Candidate Name:- ROHITHGOWDA V

Superset ID:-6430364

Mail ID:- [rohith2005v@gmail.com](mailto:rohith2005v@gmail.com)

**WEEK – 6 HANDS ON EXERCISE (JAVA FSE DEEPSKILLING)**

**REACT**

**Problem no:1**

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.

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

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

1. Install Create-react-app by running the following command in the command prompt:



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



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



1. Open the folder of myfirstreact in Visual Studio Code
2. Open the App.js file in Src Folder of myfirstreact
3. Remove the current content of “App.js”
4. Replace it with the following:



1. Run the following command to execute the React application:



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

**Solution:**

**App.js**

import React from 'react';

function App() {

  return (

    <div>

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

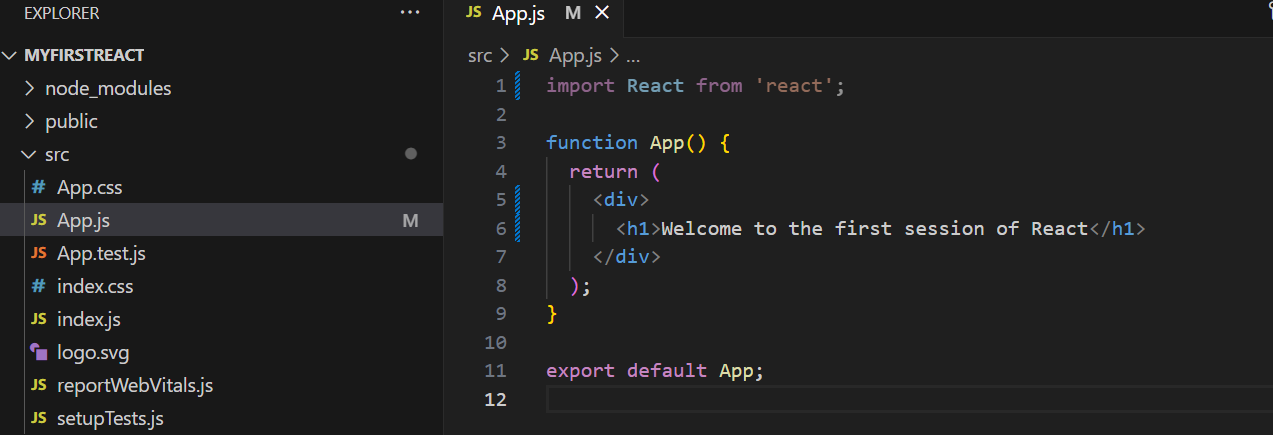
    </div>

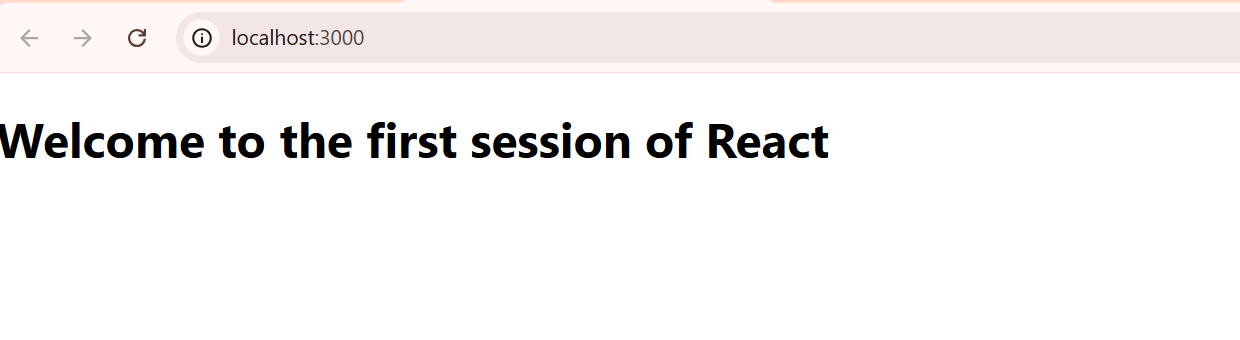
  );

