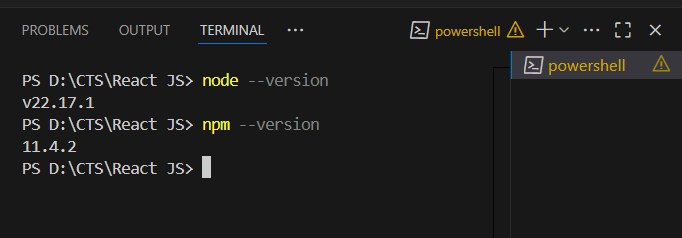
**Exercise 1:**

**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.**

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

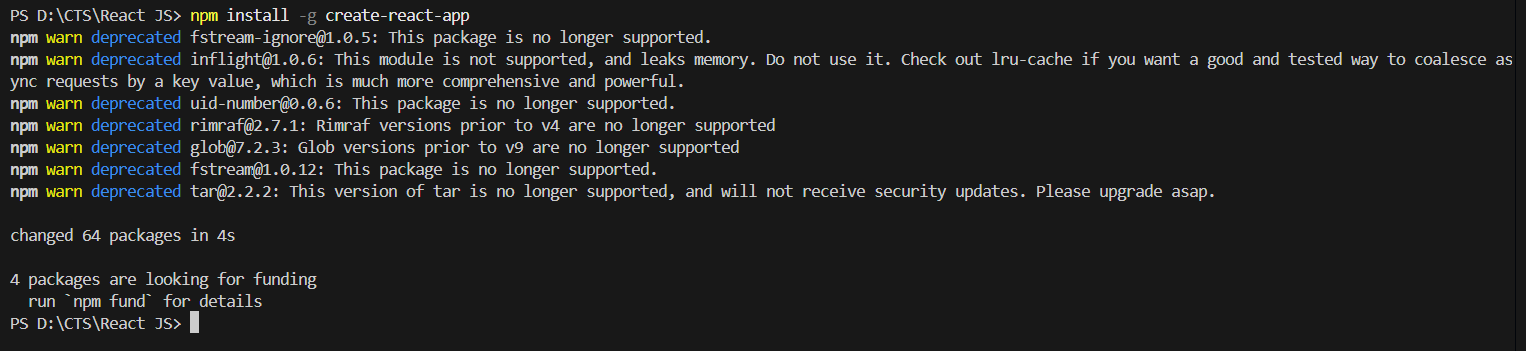
<https://nodejs.org/en/download/>

**Output:**  


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

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

