WEEK 6 – ASSIGNMENT

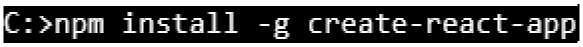
Superset ID: 6390124

React Exercises:-

Exercise 1:

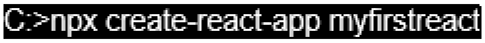
Create a new React Application with the name “myfirstreact”, Run the application to print “welcome to the first session of React” as heading of that page.

1. To create a new React app, Install Nodejs and Npm from the following link: https://nodejs.org/en/download/
2. Install Create-react-app by running the following command in the command prompt:



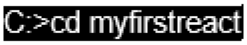
```
C:>npm install -g create-react-app
```

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



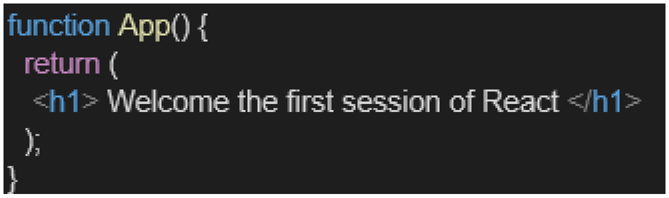
```
C:>npx create-react-app myfirstreact
```

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



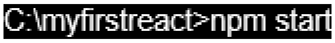
```
C:>cd myfirstreact
```

5. Open the folder of myfirstreact in Visual Studio Code
6. Open the App.js file in Src Folder of myfirstreact
7. Remove the current content of “App.js”
8. Replace it with the following:



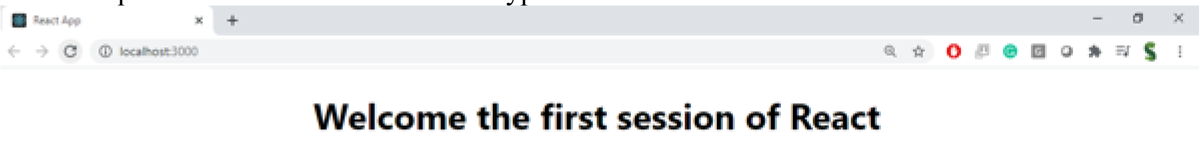
```
function App() {
  return (
   <h1> Welcome the first session of React </h1>
  );
}
```

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



```
C:\myfirstreact>npm start
```

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

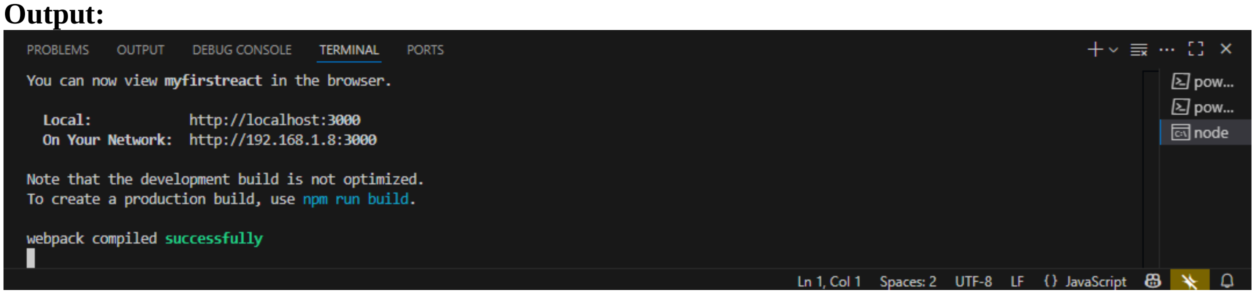


Welcome the first session of React

src/ App.js

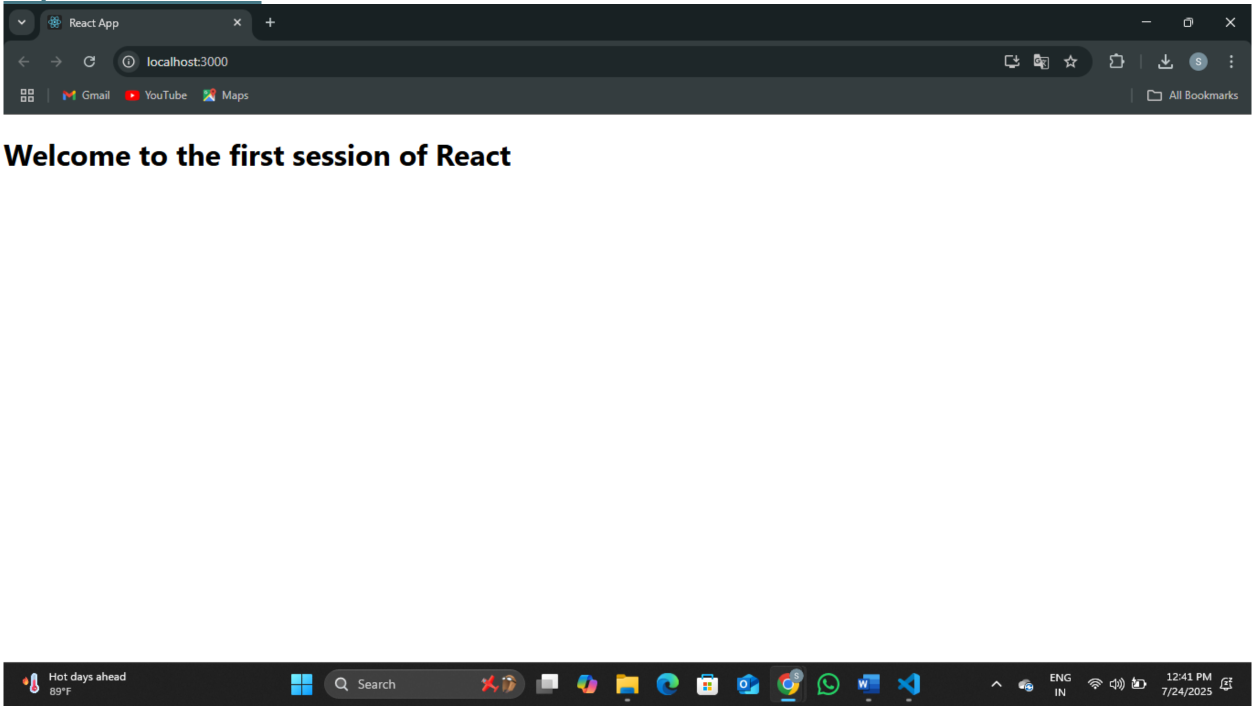
```
import React from 'react';

function App() {
  return (
    <h1>Welcome to the first session of React</h1>
  );
}

export default App;
```

Output:



```
You can now view myfirstreact in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.1.8:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

http://localhost:3000



Welcome to the first session of React

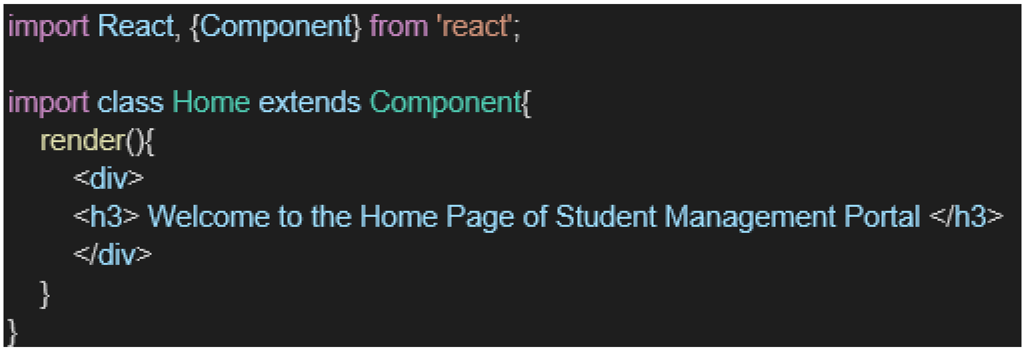
Exercise 2:

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:

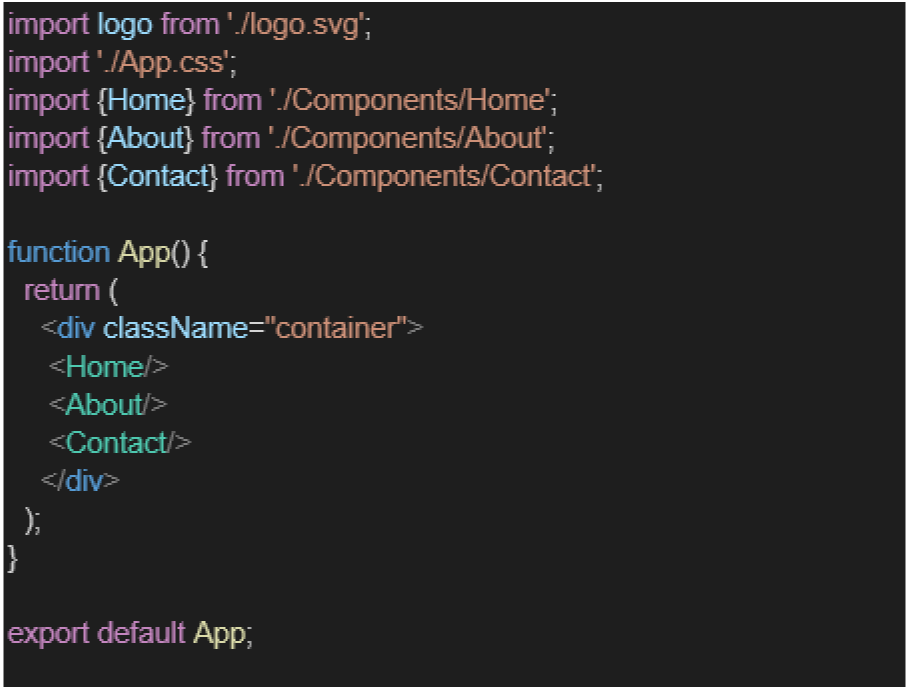
```
C:>npx create-react-app StudentApp
```

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



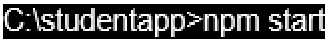
```
import React, {Component} from 'react';

import class Home extends Component{
    render(){
        <div>
        <h3> Welcome to the Home Page of Student Management Portal </h3>
        </div>
    }
}
```

4. Under Src folder add another file named “About.js”
5. 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.
6. Edit the App.js to invoke the Home, About and Contact component as follows:



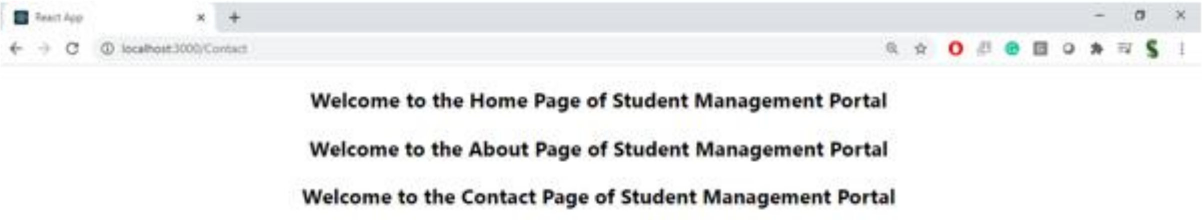
```
import logo from './logo.svg';
import './App.css';
import {Home} from './Components/Home';
import {About} from './Components/About';
import {Contact} from './Components/Contact';

function App() {
  return (
    <div className="container">
     <Home/>
     <About/>
     <Contact/>
    </div>
  );
}

export default App;
```

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



```
C:\studentapp>npm start
```

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



Welcome to the Home Page of Student Management Portal

Welcome to the About Page of Student Management Portal

Welcome to the Contact Page of Student Management Portal

src/ Components / 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;
```

src/ Components / 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;
```

src/ Components / Contact.js

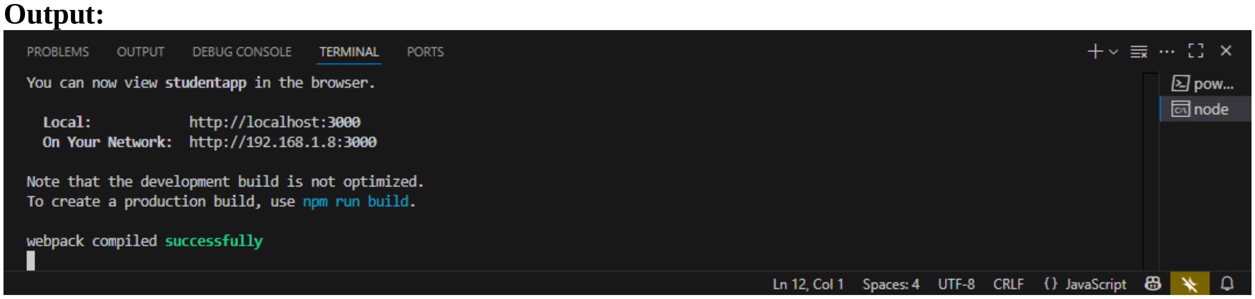
```
import React from 'react';

function Contact() {
  return (
    <div>
      <h2>Welcome to the Contact page of the Student Management Portal</h2>
    </div>
  );
}