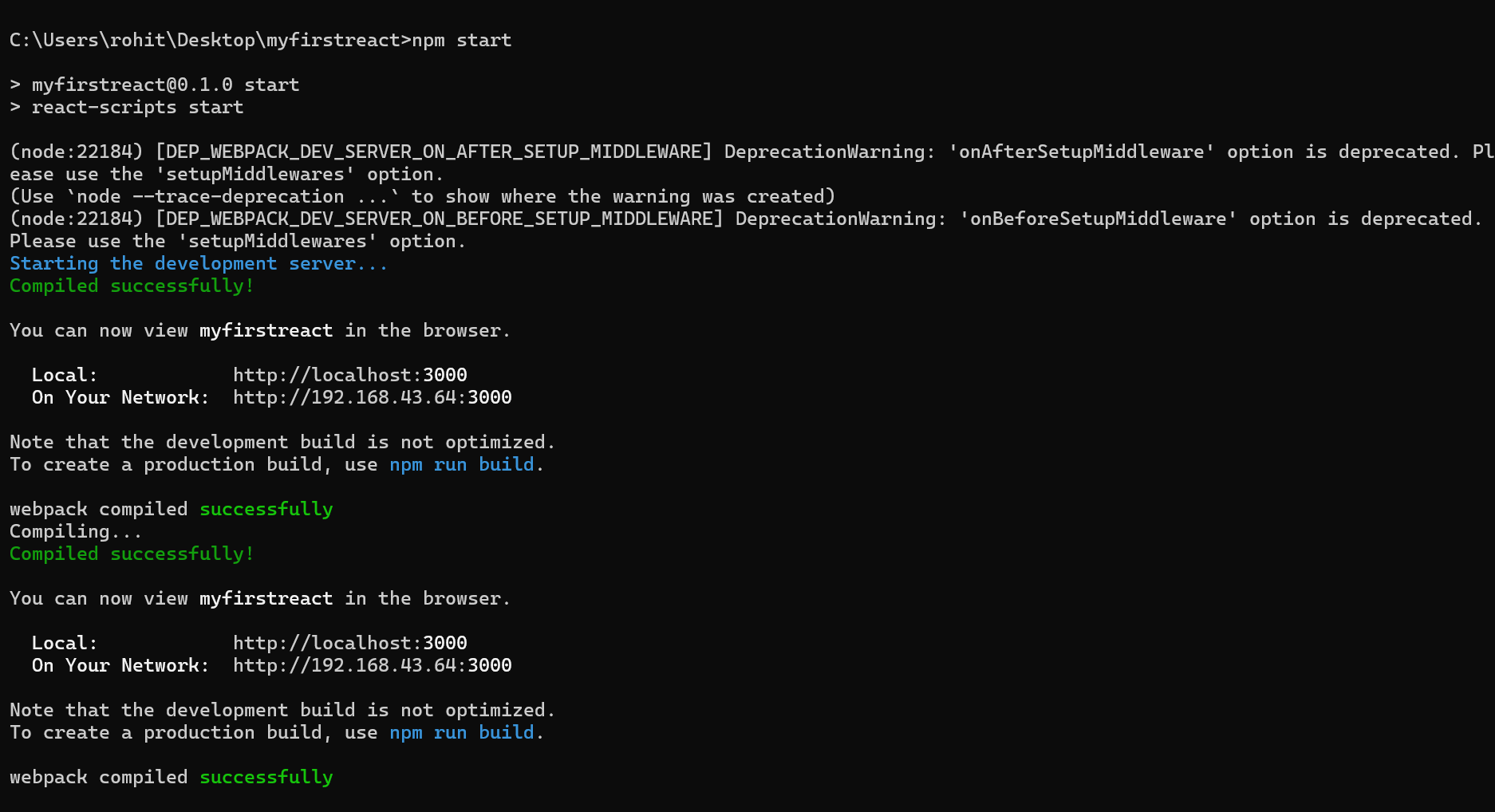
}

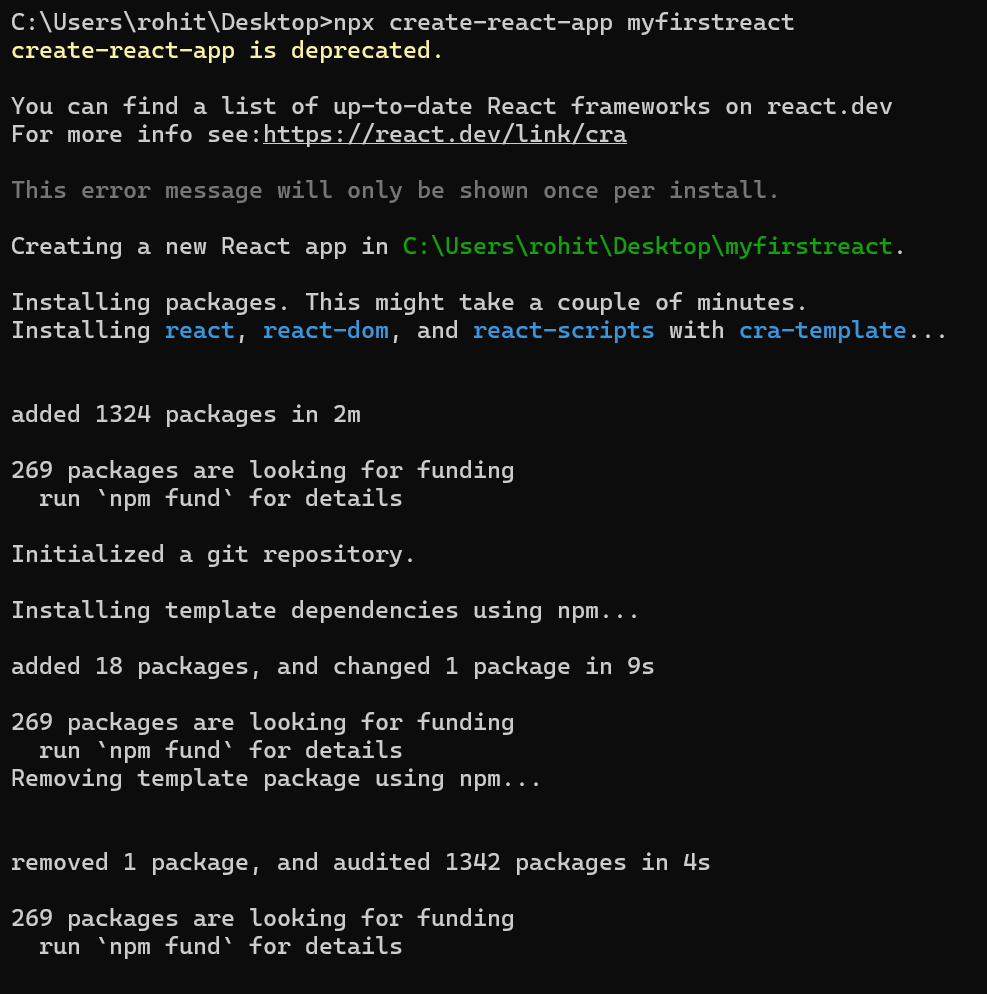
export default App;