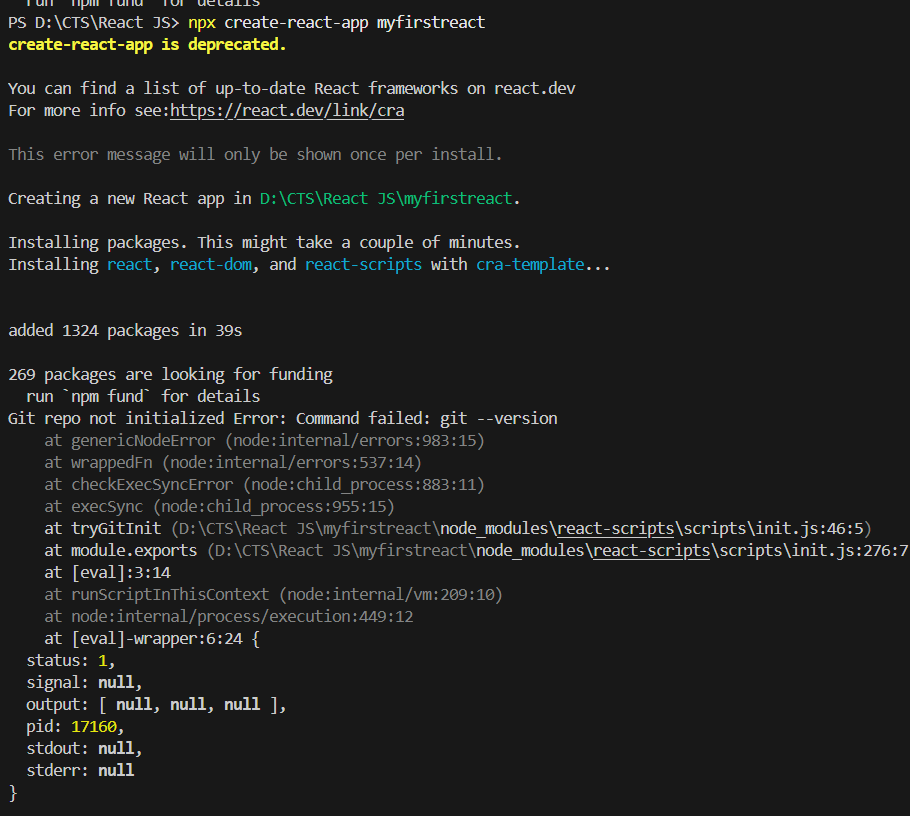
**Output:**

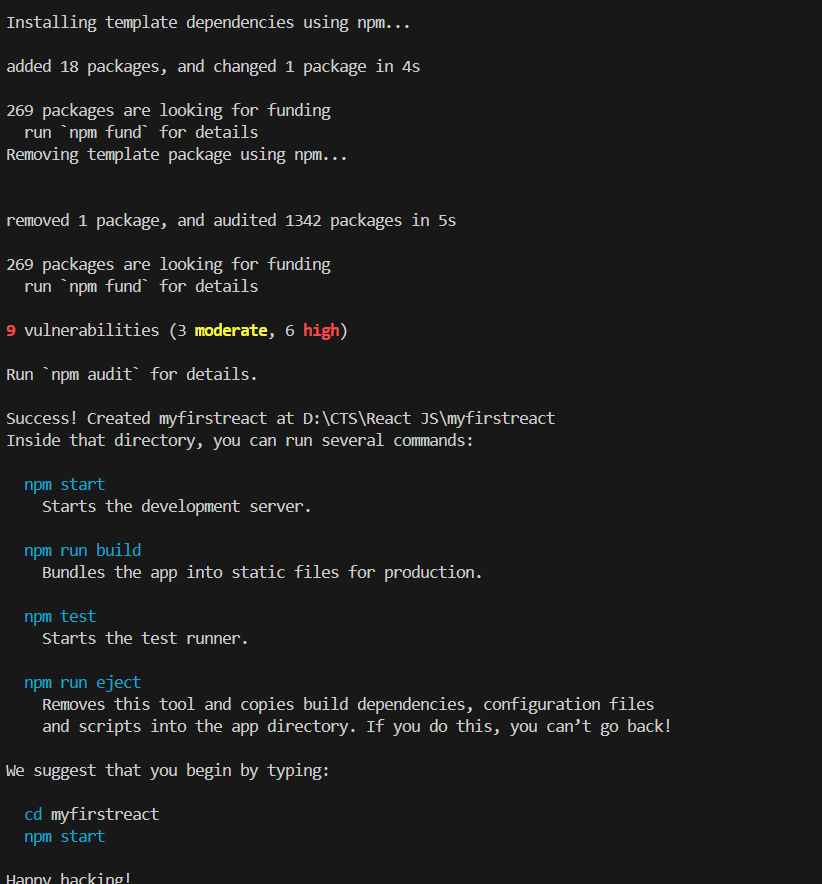


* To create a React Application with the name of “myfirstreact”, type the following command:

**npx create-react-app myfirstreact**

**Output:**



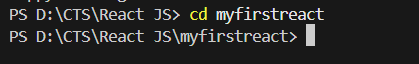


* Once the App is created, navigate into the folder of myfirstreact by typing the following command:

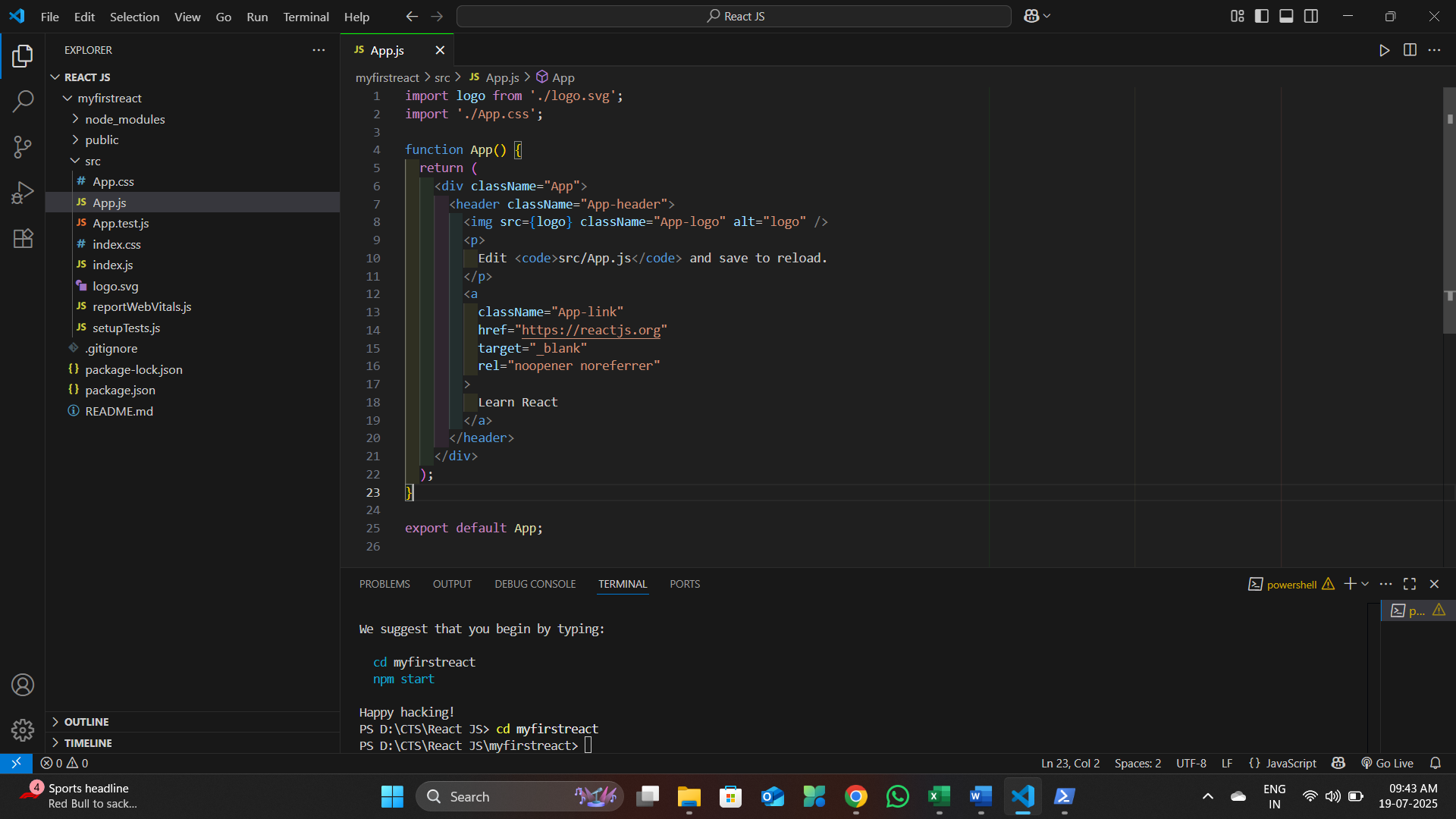
**cd myfirstreact**

**npm start**

**Output:**



* Open the folder of myfirstreact in Visual Studio Code

**Output:**  


* Open the App.js file in Src Folder of myfirstreact
* Remove the current content of “App.js”
* Replace it with the following:

**Code :**

import logo from './logo.svg';

import './App.css';

function App() {

  return (

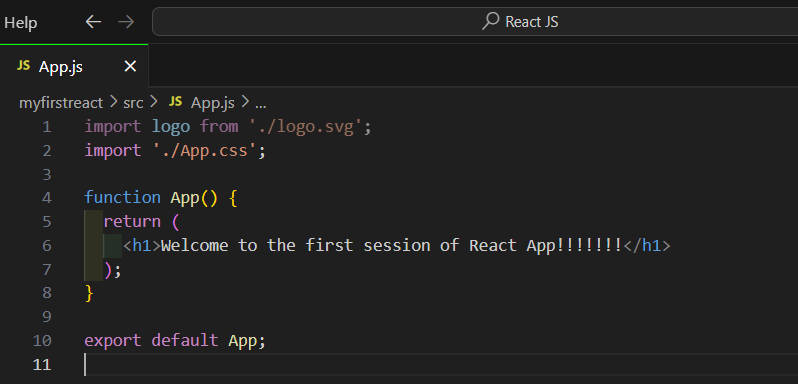
    <h1>Welcome to the first session of React App!!!!!!!</h1>

  );

}

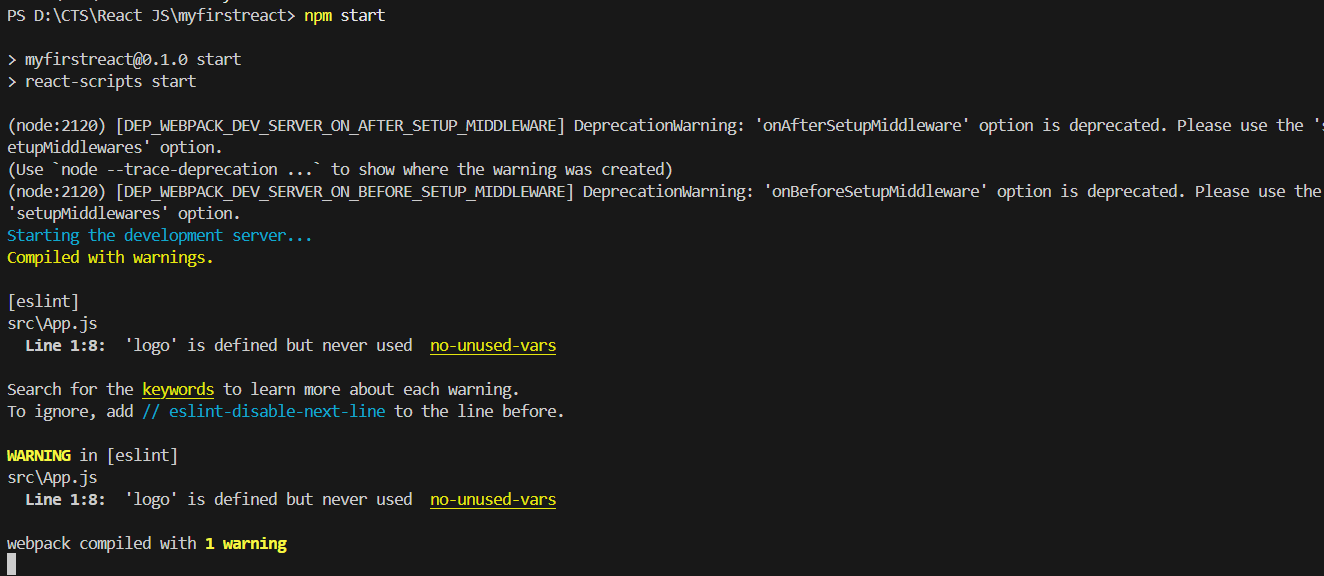
export default App;

