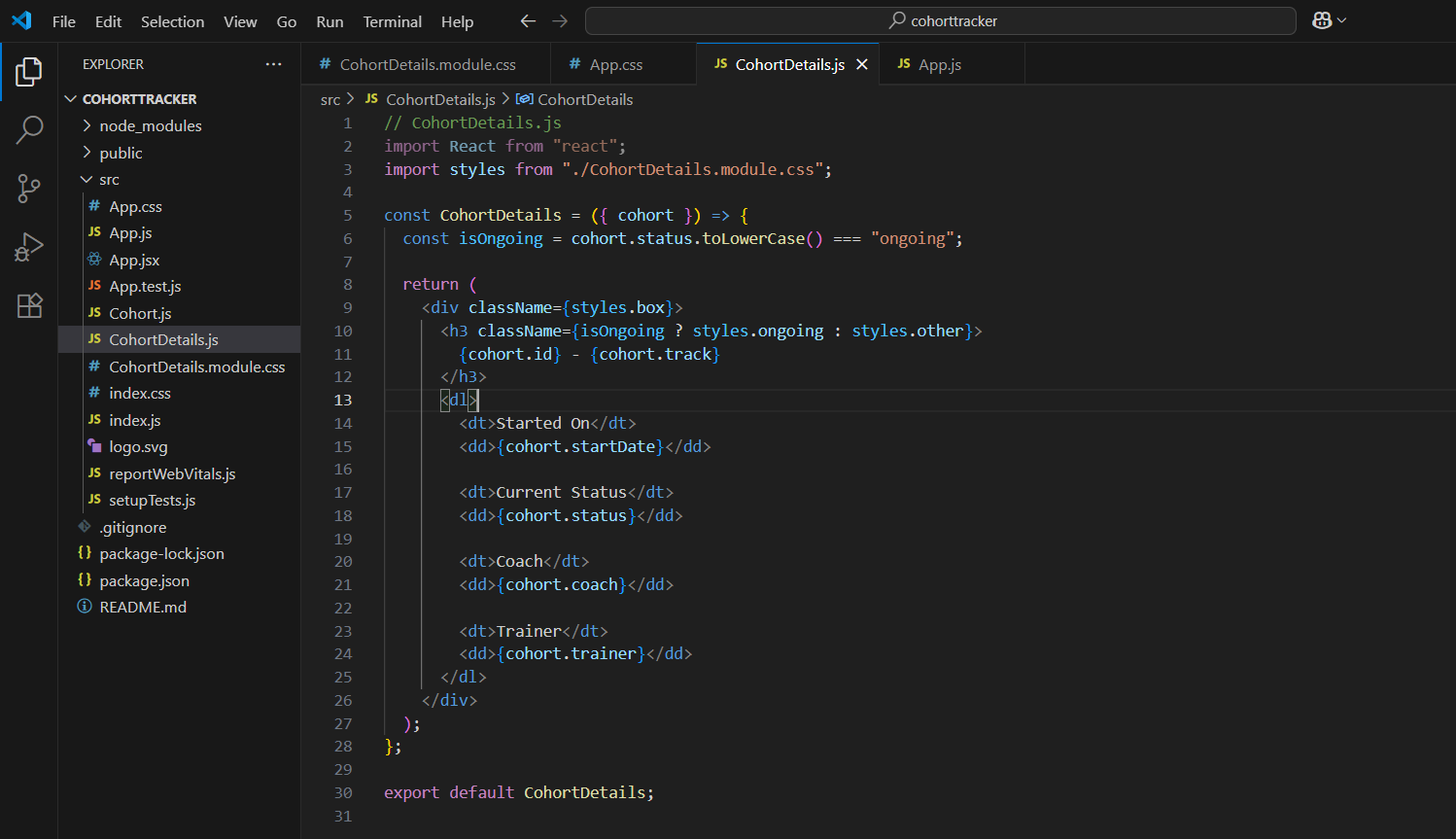
Step 4: Open the App.js file in Src Folder of cohorttracker. Update the Code:



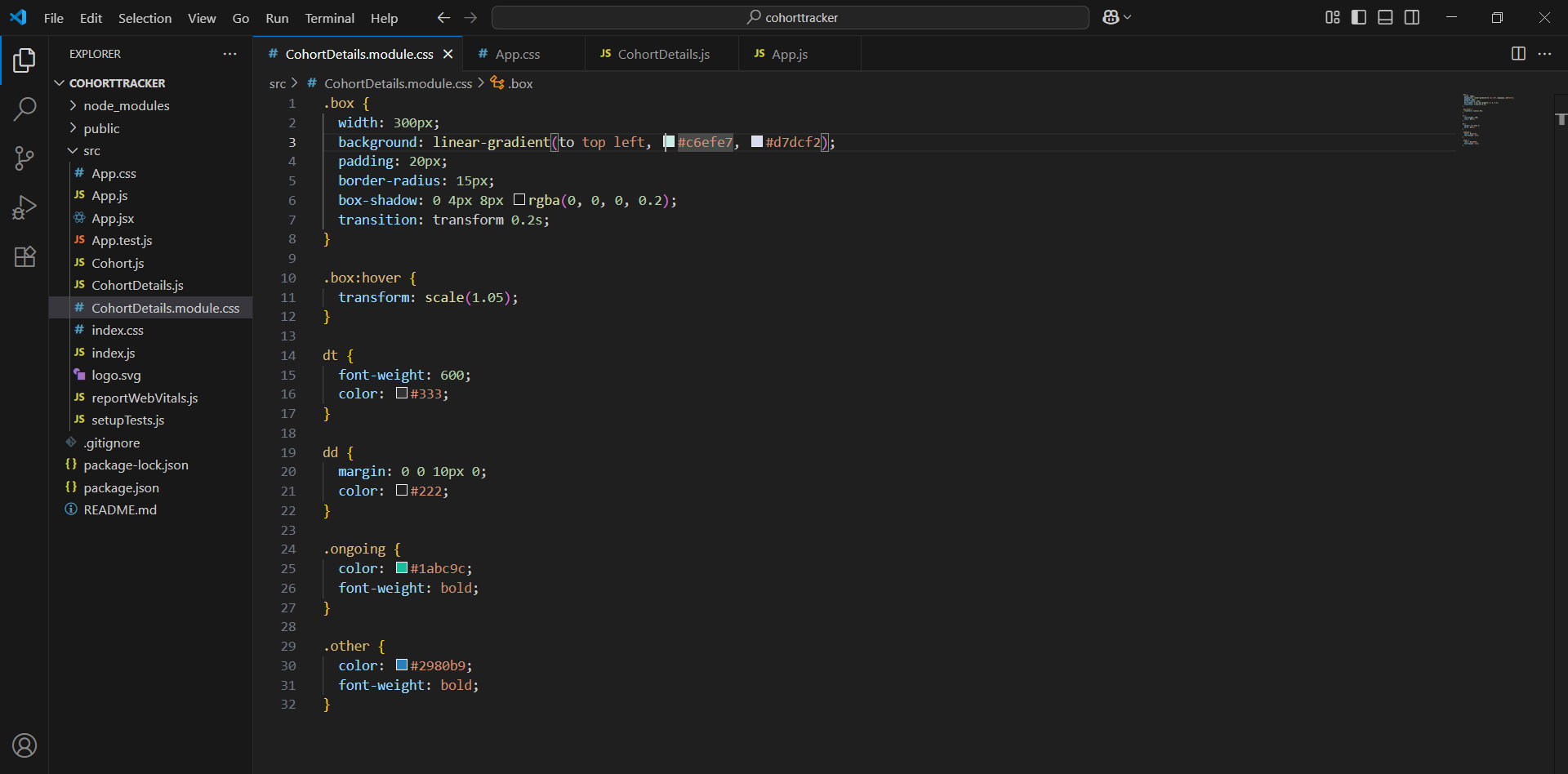
Step 5: Open the App.css file in Src Folder of cohorttracker. Update the Code:



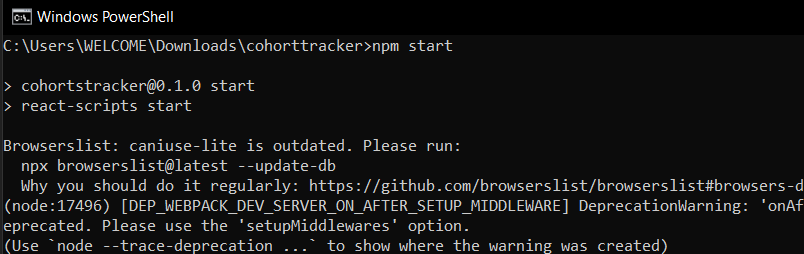
Step 6: Create the filr name CohortDetails file in Src Folder of cohorttracker. Write the Code:



Step 7: Create the filr name /CohortDetails.module.css file in Src Folder of cohorttracker. Write the Code:



Step 8: Run the following command to execute the React application:



Step 9: Open a new browser window and type “localhost:3000” in the address bar

**OUTPUT:**