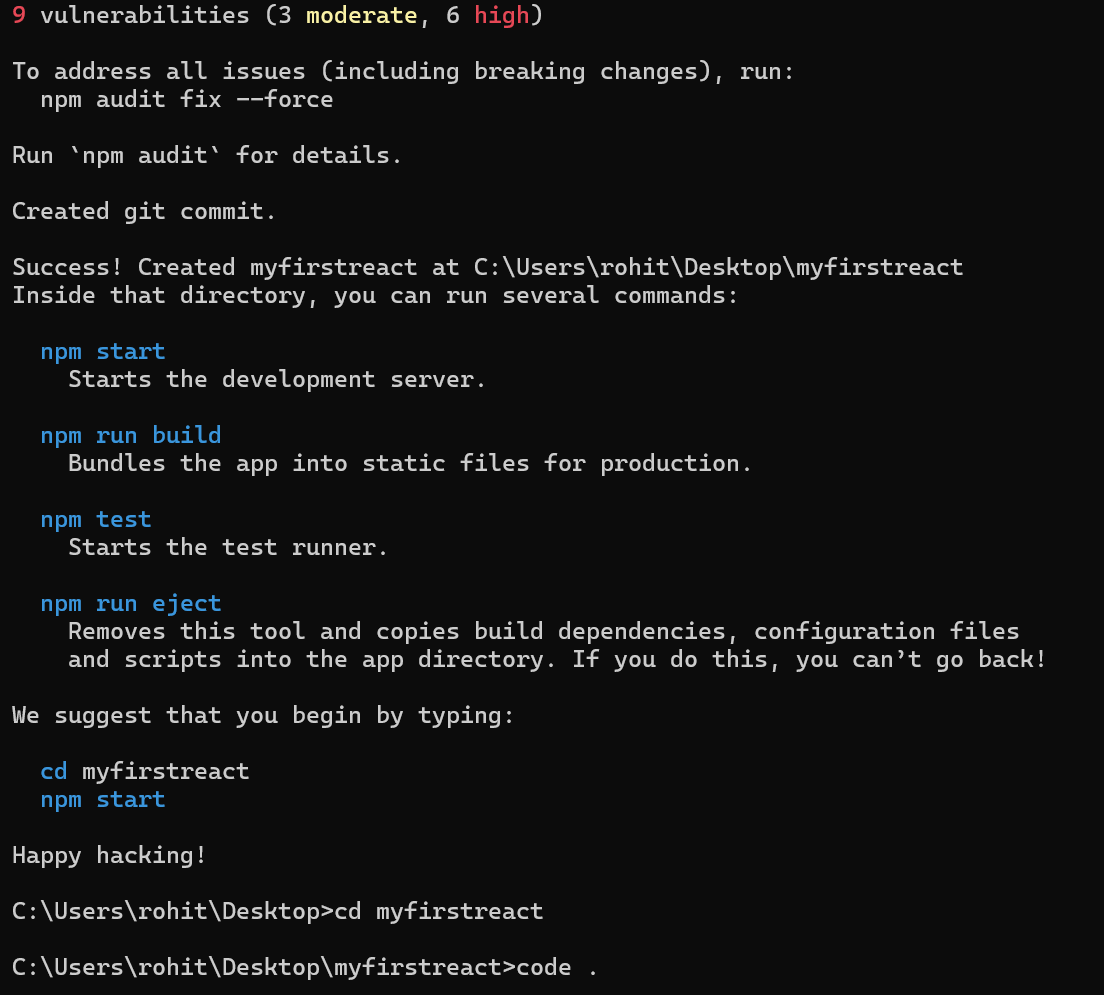
**Output Images:**

****

****







**Problem no:2**

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.

1. Create a React project named “StudentApp” type the following command in terminal of Visual studio:



1. Create a new folder under Src folder with the name “Components”. Add a new file named “Home.js”
2. Type the following code in Home.js



1. Under Src folder add another file named “About.js”
2. Repeat the same steps for Creating “About” and “Contact” component by adding a new file as ”About.js”, “Contact.js” under “Src” folder and edit the code as mentioned for “Home” Component
3. Edit the App.js to invoke the Home, About and Contact component as follows:



1. In command Prompt, navigate into StudentApp and execute the code by typing the following command:



1. Open browser and type “localhost:3000” in the address bar:

**Solution:**

Create a folder names Components in that do this process

**Home.js**

import React from 'react';

function Home() {

return (

<div>

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

</div>

);

}

export default Home;

**About.js**

import React from 'react';

function About() {

  return (

    <div>

      <h2>Welcome to the About page of the Student Management Portal</h2>

    </div>

  );

}

export default About;

**Contact.js**

import React from 'react';

function Contact() {

  return (

    <div>

      <h2>Welcome to the Contact page of the Student Management Portal</h2>

    </div>

  );

}

export default Contact;

**App.js**

import React from 'react';

import Home from './Components/Home';

import About from './Components/About';

import Contact from './Components/Contact';

function App() {

  return (

    <div className="App">

      <Home />

      <About />

      <Contact />

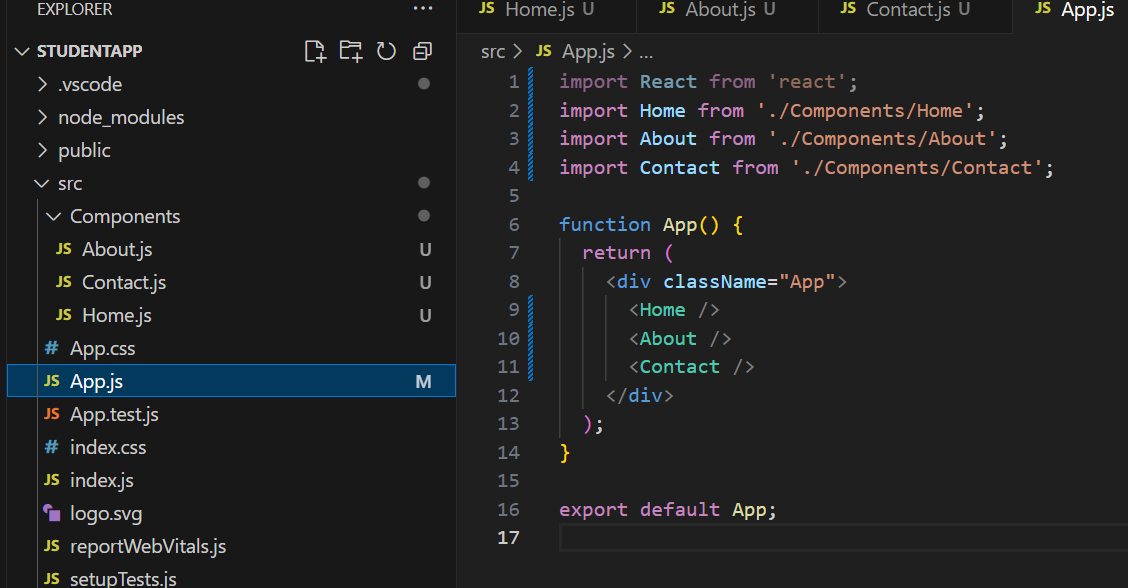
    </div>

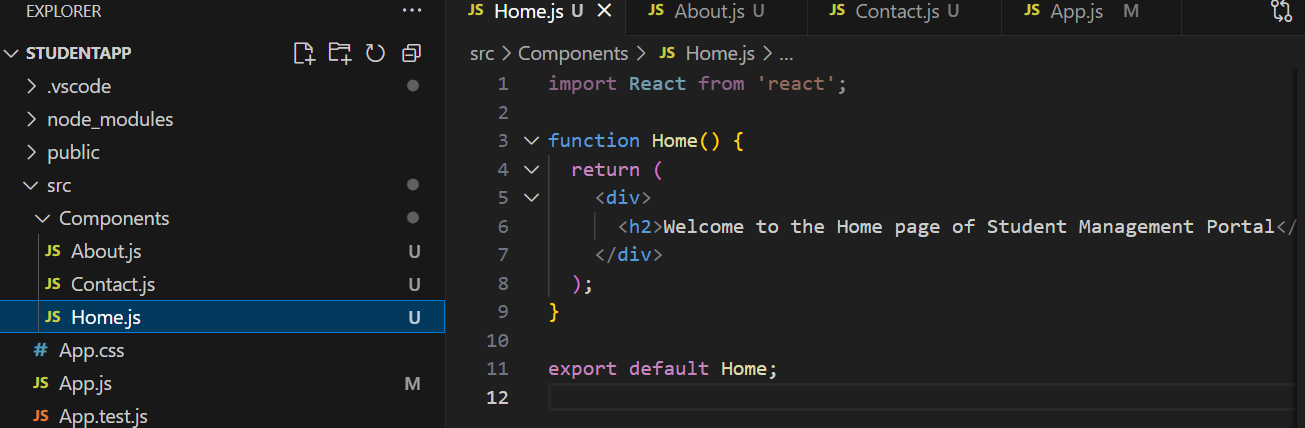
  );