**Output:**



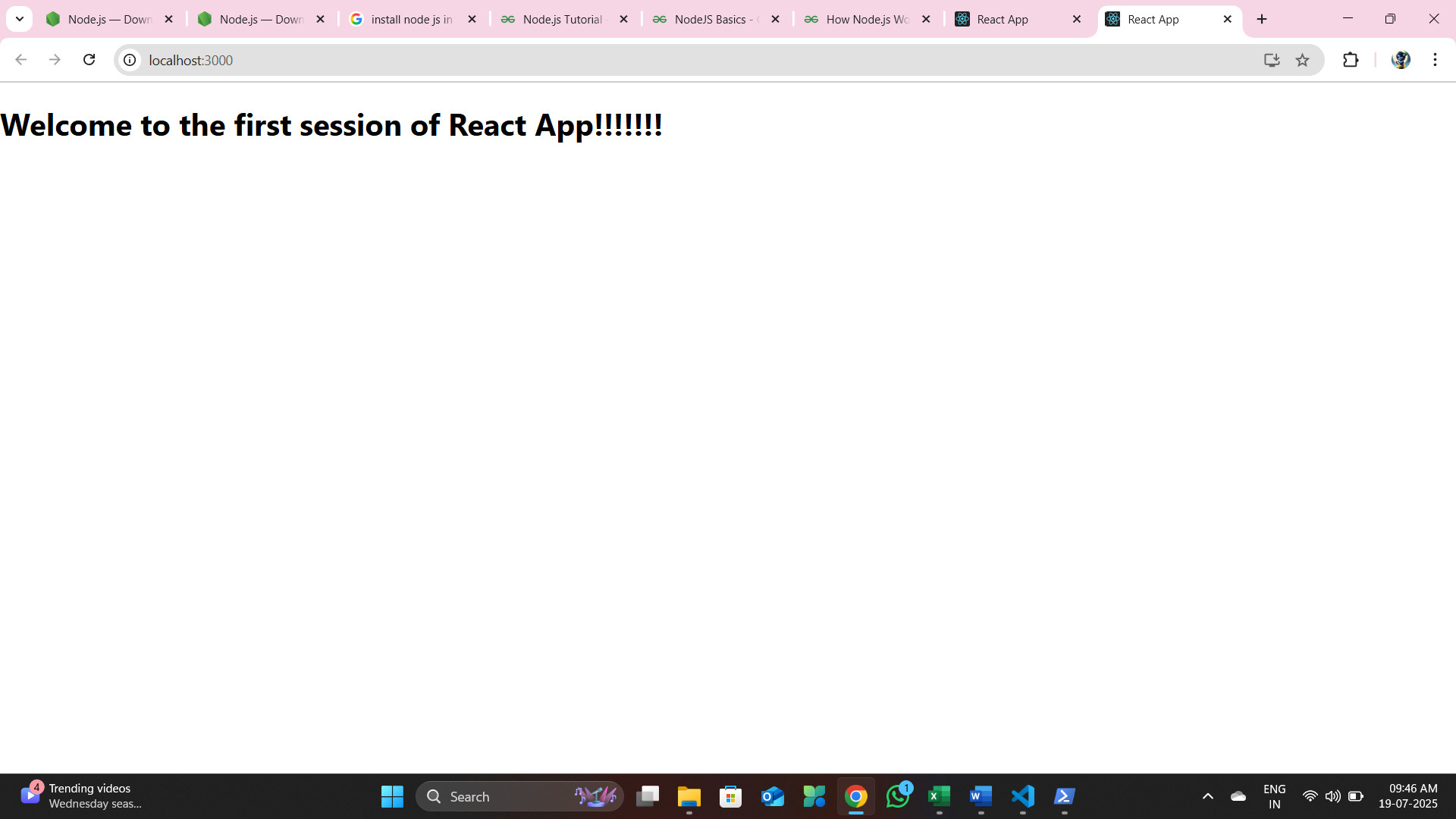
* Run the following command to execute the React application:

**npm start**

**Output:**  


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

**Output:**



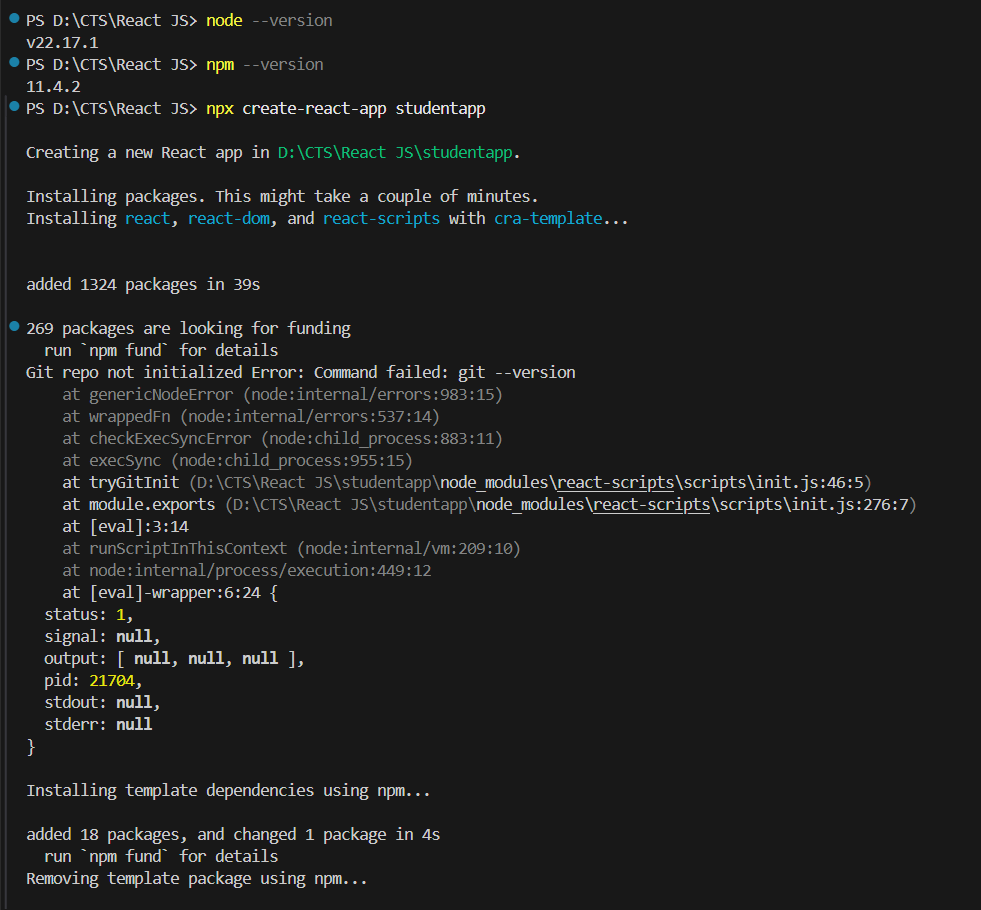
**Exercise 2:**

**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.**

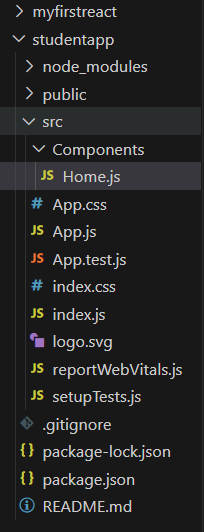
* Create a React project named “StudentApp” type the following command in terminal of Visual studio:



**npx create-react-app studentapp**



* Create a new folder under Src folder with the name “Components”. Add a new file named “Home.js”



* Type the following code in Home.js

**D:\CTS\React JS\studentapp\src\Components\Home.js**

import React, { Component } from "react";

class Home extends Component {

    render() {

        return (

            <div>

                <h3>Welcome to the Home Page of Student Management Portal</h3>

            </div>

        );

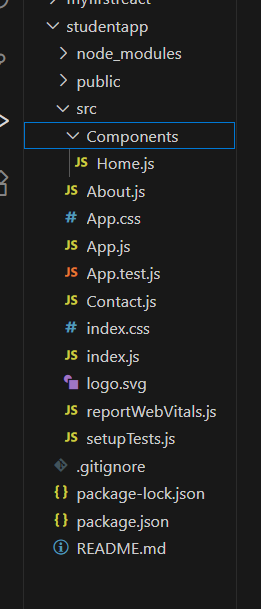
    }

}

export default Home;



* Under Src folder add another file named “About.js”



* 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.