export default Contact;
```

src/App.js

```
import React from 'react';
import './App.css';
import Home from './Components/Home';
import About from './Components/About';
import Contact from './Components/Contact';
```

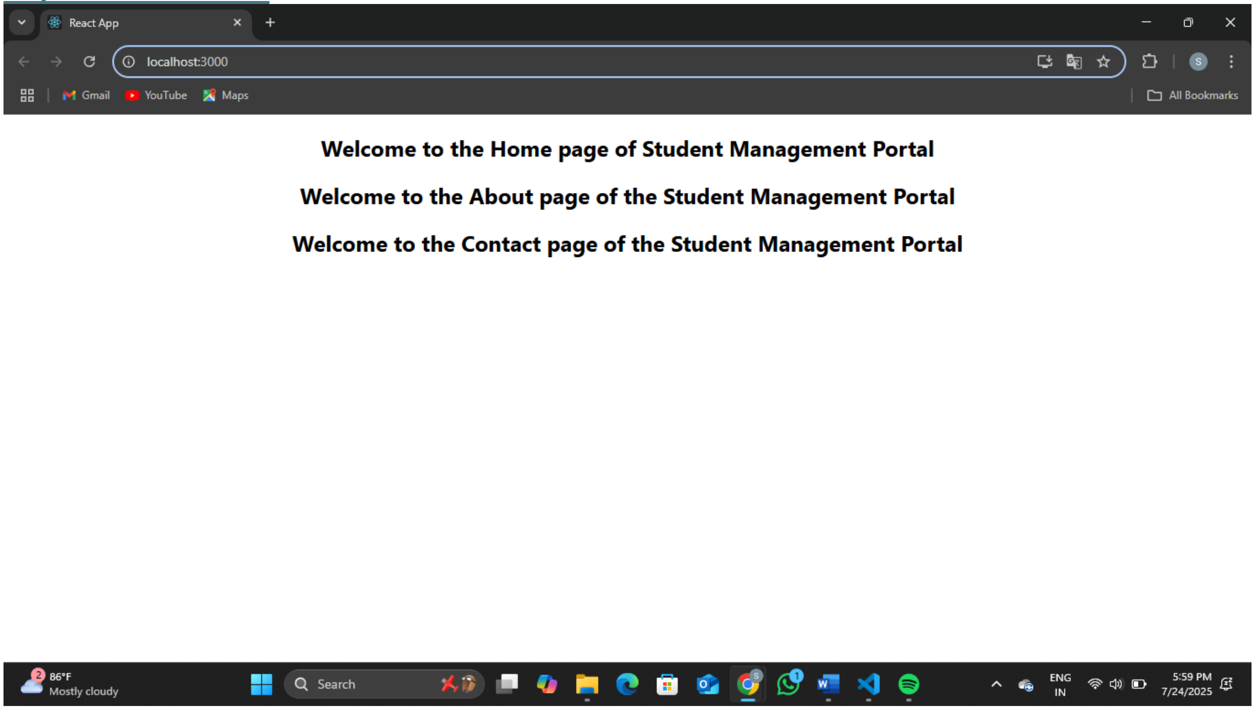
```
function App() {
  return (
    <div className="App">
      <Home />
      <About />
      <Contact />
    </div>
  );
}

export default App;
```

Output:



```
You can now view studentapp in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.1.8:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

http://localhost:3000/



Welcome to the Home page of Student Management Portal

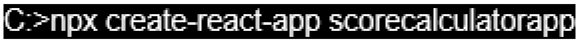
Welcome to the About page of the Student Management Portal

Welcome to the Contact page of the Student Management Portal

Exercise 3:

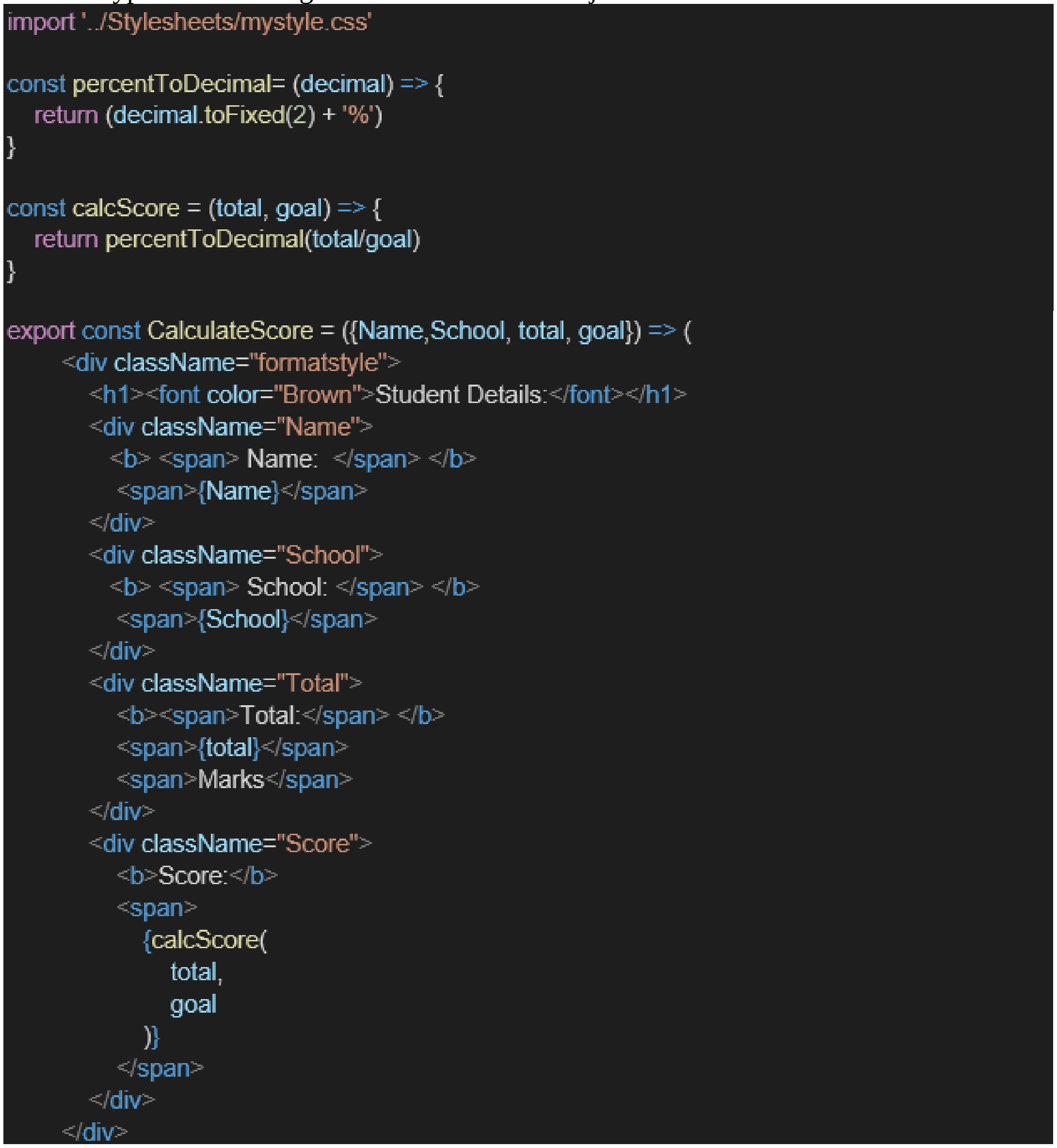
Create a react app for Student Management Portal named scorecalculatorapp and create a function component named “CalculateScore” which will accept Name, School, Total and goal in order to calculate the average score of a student and display the same.

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



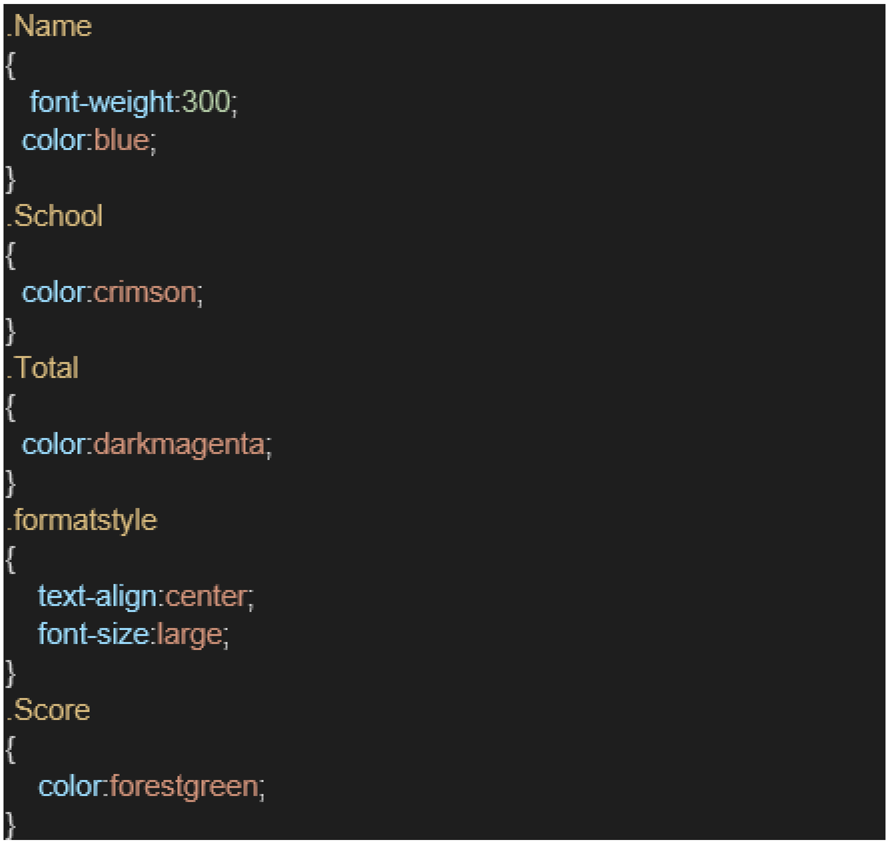
```
C:>npx create-react-app scorecalculatorapp
```

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



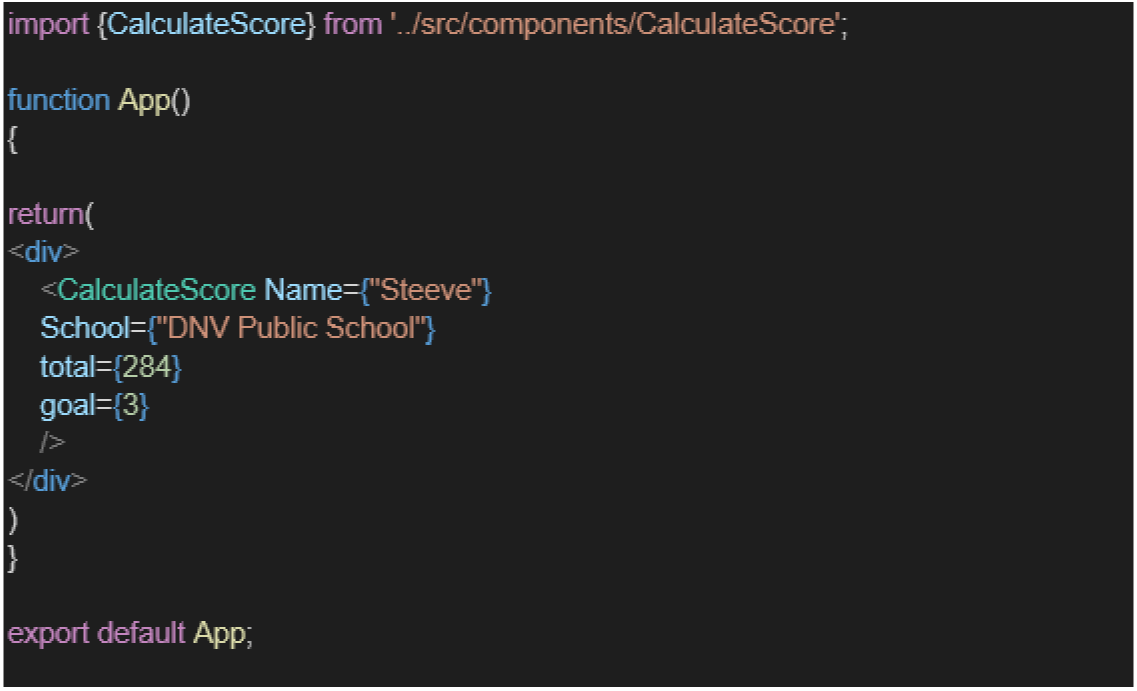
```
import '../Stylesheets/mystyle.css'

const percentToDecimal= (decimal) => {
    return (decimal.toFixed(2) + '%')
}

const calcScore = (total, goal) => {
    return percentToDecimal(total/goal)
}

export const CalculateScore = ({Name,School, total, goal}) => (
        <div className="formatstyle">
            <h1><font color="Brown">Student Details:</font></h1>
            <div className="Name">
               <b> <span> Name:  </span> </b>
                <span>{Name}</span>
            </div>
            <div className="School">
               <b> <span> School: </span> </b>
                <span>{School}</span>
            </div>
            <div className="Total">
                <b><span>Total:</span> </b>
                <span>{total}</span>
                <span>Marks</span>
            </div>
            <div className="Score">
                <b>Score:</b>
                <span>
                    {calcScore(
                        total,
                        goal
                    )}
                </span>
            </div>
        </div>
```

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



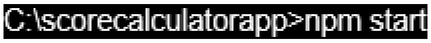
```
.Name
{
   font-weight:300;
  color:blue;
}
.School
{
  color:crimson;
}
.Total
{
  color:darkmagenta;
}
.formatstyle
{
    text-align:center;
    font-size:large;
}
.Score
{
    color:forestgreen;
}
```