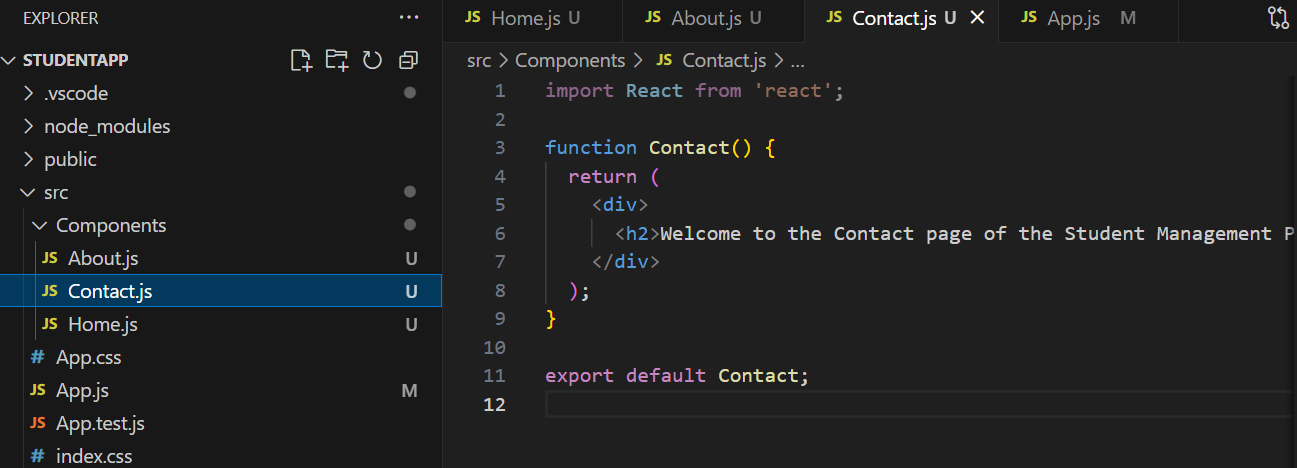
}

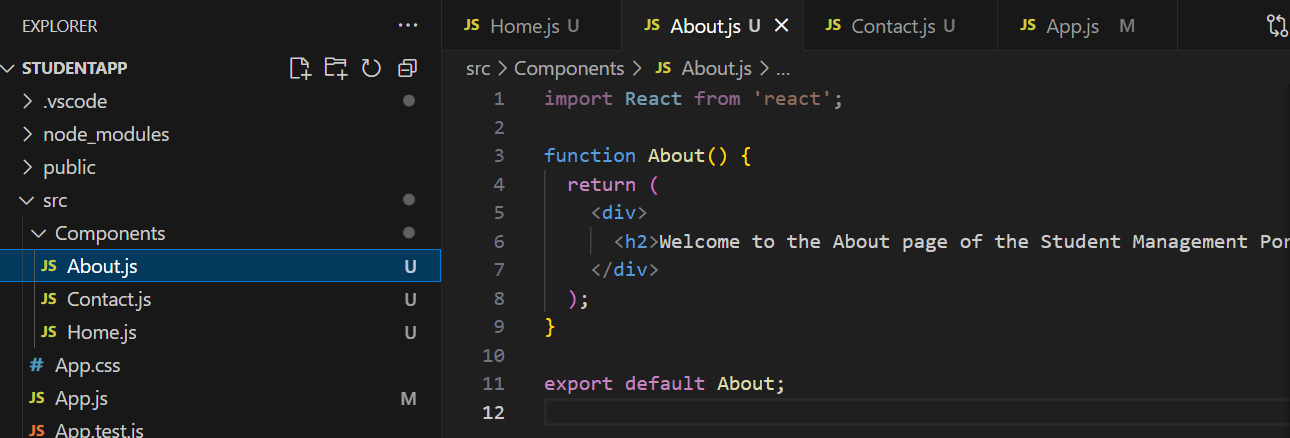
export default App;

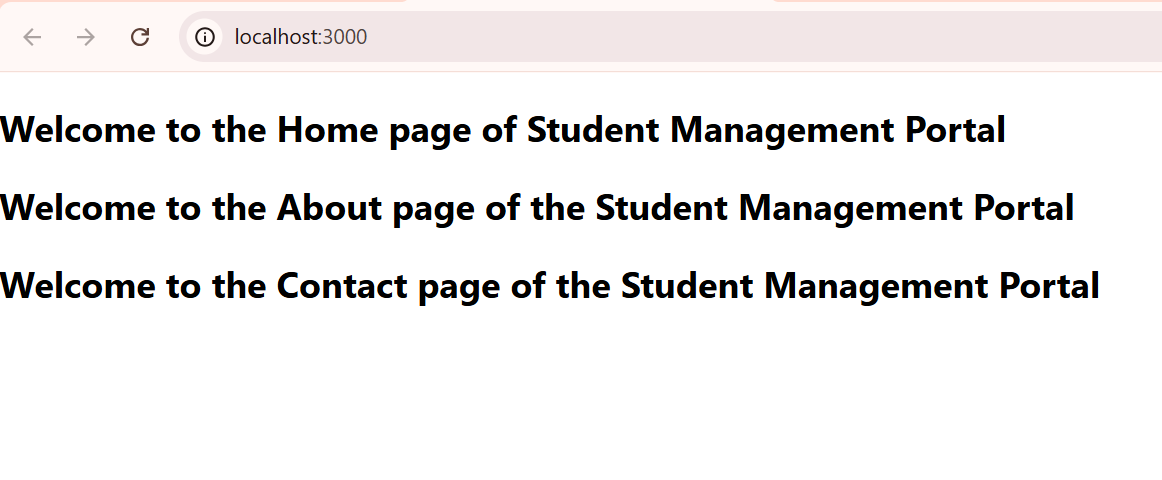
**Output Images:**

****

****

****

****



**Problem no:3**

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.

1. Create a React project named “scorecalculatorapp” type the following command in terminal of Visual studio:



1. Create a new folder under Src folder with the name “Components”. Add a new file named “CalculateScore.js”
2. Type the following code in CalculateScore.js





1. Create a Folder named Stylesheets and add a file named “mystyle.css” in order to add some styles to the components:



1. Edit the App.js to invoke the CalculateScore functional component as follows:



1. In command Prompt, navigate into scorecalculatorapp and execute the code by typing the following command:



1. Open browser and type “localhost:3000” in the address bar:

**Solution:**

Create folder Components and follow the process

**CalculateScore.js**

import React, { useState } from 'react';

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

function CalculateScore() {

  const [name, setName] = useState('');

  const [school, setSchool] = useState('');

  const [total, setTotal] = useState('');

  const [goal, setGoal] = useState('');

  const [average, setAverage] = useState(null);

  const calculate = () => {

    if (goal && total) {

      const avg = parseFloat(total) / parseFloat(goal);

      setAverage(avg.toFixed(2));