**D:\CTS\React JS\studentapp\src\About.js**

import React, { Component } from 'react';

class About extends Component {

    render() {

        return (

            <center>

                <div>

                    <h3>Welcome to the About Page of Student Management Portal</h3>

                </div>

            </center>

        );

    }

}

export default About;

**D:\CTS\React JS\studentapp\src\Contact.js**

import React, { Component } from 'react';

class Contact extends Component {

    render() {

        return (

            <center>

                <div>

                    <h3>Welcome to the Contact Page of Student Management Portal</h3>

                </div>

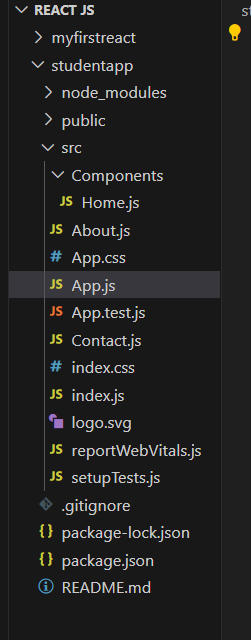
            </center>

        );

    }

}

export default Contact;



* Edit the App.js to invoke the Home, About and Contact component as follows:

**D:\CTS\React JS\studentapp\src\App.js**

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>

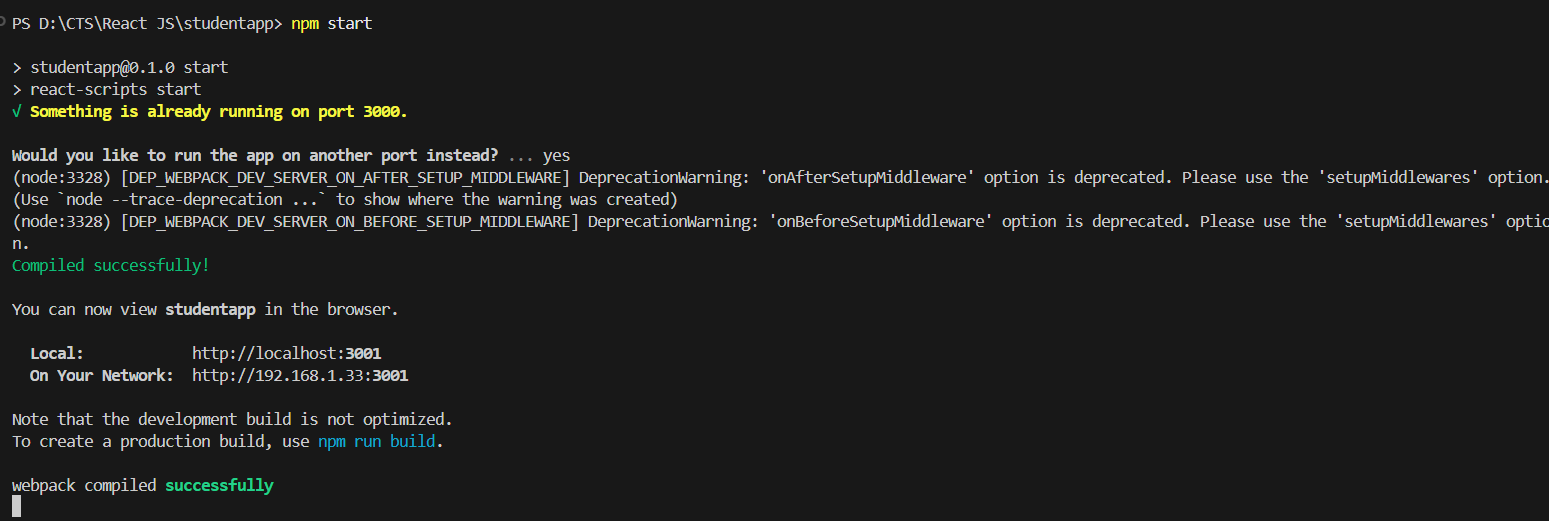
  );