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



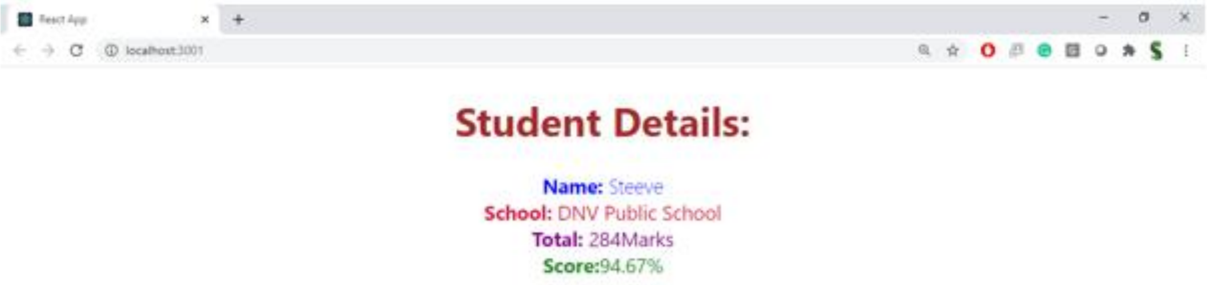
```
import {CalculateScore} from '../src/components/CalculateScore';

function App()
{

return(
<div>
    <CalculateScore Name={"Steeve"}
    School={"DNV Public School"}
    total={284}
    goal={3}
    />
</div>
)
}

export default App;
```

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



```
C:\scorecalculatorapp>npm start
```

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



Student Details:

Name: Steeve
School: DNV Public School
Total: 284Marks
Score:94.67%

src/ Components/ CalculateScore.js

```
import React from 'react';
import '../Stylesheets/mystyle.css';

function CalculateScore(props) {
  const average = props.Total / 5;

  return (
    <div className="container">
      <h2>Student Score Calculator</h2>
      <p><strong>Name:</strong> {props.Name}</p>
      <p><strong>School:</strong> {props.School}</p>
      <p><strong>Total Marks:</strong> {props.Total}</p>
      <p><strong>Goal:</strong> {props.Goal}</p>
      <p><strong>Average Score:</strong> {average}</p>
    </div>
  );
}

export default CalculateScore;
```

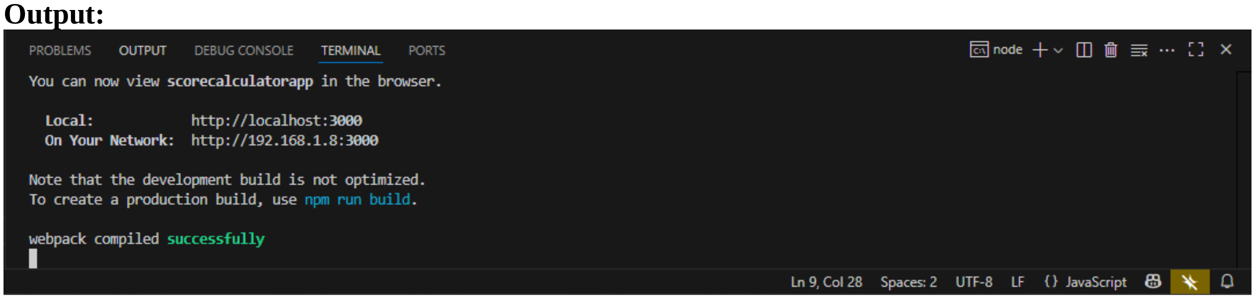
src/ Stylesheets/ mystyle.css

```
.container {
  width: 400px;
  margin: 30px auto;
  padding: 20px;
  border: 2px solid #4CAF50;
  border-radius: 10px;
  background-color: #f9f9f9;
  font-family: Arial, sans-serif;
}

.container h2 {
  text-align: center;
  color: #4CAF50;
}

.container p {
  font-size: 16px;
  margin: 8px 0;
}
```

src/App.js

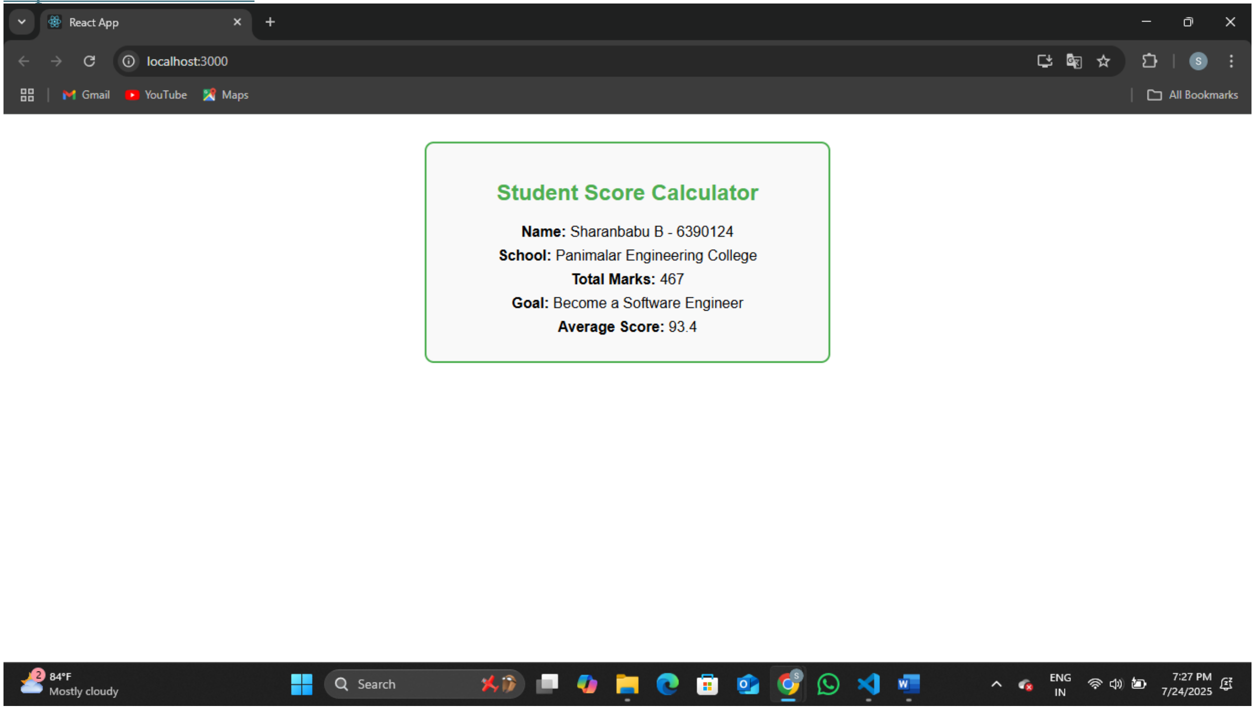
```
import React from 'react';
import './App.css';
import CalculateScore from './Components/CalculateScore';

function App() {
  return (
    <div className="App">
      <CalculateScore
        Name="Sharanbabu B - 6390124"
        School="Panimalar Engineering College"
        Total={467}
        Goal="Become a Software Engineer"
      />
    </div>
  );
}

export default App;
```

Output:



```
You can now view scorecalculatorapp in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.1.8:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

http://localhost:3000/

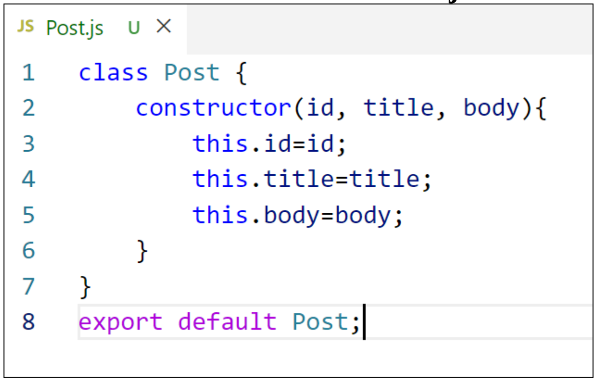


Student Score Calculator

Name: Sharanbabu B - 6390124
School: Panimalar Engineering College
Total Marks: 467
Goal: Become a Software Engineer
Average Score: 93.4

Exercise 4:

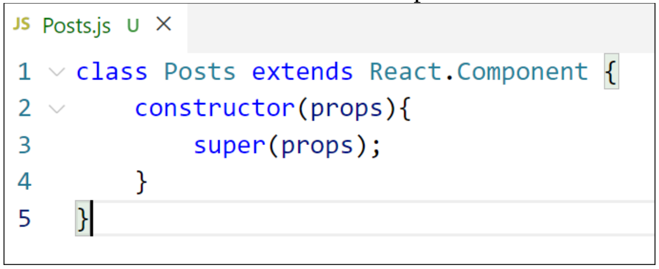
1. Create a new react application using *create-react-app* tool with the name as “blogapp”
2. Open the application using VS Code
3. Create a new file named as **Post.js** in **src folder** with following properties



```
class Post {
    constructor(id, title, body){
        this.id=id;
        this.title=title;
        this.body=body;
    }
}
export default Post;
```

Figure 1: Post class

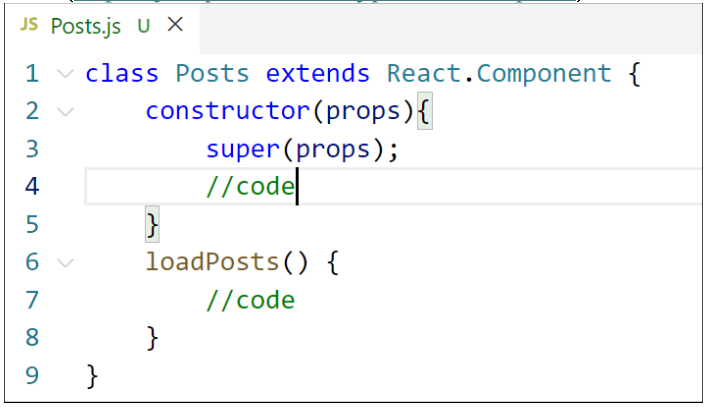
4. Create a new class based component named as **Posts** inside **Posts.js** file



```
class Posts extends React.Component {
    constructor(props){
        super(props);
    }
}
```

Figure 2: Posts Component

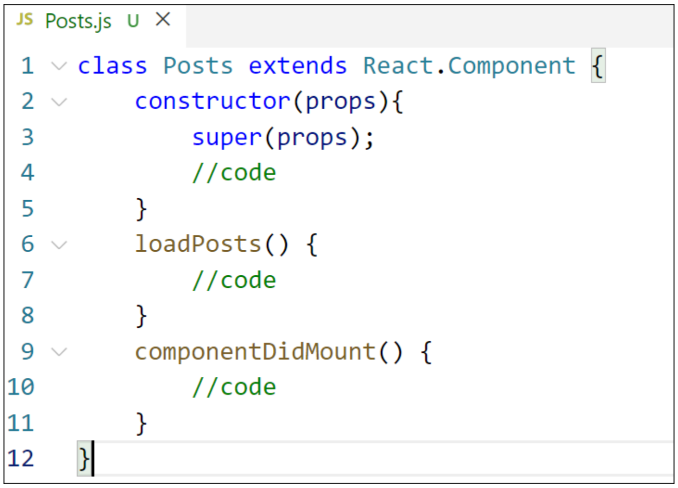
5. Initialize the component with a list of Post in state of the component using the constructor
6. Create a new method in component with the name as **loadPosts()** which will be responsible for using Fetch API and assign it to the component state created earlier. To get the posts use the url (https://jsonplaceholder.typicode.com/posts)



```
class Posts extends React.Component {
    constructor(props){
        super(props);
        //code
    }
    loadPosts() {
        //code
    }
}
```

Figure 3: loadPosts() method

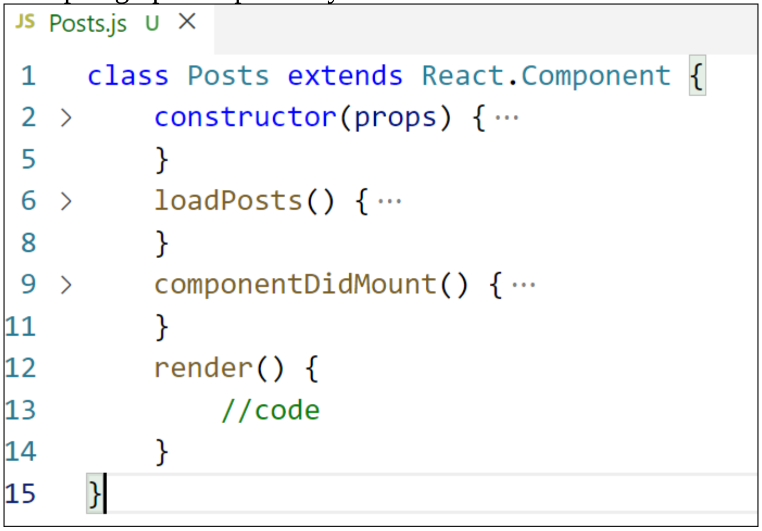
7. Implement the **componentDidMount()** hook to make calls to **loadPosts()** which will fetch the posts



```
class Posts extends React.Component {
    constructor(props){
        super(props);
        //code
    }
    loadPosts() {
        //code
    }
    componentDidMount() {
        //code
    }
}
```

Figure 4: componentDidMount() hook

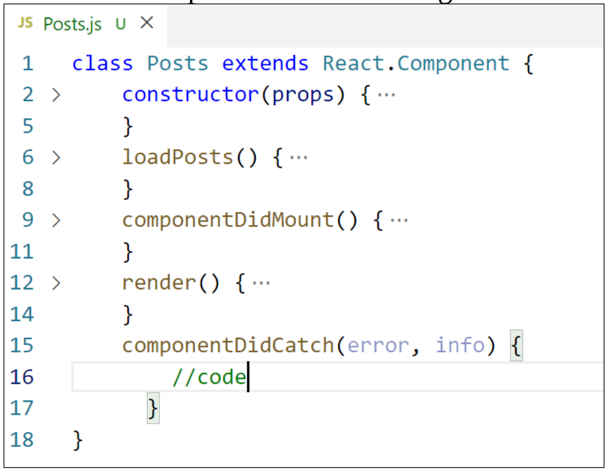
8. Implement the **render()** which will display the title and post of posts in html page using heading and paragraphs respectively.