    }

  };

  return (

    <div className="score-container">

      <h2>Student Score Calculator</h2>

      <input type="text" placeholder="Enter Name" value={name}

        onChange={(e) => setName(e.target.value)} />

      <input type="text" placeholder="Enter School" value={school}

        onChange={(e) => setSchool(e.target.value)} />

      <input type="number" placeholder="Enter Total Score" value={total}

        onChange={(e) => setTotal(e.target.value)} />

      <input type="number" placeholder="Enter Goal (Subjects)" value={goal}

        onChange={(e) => setGoal(e.target.value)} />

      <button onClick={calculate}>Calculate Average</button>

      {average && (

        <p>Hello {name} from {school}, your average score is <strong>{average}</strong>.</p>

      )}

    </div>

  );

}

export default CalculateScore;

**mystyle.css ( Do this in StyleSheets folder)**

.score-container {

  max-width: 400px;

  margin: auto;

  padding: 20px;

  border: 2px solid #0c3c6e;

  border-radius: 10px;

  background-color: #f0f8ff;

  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

.score-container h2 {

  text-align: center;

  color: #0c3c6e;

}

.score-container input {

  width: 100%;

  padding: 10px;

  margin: 8px 0;

  border: 1px solid #ccc;

  border-radius: 5px;

}

.score-container button {

  width: 100%;

  padding: 10px;

  background-color: #0c3c6e;

  color: white;

  border: none;

  border-radius: 5px;

  cursor: pointer;

}

.score-container button:hover {

  background-color: #0a2e4d;

}

.score-container p {

  margin-top: 15px;

  font-weight: bold;

  color: #333;

  text-align: center;

}

**App.js**

import React from 'react';

import './App.css';

import CalculateScore from './Components/CalculateScore';

function App() {

  return (

    <div className="App">

      <CalculateScore />

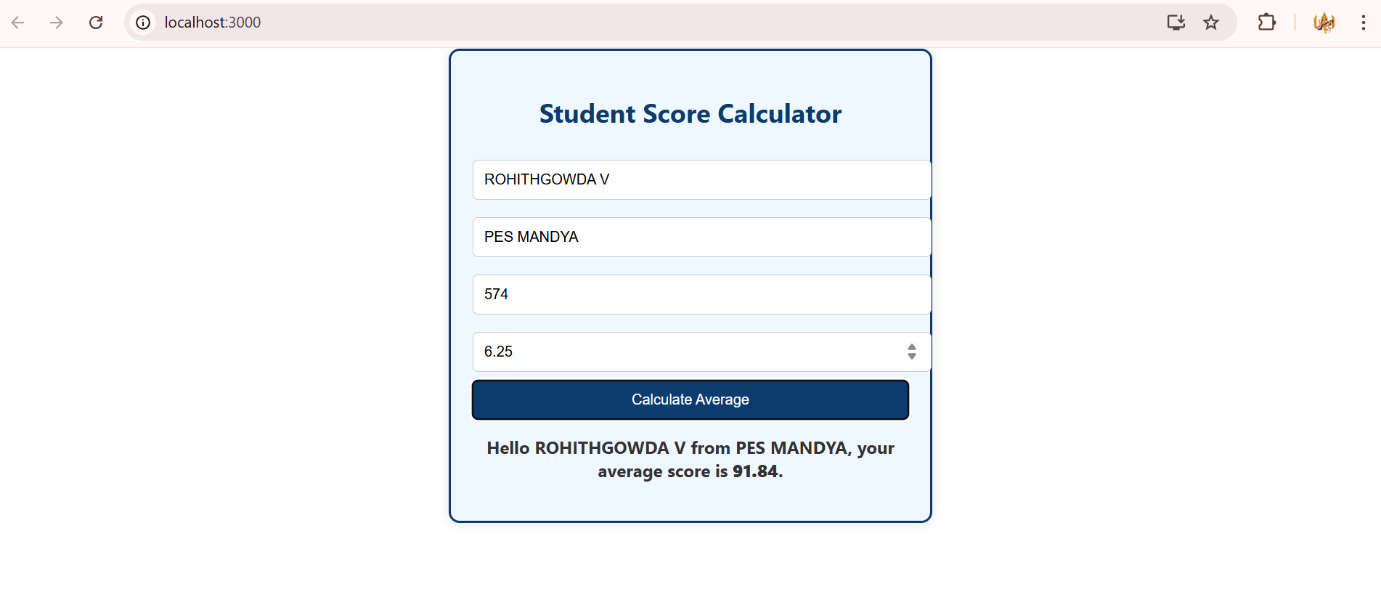
    </div>

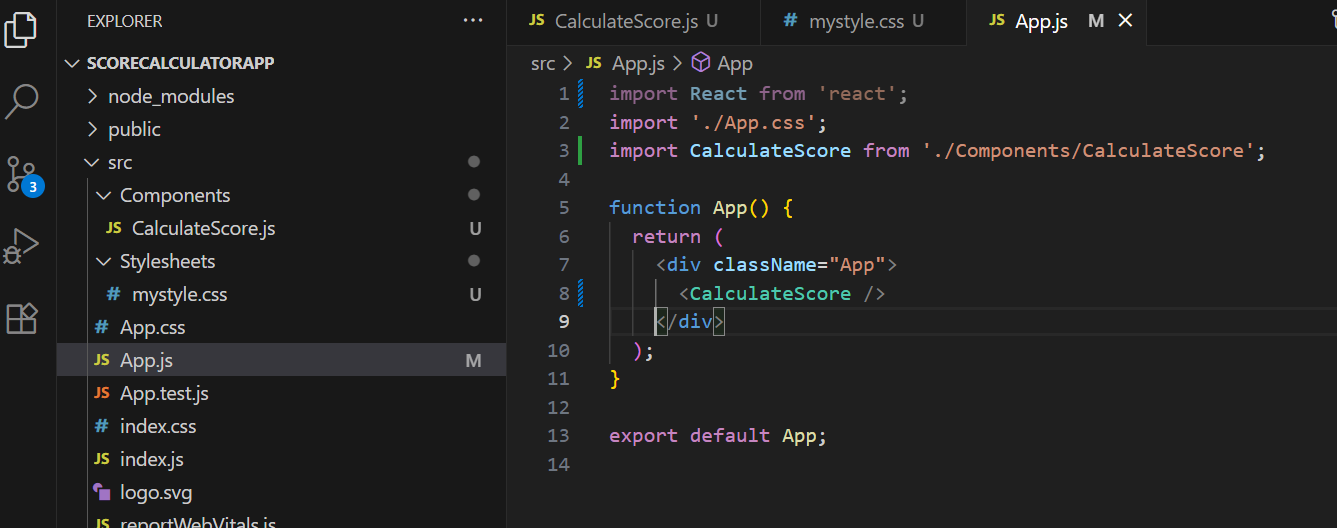
  );