```
class Posts extends React.Component {
    constructor(props) {⋯
    }
    loadPosts() {⋯
    }
    componentDidMount() {⋯
    }
    render() {
        //code
    }
}
```

Figure 5: render() method

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



```
class Posts extends React.Component {
    constructor(props) {⋯
    }
    loadPosts() {⋯
    }
    componentDidMount() {⋯
    }
    render() {⋯
    }
    componentDidCatch(error, info) {
        //code
      }
}
```

Figure 6: componentDidCatch() hook

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

src/ Post.js

```
class Post {
  constructor(userId, id, title, body) {
    this.userId = userId;
    this.id = id;
    this.title = title;
    this.body = body;
  }
}

export default Post;
```

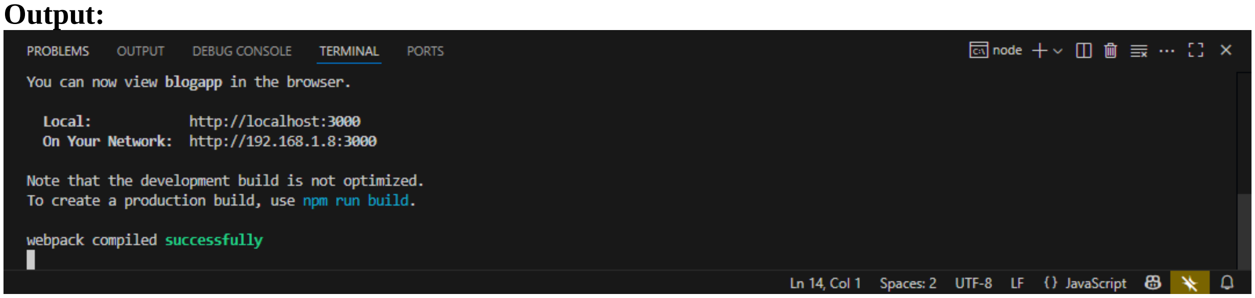
src/ Posts.js

```
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');
      if (!response.ok) throw new Error("Failed to fetch posts");
      const data = await response.json();

      const postList = data.map(post => new Post(post.userId, post.id,
post.title, post.body));
      this.setState({ posts: postList });
    } catch (err) {
      this.setState({ error: err });
    }
  }

  componentDidMount() {
    this.loadPosts();
  }

  componentDidCatch(error, info) {
    alert("An error occurred while rendering posts: " + error.toString());
  }

  render() {
    if (this.state.error) {
      return <h2>Error loading posts.</h2>;
    }

    return (
      <div>
        <h1>Blog Posts</h1>
        {this.state.posts.map(post => (
          <div key={post.id} style={{ border: "1px solid gray", padding:
"10px", margin: "10px 0" }}>
            <h3>{post.title}</h3>
            <p>{post.body}</p>
          </div>
        ))}
      </div>
    );
  }
}

export default Posts;
```

src/App.js

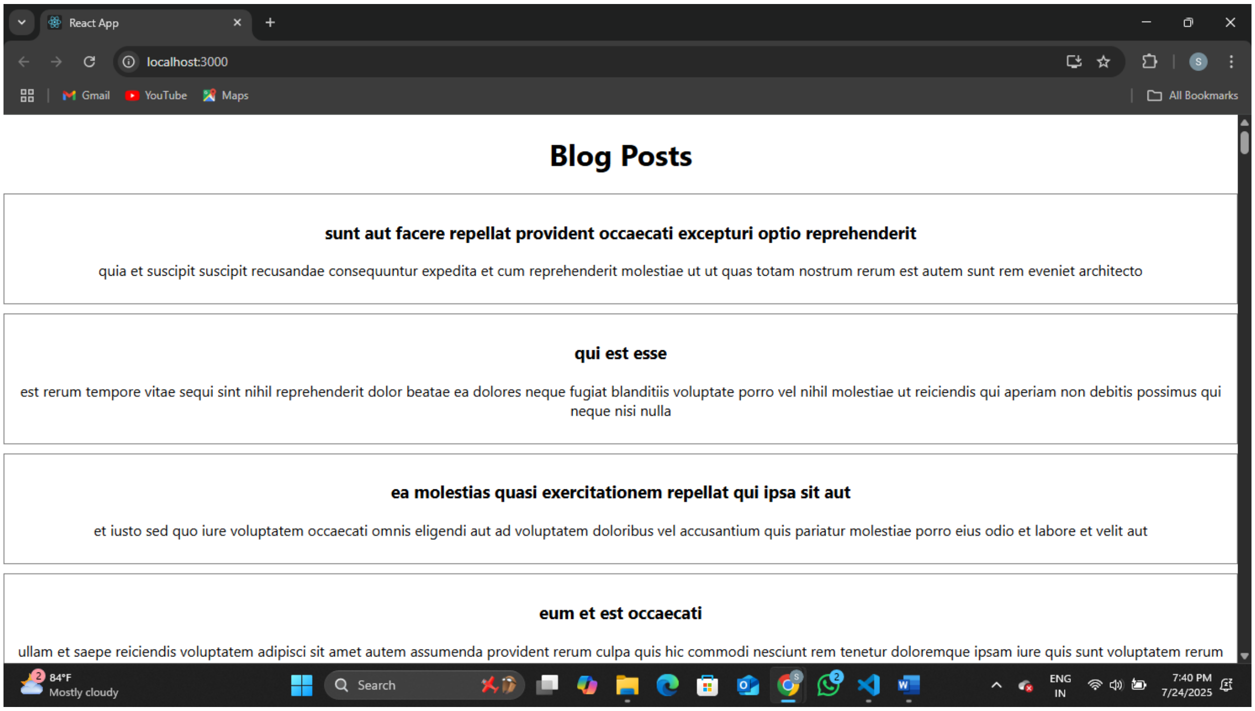
```
import React from 'react';
import './App.css';
import Posts from './Posts';

function App() {
  return (
    <div className="App">
      <Posts />
    </div>
  );
}

export default App;
```

Output:



```
You can now view blogapp in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.1.8:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

http://localhost:3000/



Blog Posts

sunt aut facere repellat provident occaecati excepturi optio reprehenderit

quia et suscipit suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut quas totam nostrum rerum est autem sunt rem eveniet architecto

qui est esse

est rerum tempore vitae sequi sint nihil reprehenderit dolor beatae ea dolores neque fugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis qui aperiam non debitis possimus qui neque nisi nulla

ea molestias quasi exercitationem repellat qui ipsa sit aut

et iusto sed quo iure voluptatem occaecati omnis eligendi aut ad voluptatem doloribus vel accusantium quis pariatur molestiae porro eius odio et labore et velit aut

eum et est occaecati

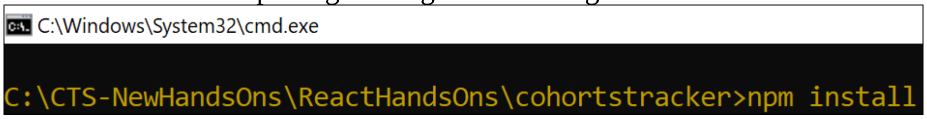
ullam et saepe reiciendis voluptatem adipisci sit amet autem assumenda provident rerum culpa quis hic commodi nesciunt rem tenetur doloremque ipsam iure quis sunt voluptatem rerum

Exercise 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.

cohorttracker.zip

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



C:\Windows\System32\cmd.exe

```
C:\CTS-NewHandsOns\ReactHandsOns\cohortstracker>npm install
```

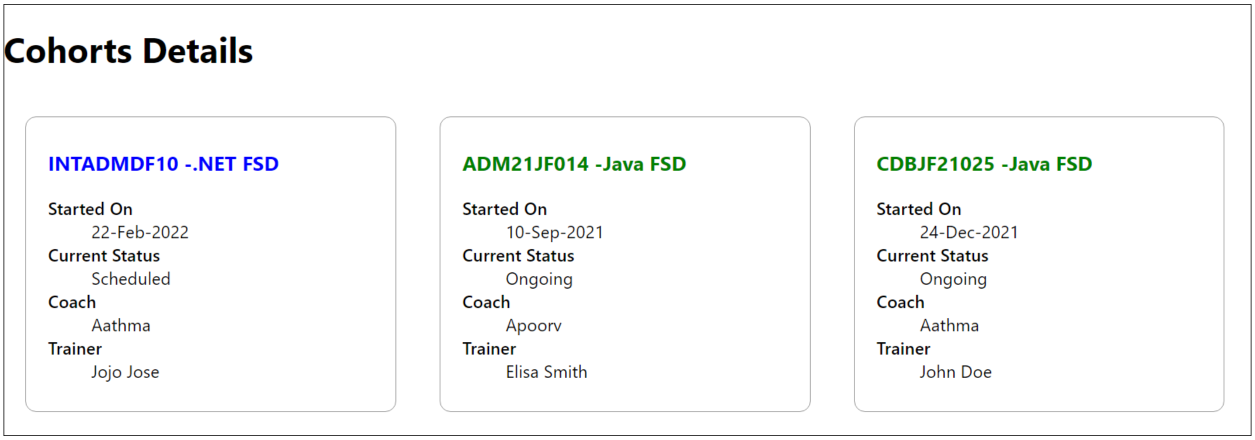
Figure 7: Restore packages

4. Open the application using VS Code
5. Create a new CSS Module in a file called “CohortDetails.module.css”
6. 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

7. Define a css style for html <dt> element using tag selector. Set the font weight to 500.
8. Open the cohort details component and import the CSS Module

9. Apply the box class to the container div
10. Define the style for <h3> element to use “green” color font when cohort status is “ongoing” and “blue” color in all other scenarios.
11. Final result should look similar to the below image



Cohorts Details

INTADMDF10 -.NET FSD

Started On
    22-Feb-2022
Current Status
    Scheduled
Coach
    Aathma
Trainer
    Jojo Jose

ADM21JF014 -Java FSD

Started On
    10-Sep-2021
Current Status
    Ongoing
Coach
    Apoorv
Trainer
    Elisa Smith

CDBJF21025 -Java FSD

Started On
    24-Dec-2021
Current Status
    Ongoing
Coach
    Aathma
Trainer
    John Doe

Figure 8: Final Result

src/ 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;
}
```

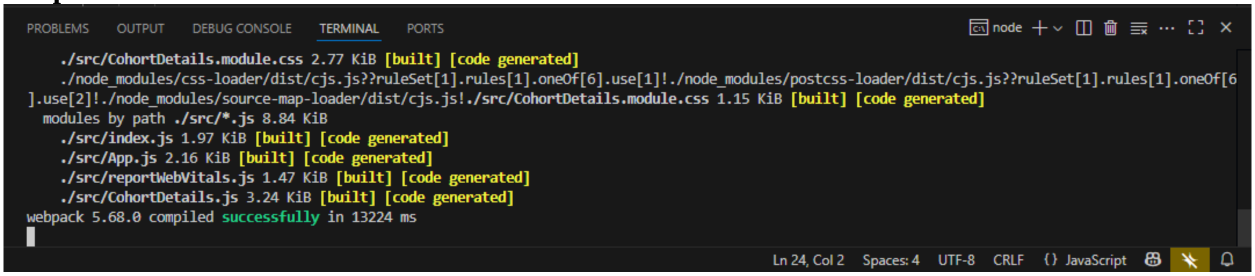
src/ CohortDetails.js

```
import React from 'react';
import styles from './CohortDetails.module.css'; // import CSS Module

function CohortDetails(props) {
    const { cohortName, startDate, endDate, status } = props;

    const headingStyle = {
        color: status === 'ongoing' ? 'green' : 'blue',
    };

    return (
        <div className={styles.box}>
            <h3 style={headingStyle}>{cohortName}</h3>
            <dl>
                <dt>Start Date:</dt>
                <dd>{startDate}</dd>
                <dt>End Date:</dt>
                <dd>{endDate}</dd>
                <dt>Status:</dt>
                <dd>{status}</dd>
            </dl>
        </div>
    );
}

export default CohortDetails;
```

src/ App.js

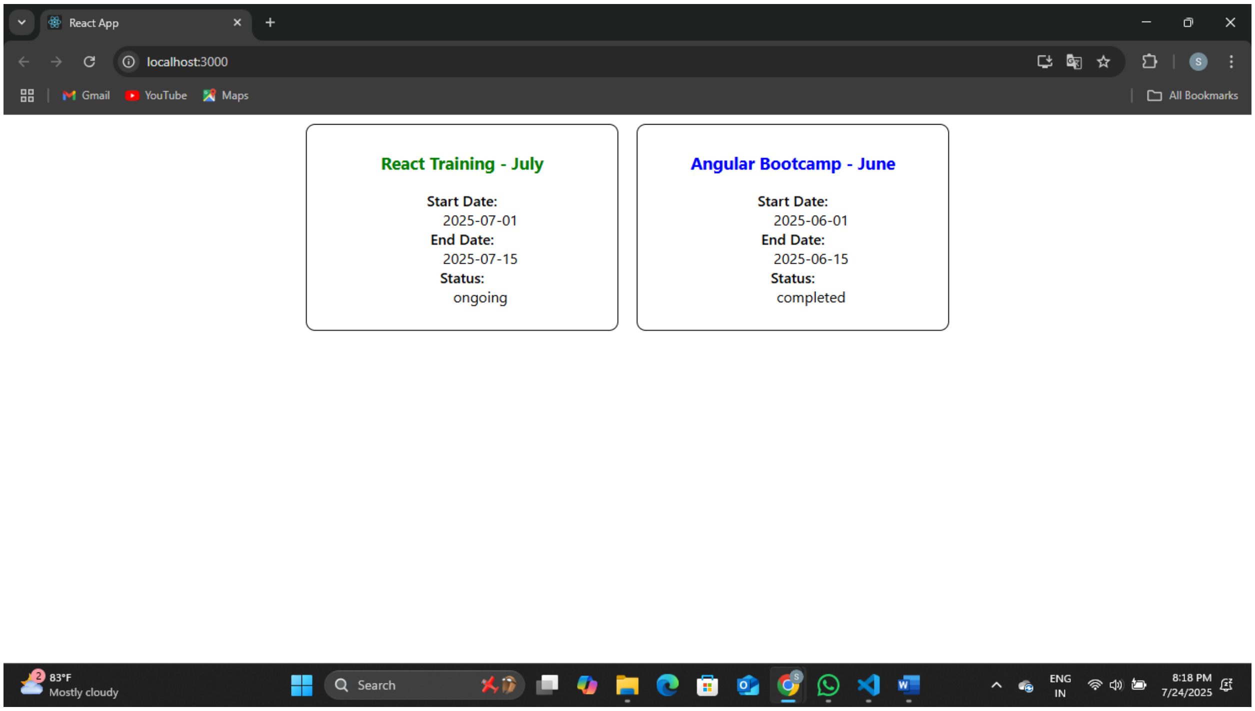
```
import React from 'react';
import './App.css';
import CohortDetails from './CohortDetails';

function App() {
  return (
    <div className="App">
      <CohortDetails
        cohortName="React Training - July"
        startDate="2025-07-01"
        endDate="2025-07-15"
        status="ongoing"
      />
      <CohortDetails
        cohortName="Angular Bootcamp - June"
        startDate="2025-06-01"
        endDate="2025-06-15"
        status="completed"
      />
    </div>
  );
}

export default App;
```

Output:



```
    ./src/CohortDetails.module.css 2.77 KiB [built] [code generated]
    ./node_modules/css-loader/dist/cjs.js??ruleSet[1].rules[1].oneOf[6].use[1]!./node_modules/postcss-loader/dist/cjs.js??ruleSet[1].rules[1].oneOf[6
].use[2]!./node_modules/source-map-loader/dist/cjs.js!./src/CohortDetails.module.css 1.15 KiB [built] [code generated]
  modules by path ./src/*.js 8.84 KiB
    ./src/index.js 1.97 KiB [built] [code generated]
    ./src/App.js 2.16 KiB [built] [code generated]
    ./src/reportWebVitals.js 1.47 KiB [built] [code generated]
    ./src/CohortDetails.js 3.24 KiB [built] [code generated]
webpack 5.68.0 compiled successfully in 13224 ms
```

http://localhost:3000/



React Training - July

Start Date:
2025-07-01
End Date:
2025-07-15
Status:
ongoing

Angular Bootcamp - June

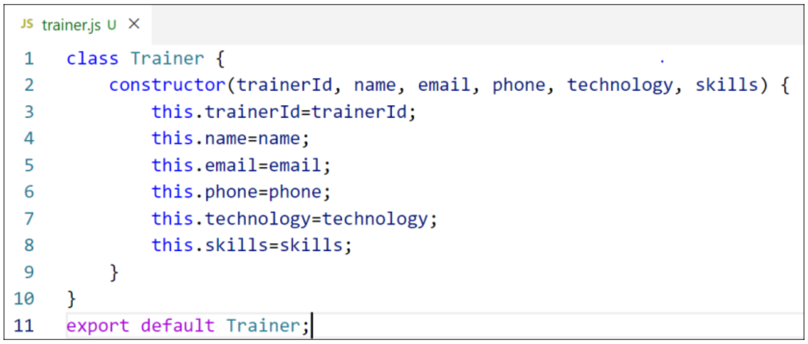
Start Date:
2025-06-01
End Date:
2025-06-15
Status:
completed

Exercise 6:

Cognizant Academy teams want to maintain a list of trainers along with their expertise in a SPA using React as the technology. You are assigned the task of creating this React app.
The following trainers' data application will deal.

1. T-ID
2. Name
3. Phone
4. Email
5. Stream
6. Skills

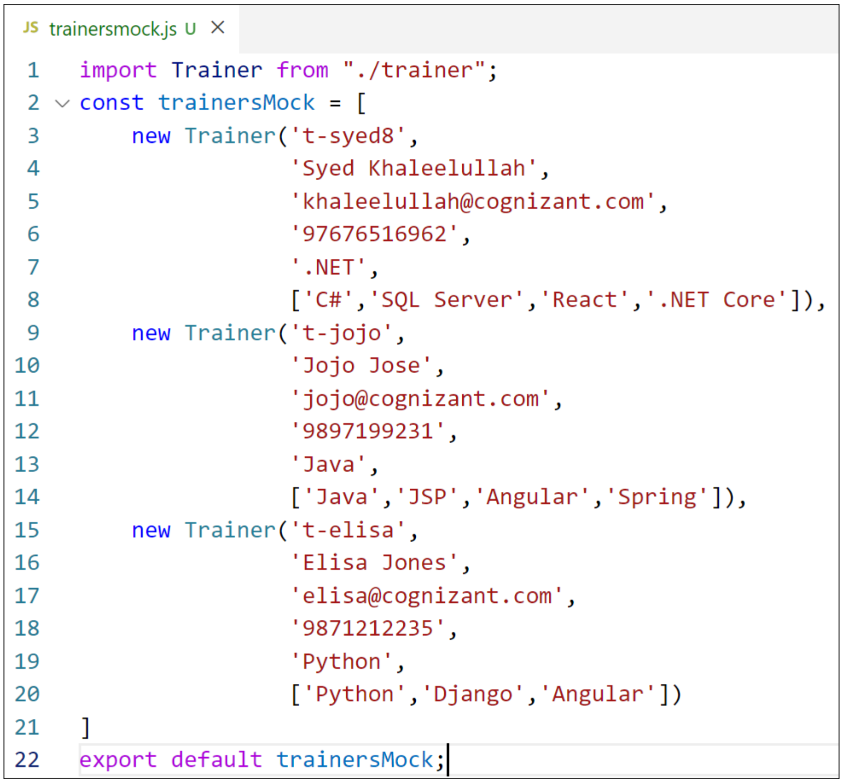
1. Create a new React app using *create-react-app* tool with the as "TrainersApp"
2. Open the application using the VS Code
3. Add a new file called *trainer.js* inside the **src folder** and define a class named as "Trainer" with the following properties
   a. TrainerId
   b. Name
   c. Email
   d. Phone
   e. Technology
   f. Skills



```
class Trainer {
    constructor(trainerId, name, email, phone, technology, skills) {
        this.trainerId=trainerId;
        this.name=name;
        this.email=email;
        this.phone=phone;
        this.technology=technology;
        this.skills=skills;
    }
}
export default Trainer;
```

Figure 9: Trainer.js

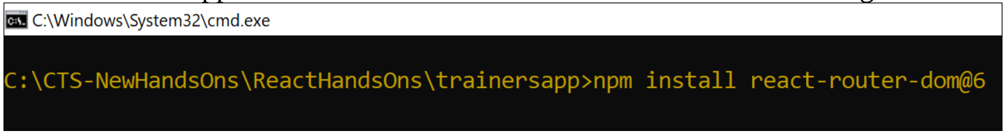
4. Create a new TrainersMock.js file which will contain the mock trainer data. Refer the following screenshot for mock data



```
import Trainer from "./trainer";
const trainersMock = [
    new Trainer('t-syed8',
                'Syed Khaleelullah',
                'khaleelullah@cognizant.com',
                '97676516962',
                '.NET',
                ['C#','SQL Server','React','.NET Core']),
    new Trainer('t-jojo',
                'Jojo Jose',
                'jojo@cognizant.com',
                '9897199231',
                'Java',
                ['Java','JSP','Angular','Spring']),
    new Trainer('t-elisa',
                'Elisa Jones',
                'elisa@cognizant.com',
                '9871212235',
                'Python',
                ['Python','Django','Angular'])
]
export default trainersMock;
```

Figure 10: TrainersMock.js

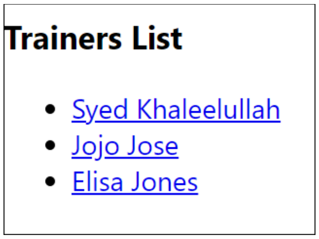
5. Install the support for React router for the dom. Execute the following command.



```
C:\CTS-NewHandsOns\ReactHandsOns\trainersapp>npm install react-router-dom@6
```

Figure 11: Install React Router

6. Create new component named as **TrainersList** inside *Trainerlist.js* file. The component should accept the trainer’s data as parameter and render it as a list. The list should display names of each trainers which must be clickable like a hyper link. Refer the following screenshot for the component layout.

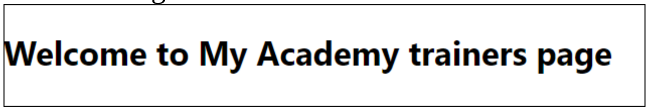


Trainers List

- Syed Khaleelullah
- Jojo Jose
- Elisa Jones

Figure 12: TrainersList Component

7. Create a new component named as Home inside Home.js which will be responsible for displaying the following



Welcome to My Academy trainers page

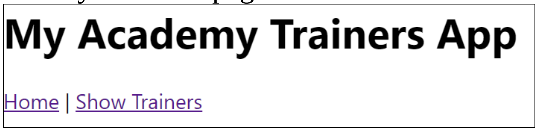
Figure 13: Home Component

8. Modify the App component to add support for routing and defining the navigation links to Home component and TrainersList component. Use BrowserRouter, Routes, Route and Link components from the react-router-dom library.

Define the following URL

1. / - must redirect to home component
2. /trainers – must redirect to trainers list component.

The layout of the page must be similar to the following



My Academy Trainers App

Home | Show Trainers

Figure 14: App Component

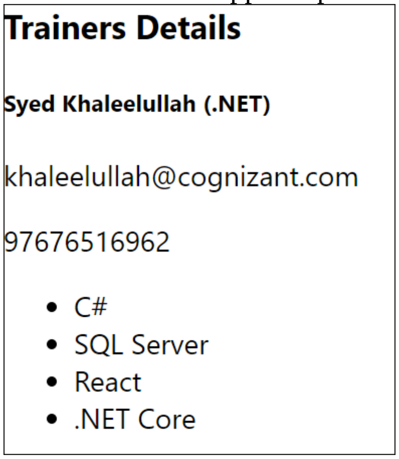
9. Create a new component named **TrainerDetail** in *TrainerDetails.js* file.

The component should retrieve a parameter named id from the URL with the help of “useParams” hook from the React router DOM library.

It should query the mock trainer data using the id and display the trainer details as show in screenshot.

Modify the TrainersList component to add Links to TrainerDetail component while passing the ID.

Define a route in App component for the same.



Trainers Details

Syed Khaleelullah (.NET)

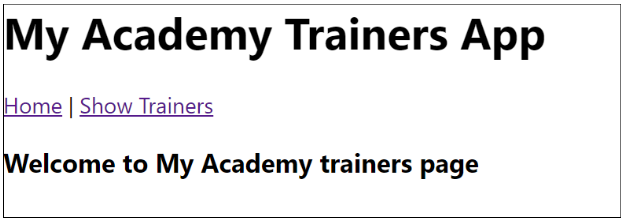
khaleelullah@cognizant.com

97676516962

- C#
- SQL Server
- React
- .NET Core

Figure 15: Trainers Detail Component

10. Build and run the application. The complete layout of the application will look as follows.

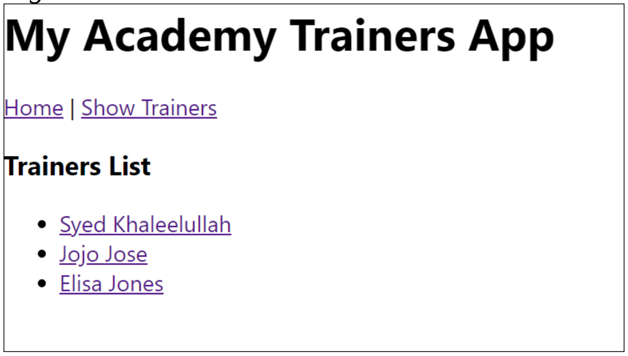


My Academy Trainers App

Home | Show Trainers

Welcome to My Academy trainers page

Figure 16: Home



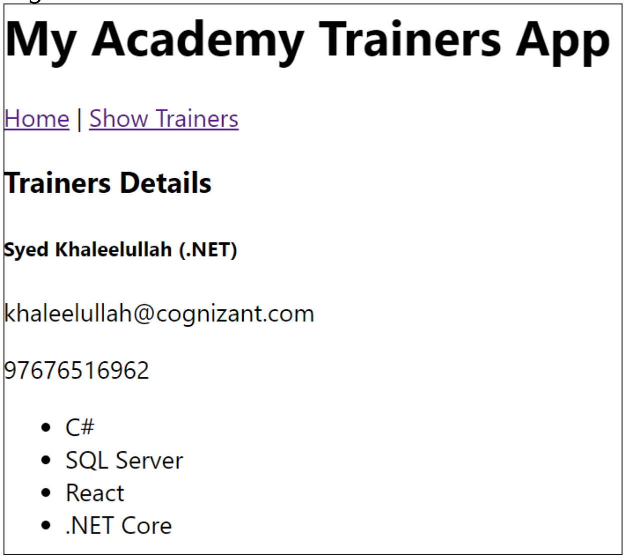
My Academy Trainers App

Home | Show Trainers

Trainers List

- Syed Khaleelullah
- Jojo Jose
- Elisa Jones

Figure 17: Trainers List



My Academy Trainers App

Home | Show Trainers

Trainers Details

Syed Khaleelullah (.NET)

khaleelullah@cognizant.com

97676516962

- C#
- SQL Server
- React
- .NET Core

Figure 18: Trainer Details

src/ Trainer.js