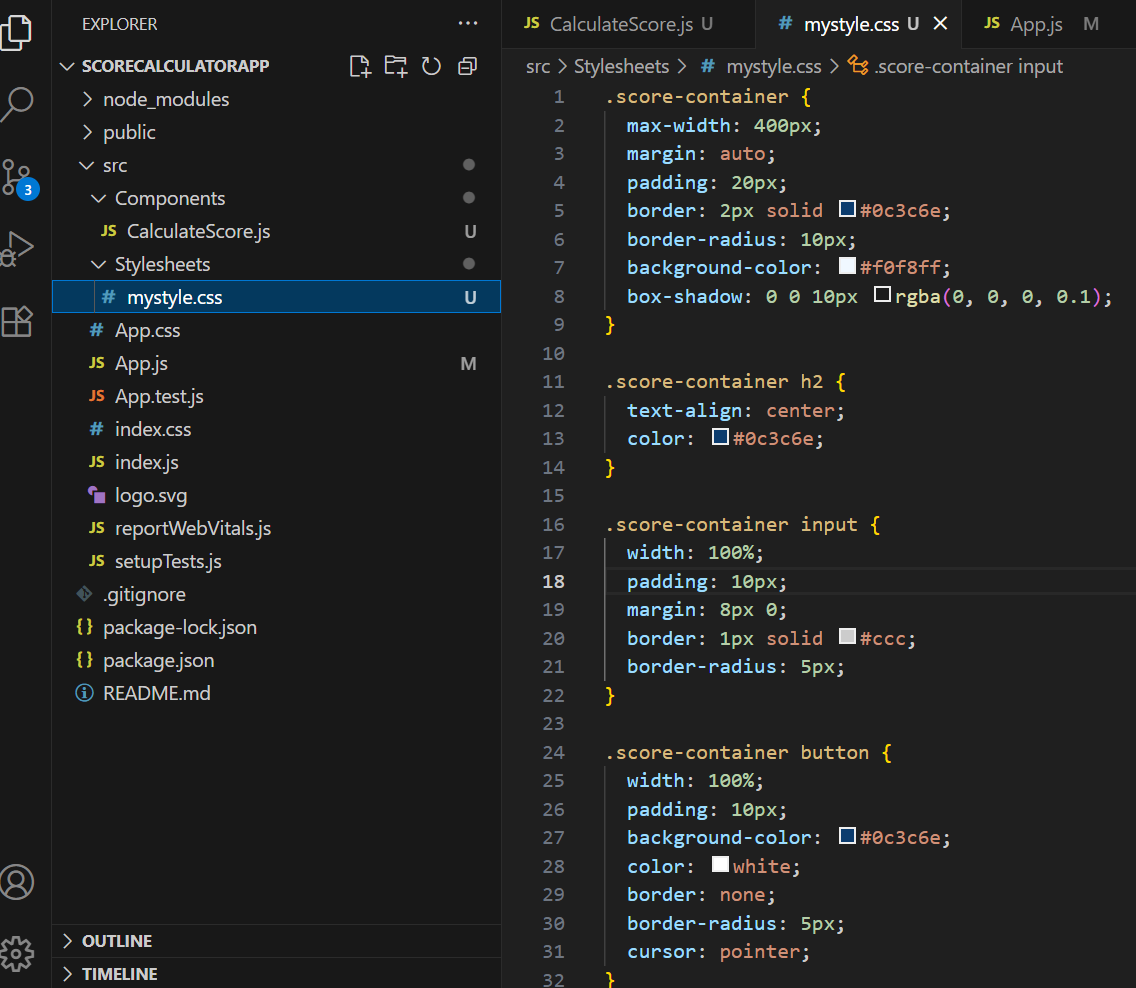
}

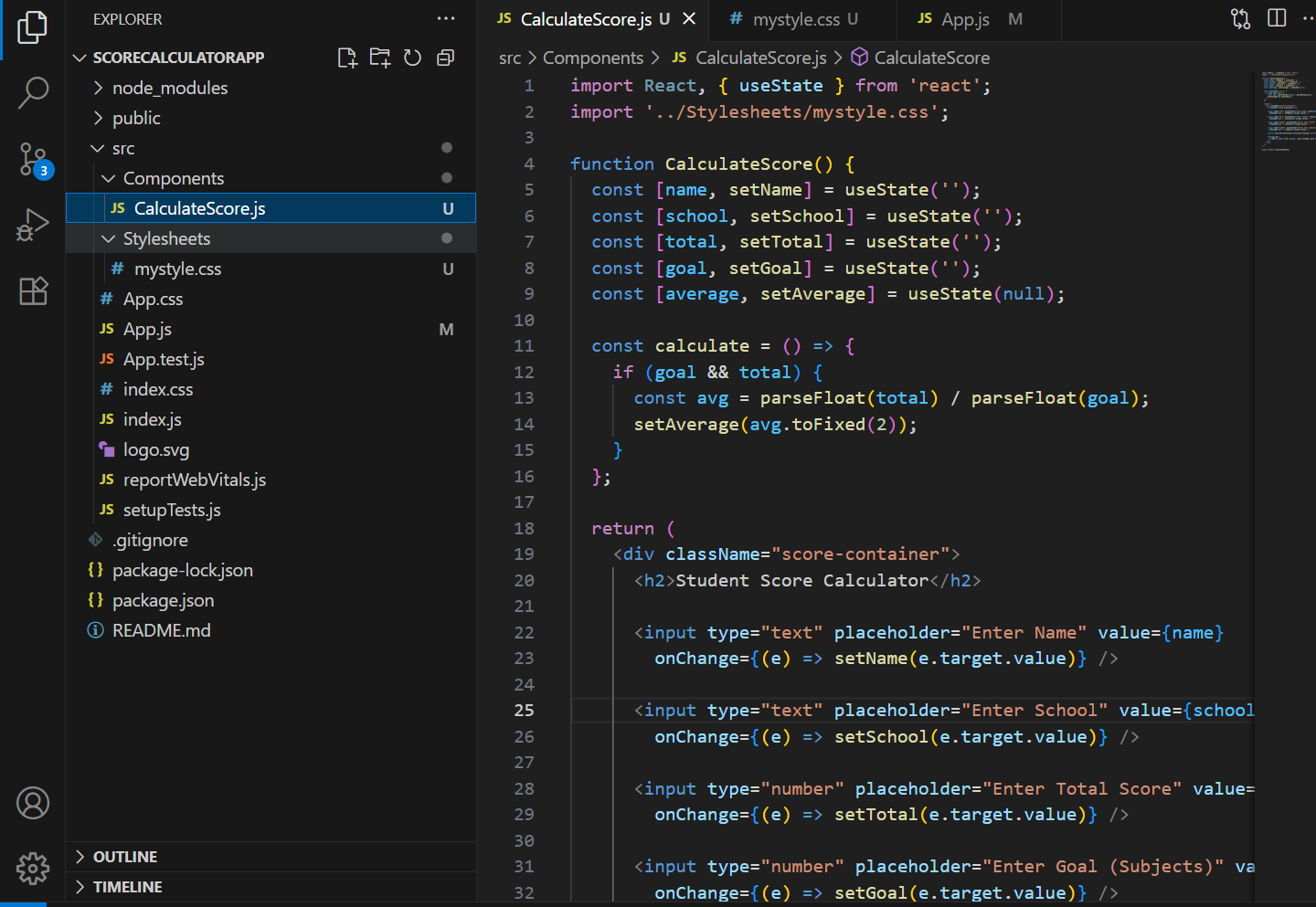
export default App;

**Output Images:**









**Problem no:4**

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.