```
class Trainer {
  constructor(trainerId, name, phone, email, technology, skills) {
    this.trainerId = trainerId;
    this.name = name;
    this.phone = phone;
    this.email = email;
    this.technology = technology;
    this.skills = skills;
  }
}

export default Trainer;
```

src/ TrainersMock.js

```
import Trainer from './Trainer';

const trainers = [
  new Trainer(1, "John Doe", "1234567890", "john@example.com", "React",
"React, JavaScript"),
  new Trainer(2, "Jane Smith", "9876543210", "jane@example.com", "Angular",
"Angular, TypeScript"),
  new Trainer(3, "Bob Johnson", "4567891230", "bob@example.com", "Vue", "Vue,
HTML, CSS")
];

export default trainers;
```

src/ TrainerList.js

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

function TrainerList({ trainers }) {
  return (
    <div>
      <h2>Trainer List</h2>
      <ul>
        {trainers.map(trainer => (
          <li key={trainer.trainerId}>
            <Link to={`/trainer/${trainer.trainerId}`}>{trainer.name}</Link>
          </li>
        ))}
      </ul>
    </div>
  );
}

export default TrainerList;
```

src/ TrainerDetails.js

```
import React from 'react';
import { useParams } from 'react-router-dom';
import trainers from './TrainersMock';

function TrainerDetails() {
  const { id } = useParams();
  const trainer = trainers.find(t => t.trainerId === parseInt(id));

  if (!trainer) return <p>Trainer not found.</p>;

  return (
    <div>
      <h2>{trainer.name}</h2>
      <p><strong>Phone:</strong> {trainer.phone}</p>
      <p><strong>Email:</strong> {trainer.email}</p>
      <p><strong>Technology:</strong> {trainer.technology}</p>
      <p><strong>Skills:</strong> {trainer.skills}</p>
    </div>
  );
}
```

```
export default TrainerDetails;
```

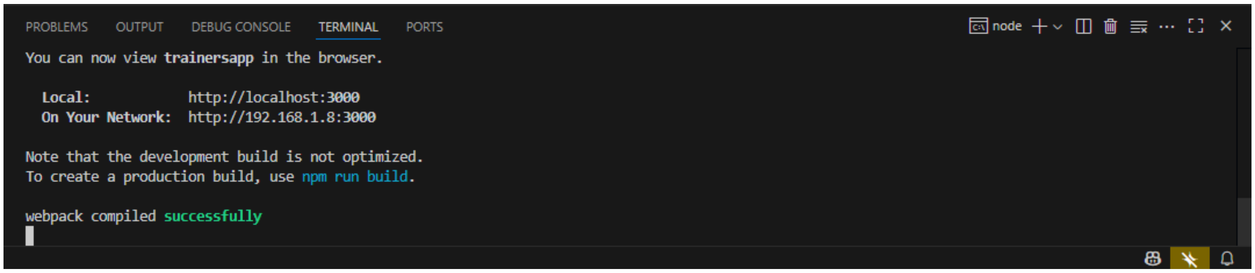
src/ Home.js

```
import React from 'react';

function Home() {
  return (
    <div>
      <h2>Welcome to Trainers App Dashboard</h2>
      <p>Use the navigation menu to explore trainer details.</p>
    </div>
  );
}

export default Home;
```

src/ App.js

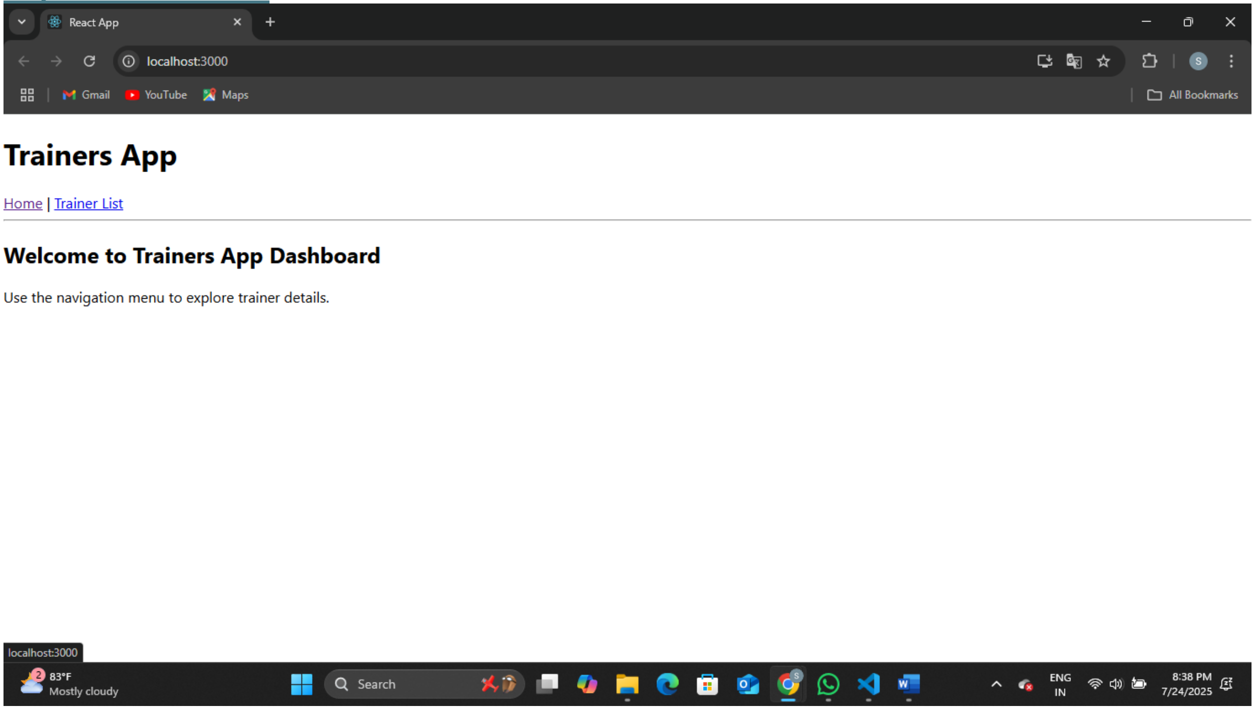
```
import React from 'react';
import { BrowserRouter as Router, Routes, Route, Link } from 'react-router-
dom';
import Home from './Home';
import TrainerList from './TrainerList';
import TrainerDetails from './TrainerDetails';
import trainers from './TrainersMock';

function App() {
  return (
    <Router>
      <div>
        <h1>Trainers App</h1>
        <nav>
          <Link to="/">Home</Link> | <Link to="/trainers">Trainer List</Link>
        </nav>
        <hr />
        <Routes>
          <Route path="/" element={<Home />} />
          <Route path="/trainers" element={<TrainerList trainers={trainers}
/>} />
          <Route path="/trainer/:id" element={<TrainerDetails />} />
        </Routes>
      </div>
    </Router>
  );
}

export default App;
```

Output:



```
You can now view trainersapp in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.1.8:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

http://localhost:3000/

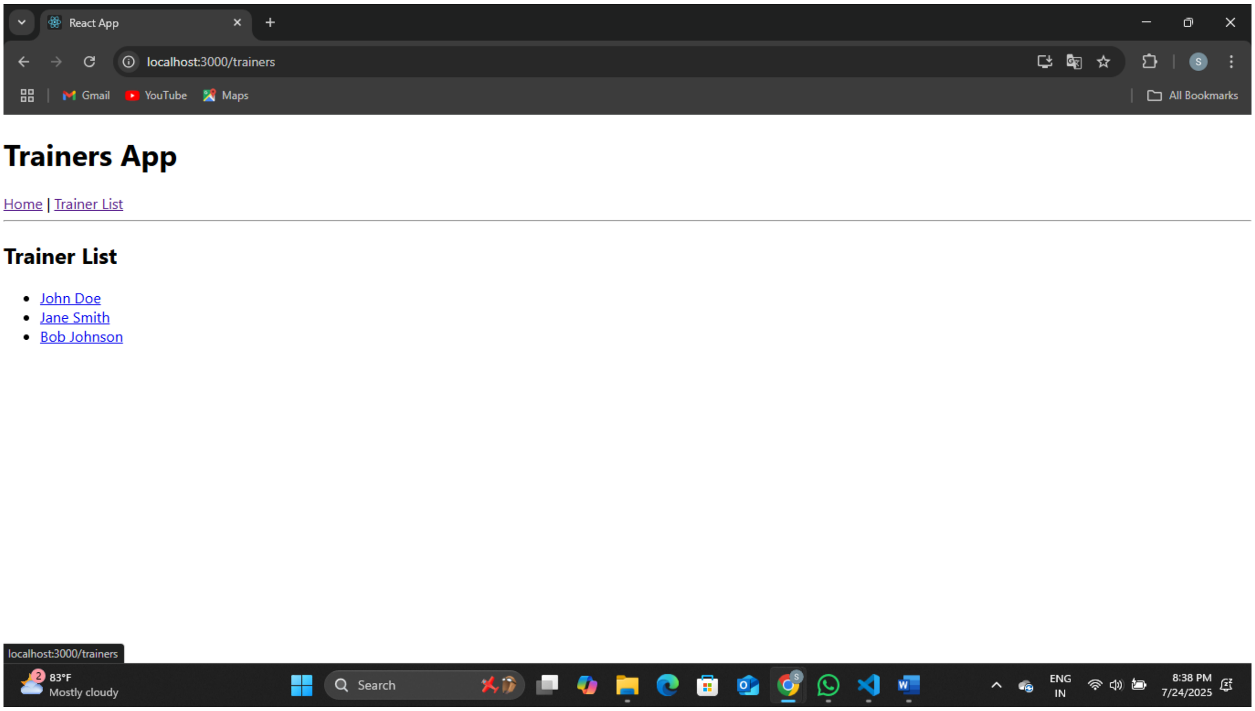


Trainers App

Home | Trainer List

Welcome to Trainers App Dashboard

Use the navigation menu to explore trainer details.

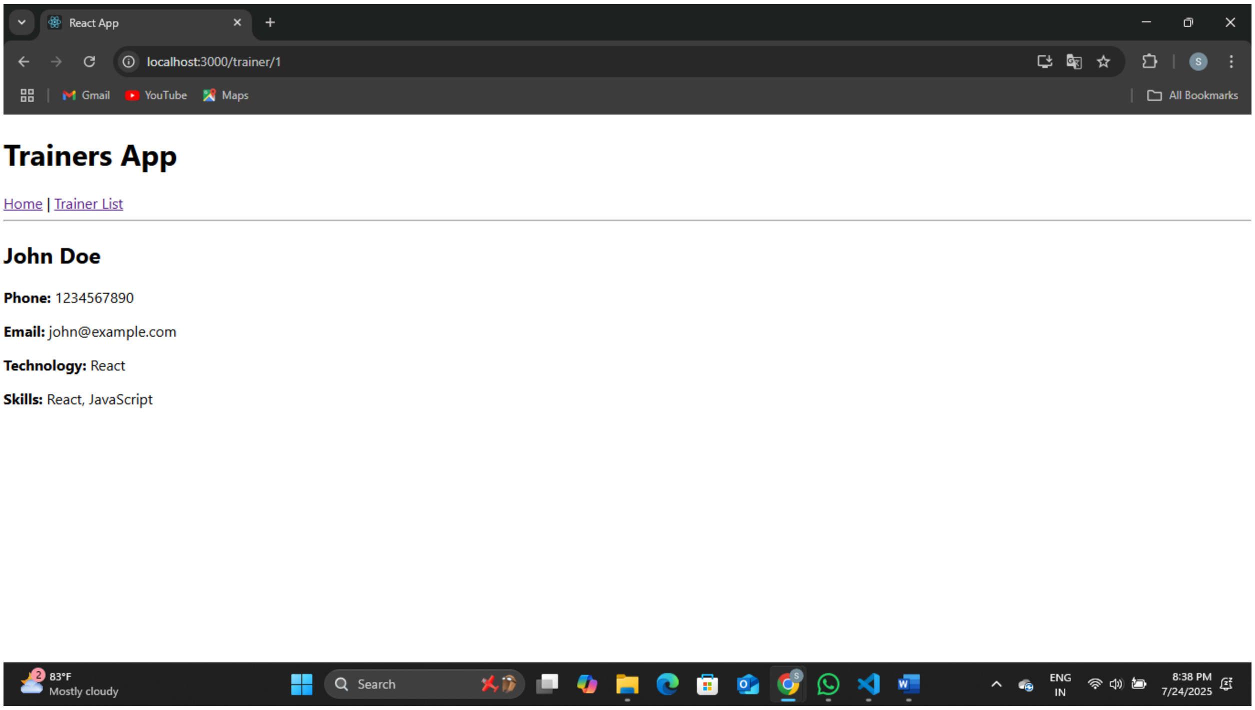


Trainers App

Home | Trainer List

Trainer List

- John Doe
- Jane Smith
- Bob Johnson



Trainers App

Home | Trainer List

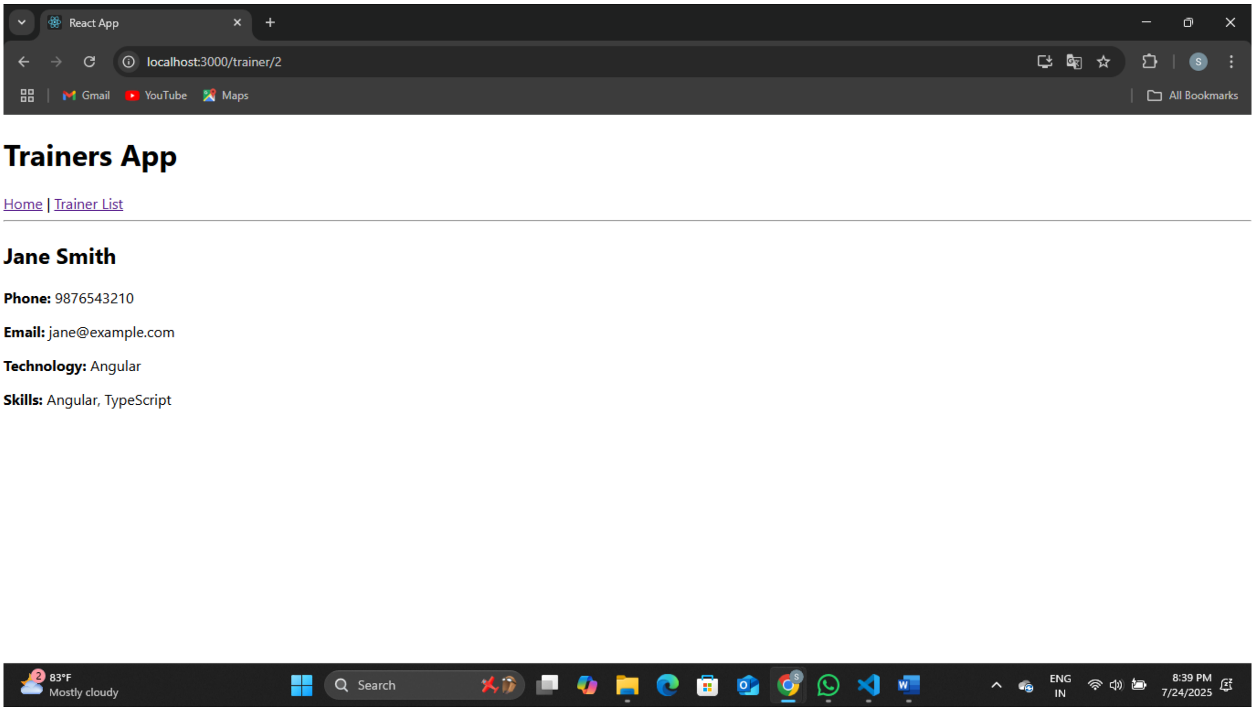
John Doe

Phone: 1234567890

Email: john@example.com

Technology: React

Skills: React, JavaScript



Trainers App

Home | Trainer List

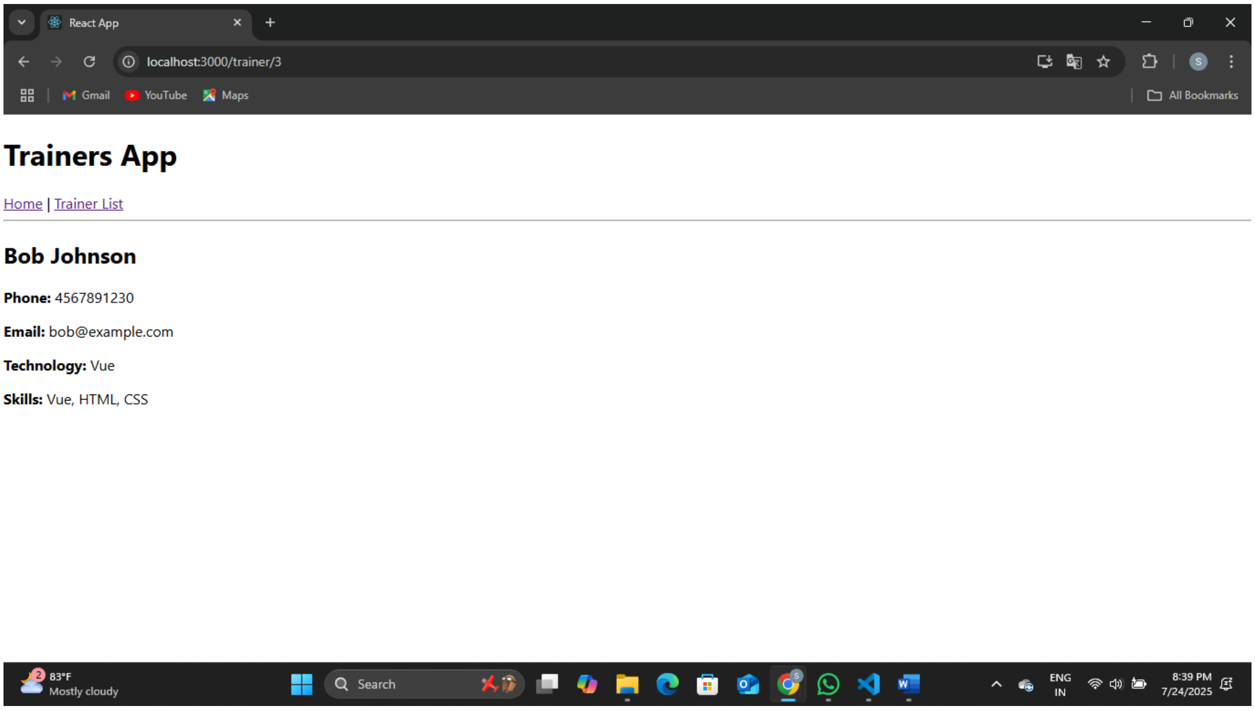
Jane Smith

Phone: 9876543210

Email: jane@example.com

Technology: Angular

Skills: Angular, TypeScript



Trainers App

Home | Trainer List

Bob Johnson

Phone: 4567891230

Email: bob@example.com

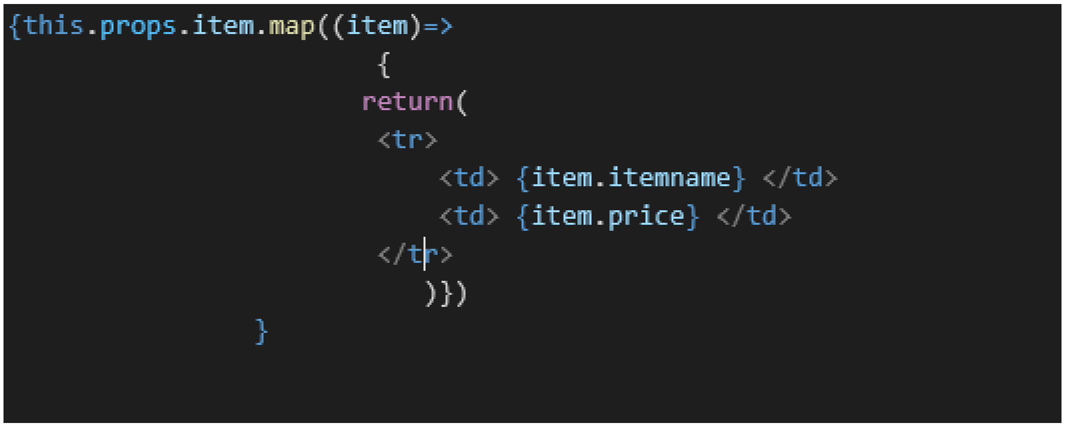
Technology: Vue

Skills: Vue, HTML, CSS

Exercise 7:

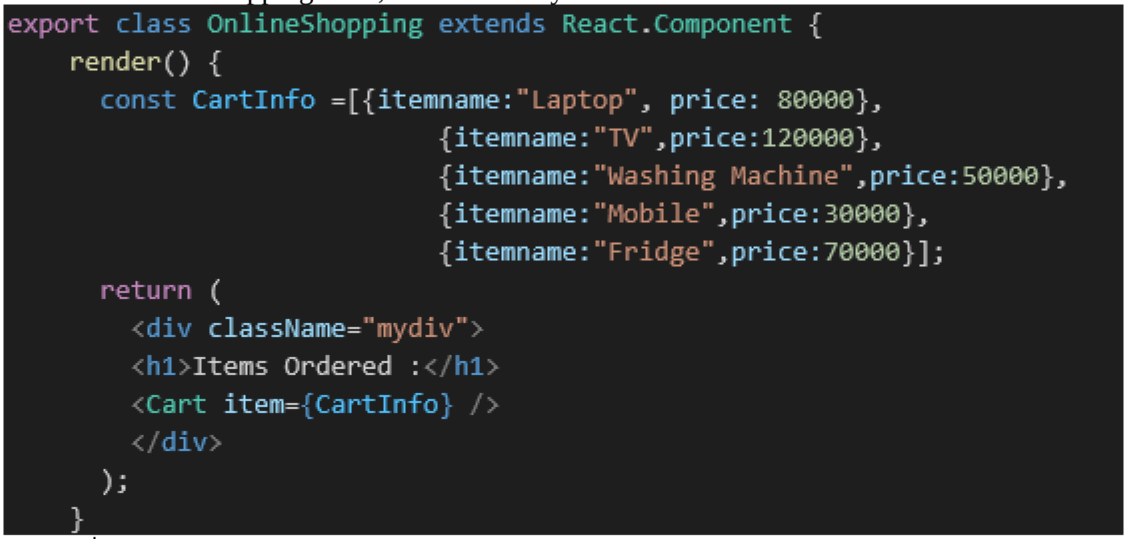
Create a React Application named “shoppingapp” with a class component named “OnlineShopping” and “Cart”.

1. In Cart class, create 2 properties as mentioned below:
    - Itemname
    - Price



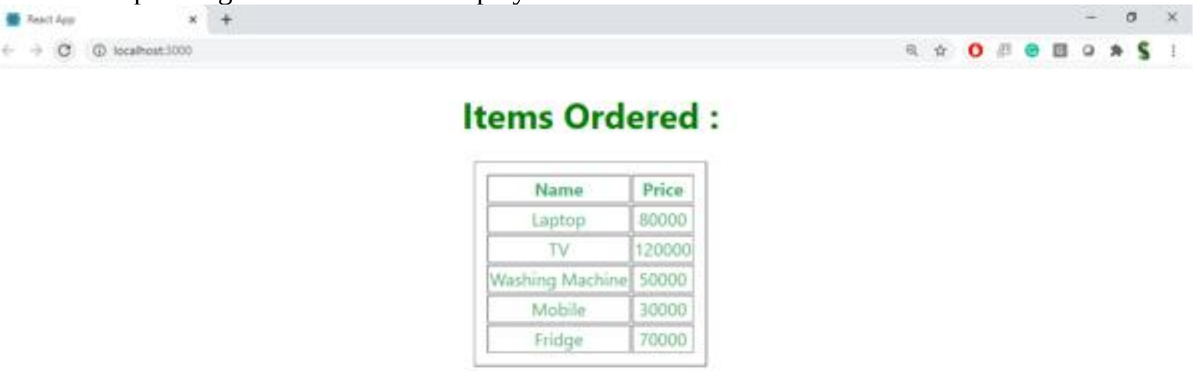
```
{this.props.item.map((item)=>
                        {
                       return(
                        <tr>
                            <td> {item.itemname} </td>
                            <td> {item.price} </td>
                        </tr>
                           )})
                }
```

2. In OnlineShopping class, create an array of Cart and initialize 5 items.



```
export class OnlineShopping extends React.Component {
    render() {
      const CartInfo =[{itemname:"Laptop", price: 80000},
                            {itemname:"TV",price:120000},
                            {itemname:"Washing Machine",price:50000},
                            {itemname:"Mobile",price:30000},
                            {itemname:"Fridge",price:70000}];
      return (
        <div className="mydiv">
        <h1>Items Ordered :</h1>
        <Cart item={CartInfo} />
        </div>
      );
    }
```

3. Loop through these items and display the data as shown below:



Items Ordered :

Name	Price
Laptop	80000
TV	120000
Washing Machine	50000
Mobile	30000
Fridge	70000

src/ Cart.js

```
import React from 'react';

class Cart extends React.Component {
  render() {
    return (
      <div style={{
        border: "1px solid black",
        padding: "10px",
        margin: "10px",
        width: "300px"
      }}>
        <h3>Item: {this.props.itemname}</h3>
        <p>Price: ₹{this.props.price}</p>
      </div>
    );
  }
}

export default Cart;
```

src/ OnlineShopping.js

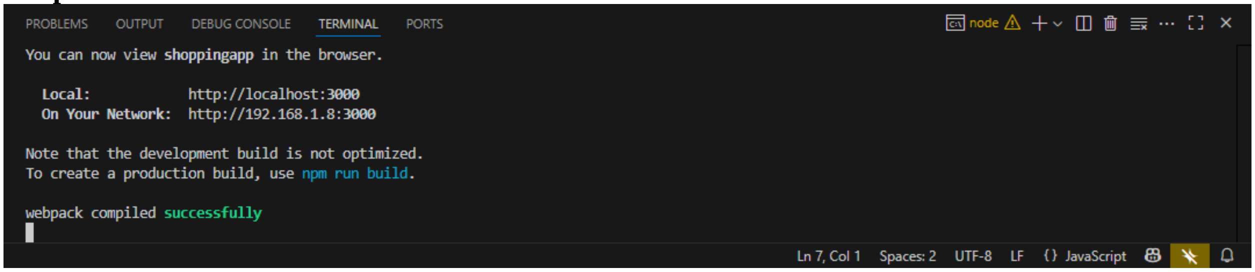
```
import React from 'react';
import Cart from './Cart';

class OnlineShopping extends React.Component {
  constructor(props) {
    super(props);
    this.items = [
      { itemname: "T-shirt", price: 499 },
      { itemname: "Jeans", price: 899 },
      { itemname: "Shoes", price: 1299 },
      { itemname: "Backpack", price: 699 },
      { itemname: "Watch", price: 1499 }
    ];
  }

  render() {
    return (
      <div>
        <h2>Welcome to the Online Shopping Portal</h2>
        {this.items.map((item, index) => (
          <Cart key={index} itemname={item.itemname} price={item.price} />
        ))}
      </div>
    );
  }
}

export default OnlineShopping;
```

src/ index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import OnlineShopping from './OnlineShopping';
```