**Solution:**

**Post.js**

class Post {

  constructor(userId, id, title, body) {

    this.userId = userId;

    this.id = id;

    this.title = title;

    this.body = body;

  }

}

export default Post;

**Posts.js**

import React, { Component } from 'react';

import Post from './Post';

class Posts extends Component {

  constructor(props) {

    super(props);

    this.state = {

      posts: [],

    };

  }

  loadPosts() {

    fetch('https://jsonplaceholder.typicode.com/posts')

      .then((response) => response.json())

      .then((data) => {

        const postsList = data.map(

          (p) => new Post(p.userId, p.id, p.title, p.body)

        );

        this.setState({ posts: postsList });

      })

      .catch((error) => {

        console.error("Error fetching posts:", error);

        alert("Error fetching posts");

      });

  }

  componentDidMount() {

    this.loadPosts();

  }

  componentDidCatch(error, info) {

    alert("An error occurred: " + error);

  }

  render() {

    return (

      <div>

        <h2>Blog Posts</h2>

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

          <div key={post.id} style={{ marginBottom: '20px' }}>

            <h3>{post.title}</h3>

            <p>{post.body}</p>

            <hr />

          </div>

        ))}

      </div>

    );

  }

}

export default Posts;

**App.js**

import React from 'react';

import './App.css';

import Posts from './Posts';

function App() {

  return (

    <div className="App">

      <h1>Welcome to BlogApp</h1>

      <Posts />

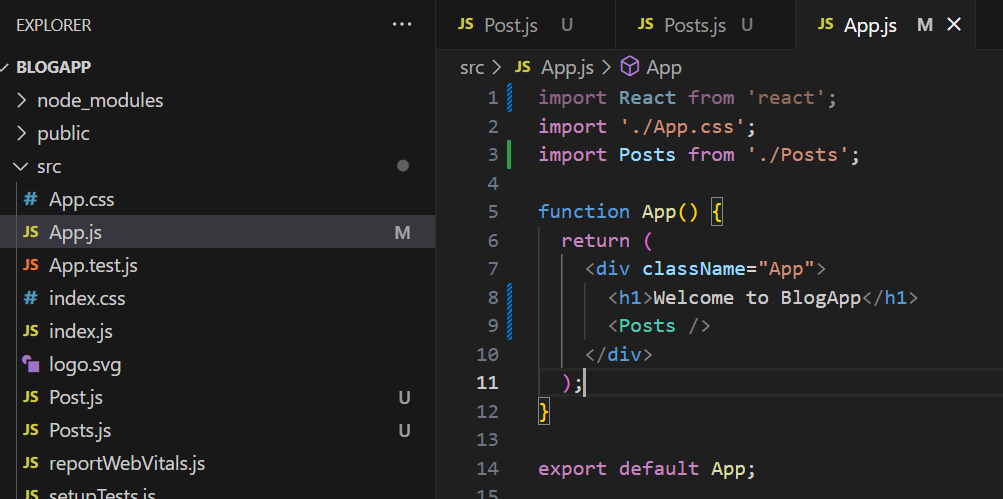
    </div>

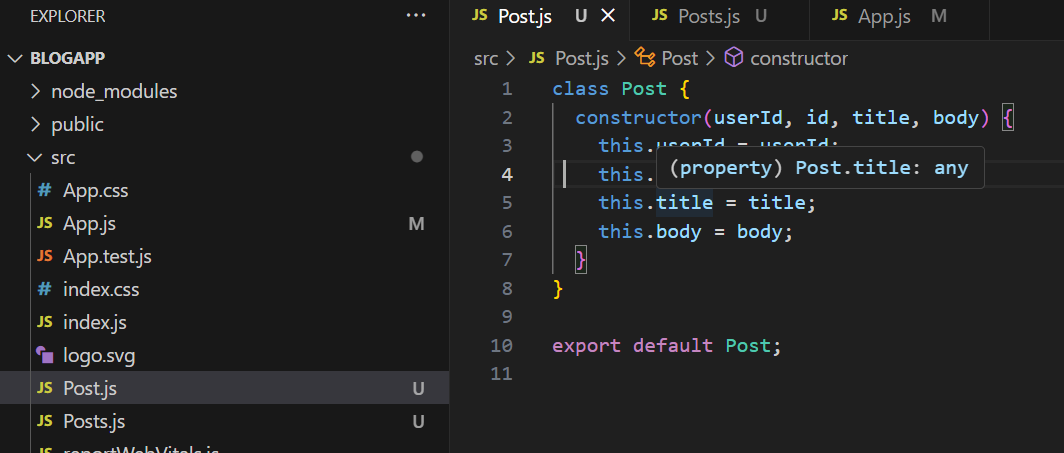
  );