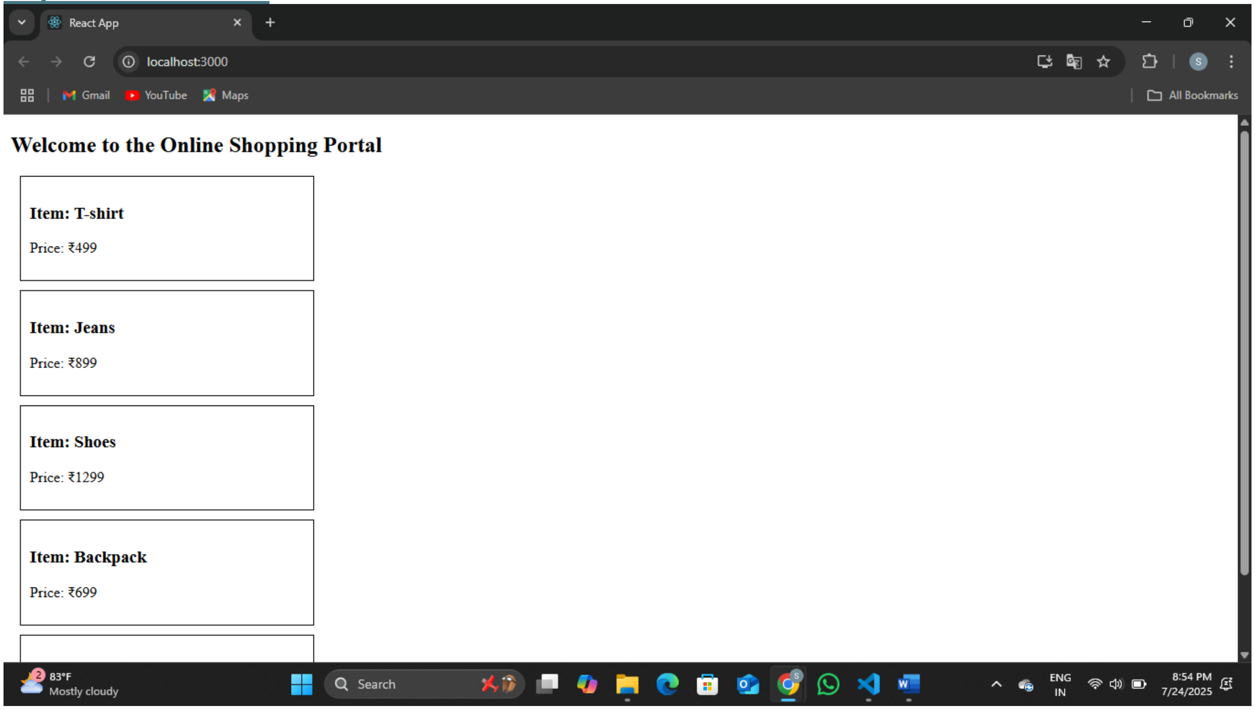
```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<OnlineShopping />);
```

Output:



```
You can now view shoppingapp in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.1.8:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

http://localhost:3000/



Welcome to the Online Shopping Portal

Item: T-shirt

Price: ₹499

Item: Jeans

Price: ₹899

Item: Shoes

Price: ₹1299

Item: Backpack

Price: ₹699

Exercise 8:

Create a React App “counterapp” which will have a component named “CountPeople” which will have 2 methods.

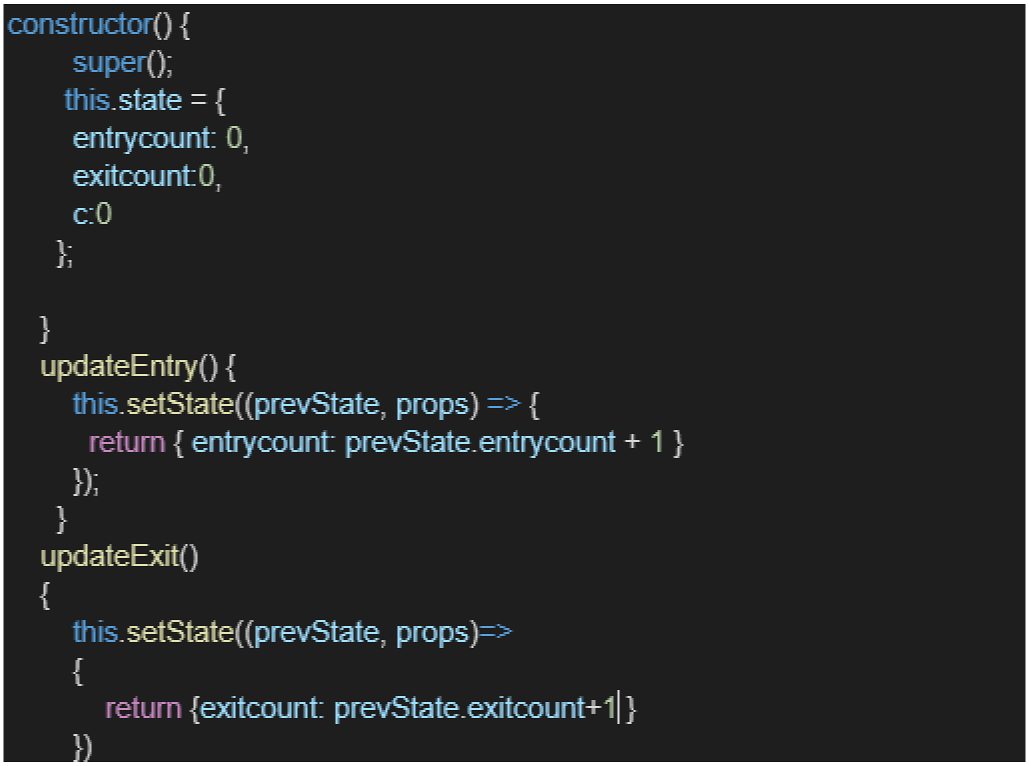
UpdateEntry() → which will display the number of people who entered the mall.

UpdateExit() → which will display the number of people who exited the mall.

Use Constructor and state to Store the entrycount and exitcount.

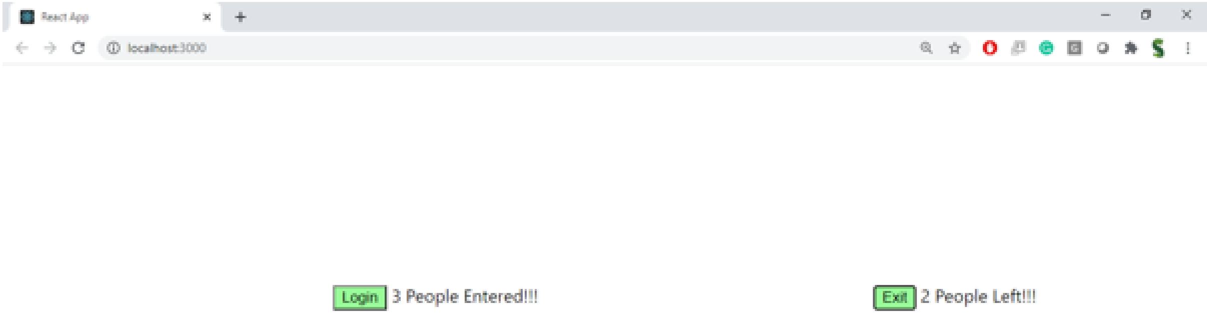
The component has 2 buttons

1. Login → when clicked, the entrycount should get incremented by 1
2. Exit → when clicked, the exitcount should get incremented by 1



```
constructor() {
    super();
   this.state = {
    entrycount: 0,
    exitcount:0,
    c:0
   };

  }
  updateEntry() {
    this.setState((prevState, props) => {
      return { entrycount: prevState.entrycount + 1 }
    });
   }
  updateExit()
  {
    this.setState((prevState, props)=>
    {
      return {exitcount: prevState.exitcount+1 }
    })
```

The output should be as follows:



3 People Entered!!!

2 People Left!!!

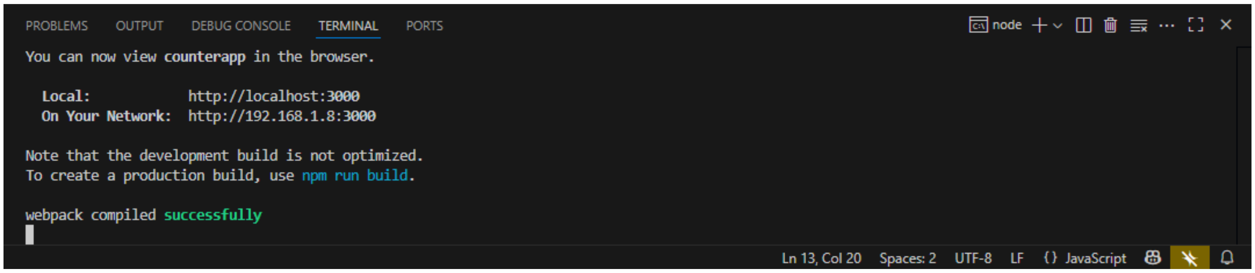
src/ CountPeople.js

```
import React from 'react';

class CountPeople extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      entryCount: 0,
      exitCount: 0
    };
  }

  UpdateEntry = () => {
    this.setState(prevState => ({
      entryCount: prevState.entryCount + 1
    }));
  }

  UpdateExit = () => {
    this.setState(prevState => ({
      exitCount: prevState.exitCount + 1
    }));
  }

  render() {
    return (
      <div style={{ textAlign: 'center', marginTop: '50px' }}>
        <h2>People Counter for Shopping Mall</h2>
        <p><strong>Number of People Entered:</strong>
{this.state.entryCount}</p>
        <p><strong>Number of People Exited:</strong>
{this.state.exitCount}</p>

        <button onClick={this.UpdateEntry} style={{ margin: '10px', padding:
'10px' }}>
          Login
        </button>
        <button onClick={this.UpdateExit} style={{ margin: '10px', padding:
'10px' }}>
          Exit
        </button>
      </div>
    );
  }
}

export default CountPeople;
```

src/ App.js

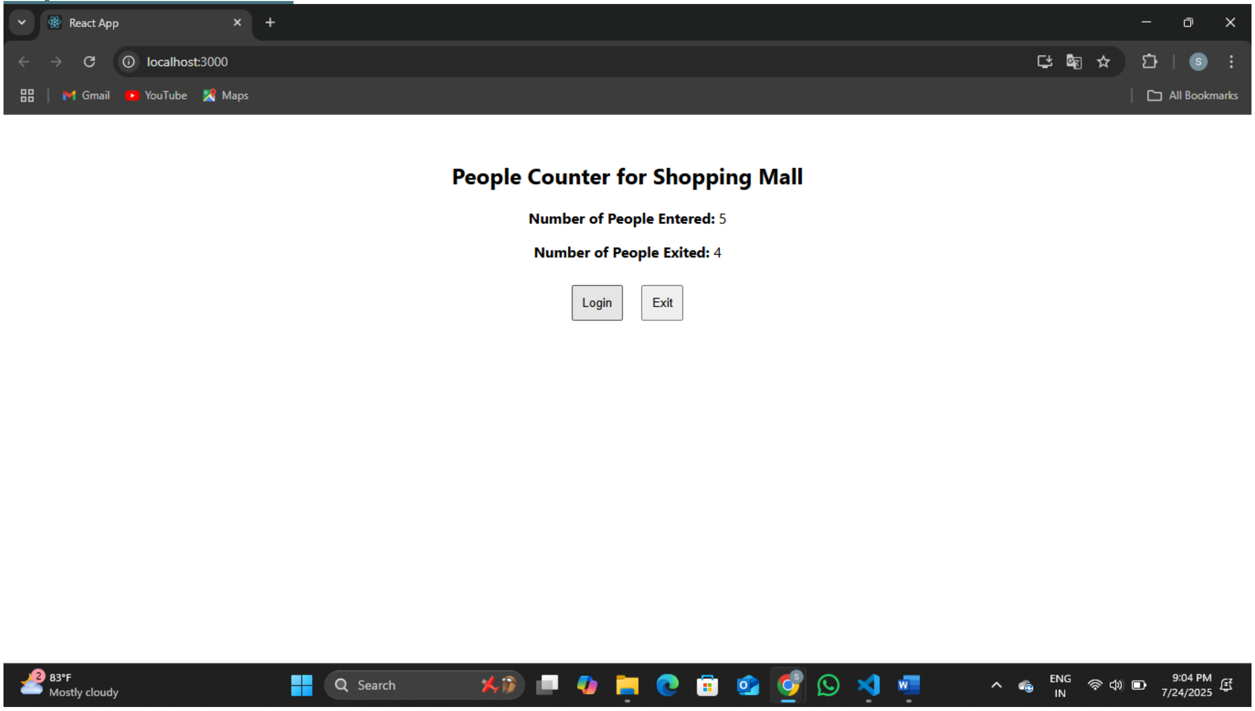
```
import React from 'react';
import './App.css';
import CountPeople from './CountPeople';

function App() {
  return (
    <div className="App">
      <CountPeople />
    </div>
  );
}

export default App;
```

Output:



```
You can now view counterapp in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.1.8:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

http://localhost:3000/



People Counter for Shopping Mall

Number of People Entered: 5

Number of People Exited: 4