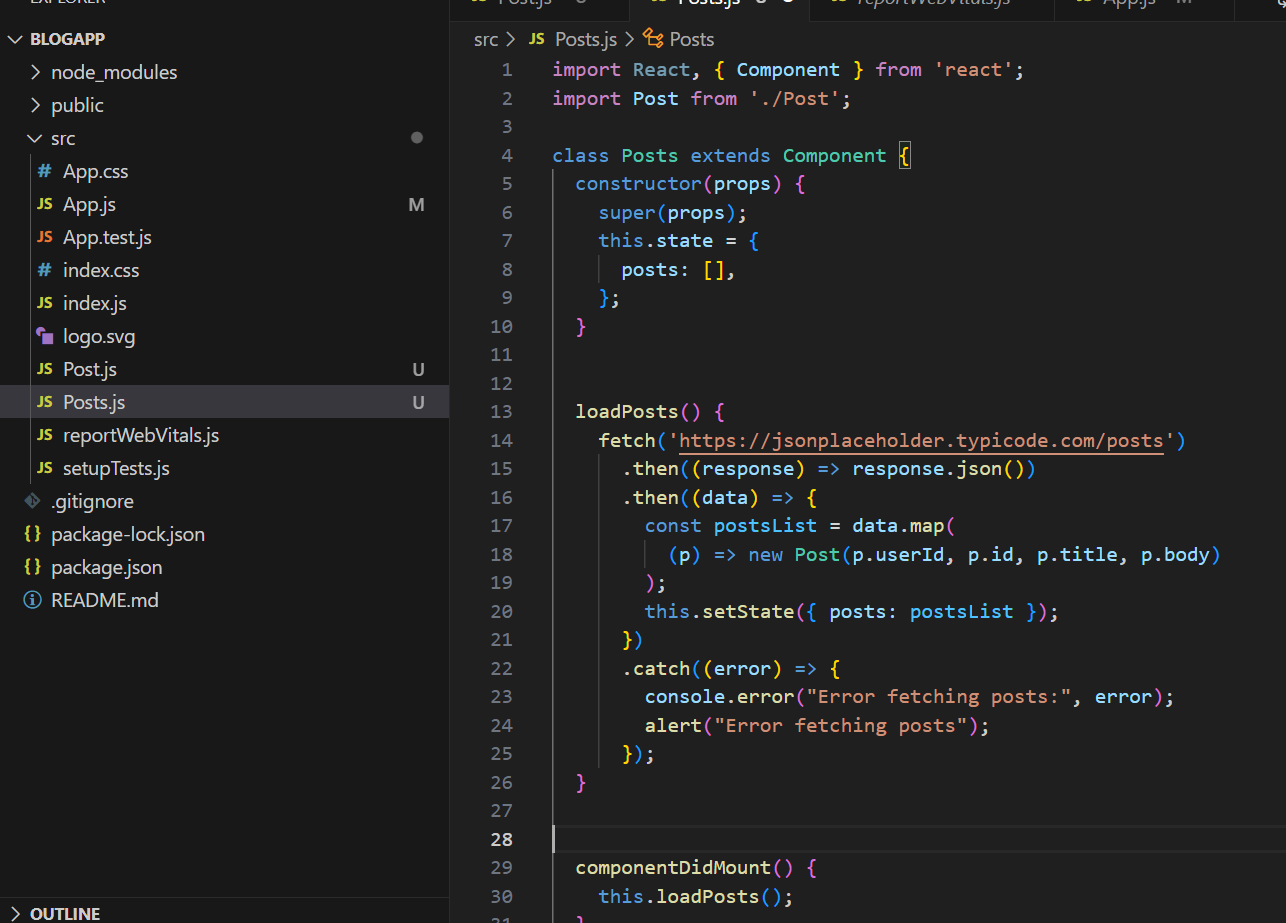
}

export default App;

**Output Images:**









**Problem no:5**

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



*Figure 2: Final Result*

**Solution:**

**CohorotDetails.module.css**

.box {

  width: 300px;

  display: inline-block;

  margin: 10px;

  padding: 10px 20px;

  border: 1px solid black;

  border-radius: 10px;

}

dt {

  font-weight: 500;

}

**CohorotDetails.js**

import React from 'react';

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

function CohortDetails(props) {

    const { cohort } = props;

    return (

        <div className={styles.box}>

            <h3 style={{ color: cohort.currentStatus === "Ongoing" ? "green" : "blue" }}>

                {cohort.cohortCode} - <span>{cohort.technology}</span>

            </h3>

            <dl>

                <dt>Started On</dt>

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

                <dt>Current Status</dt>

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

                <dt>Coach</dt>

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

                <dt>Trainer</dt>

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

            </dl>

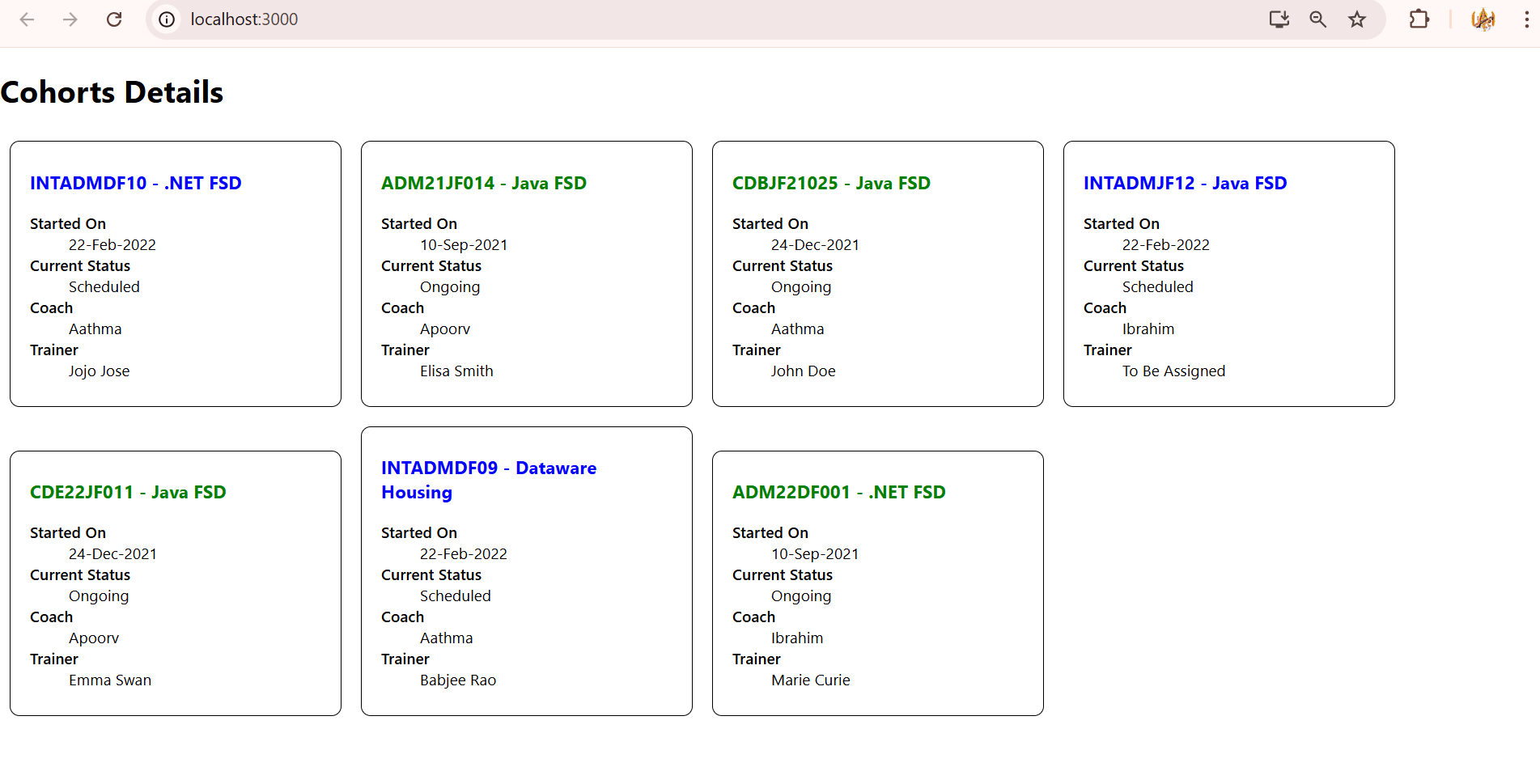
        </div>

    );

}

export default CohortDetails;

**Output Images:**

****