}

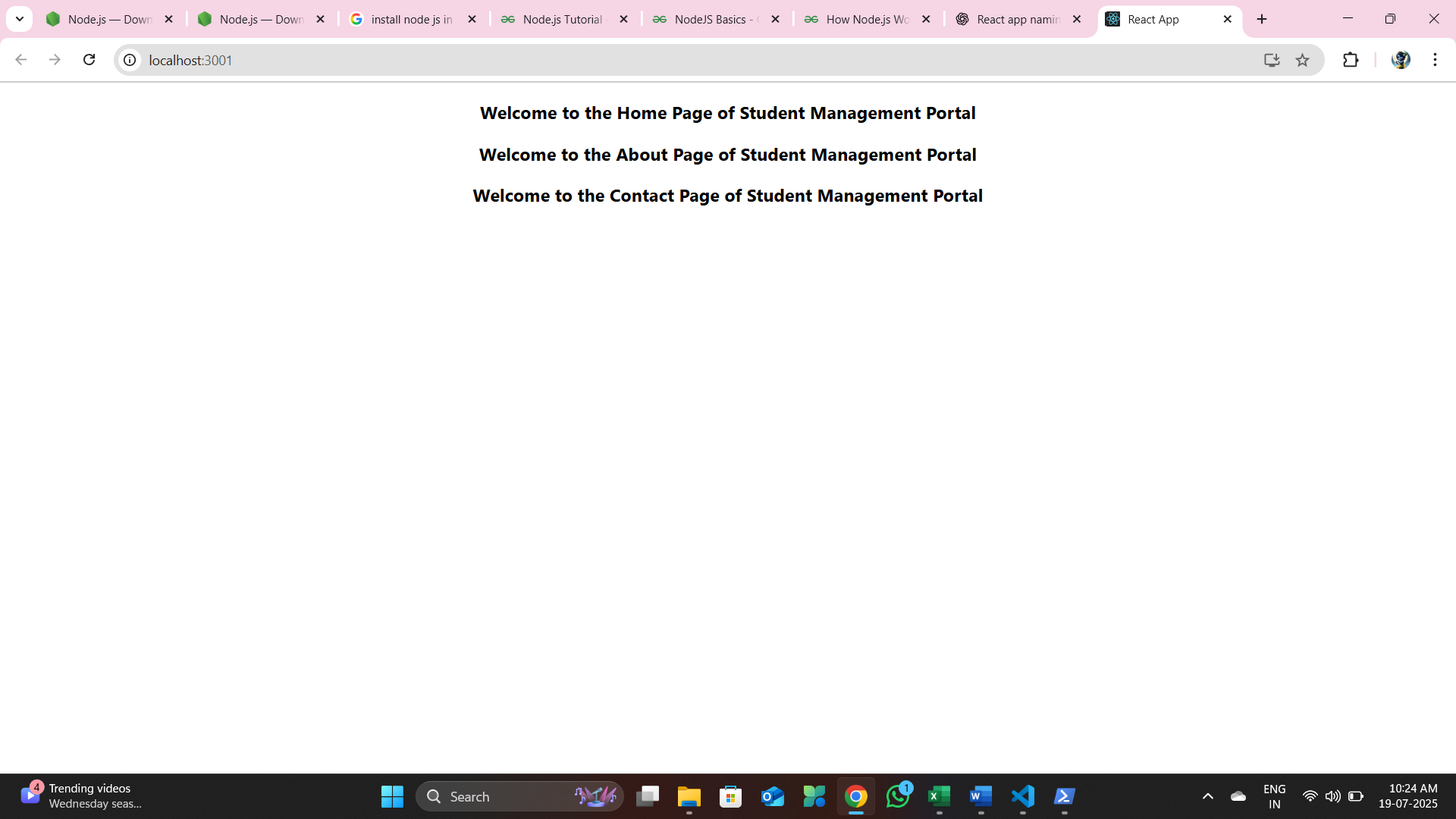
export default App;

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

**npm start**



* Open browser and type “localhost:3001” in the address bar:

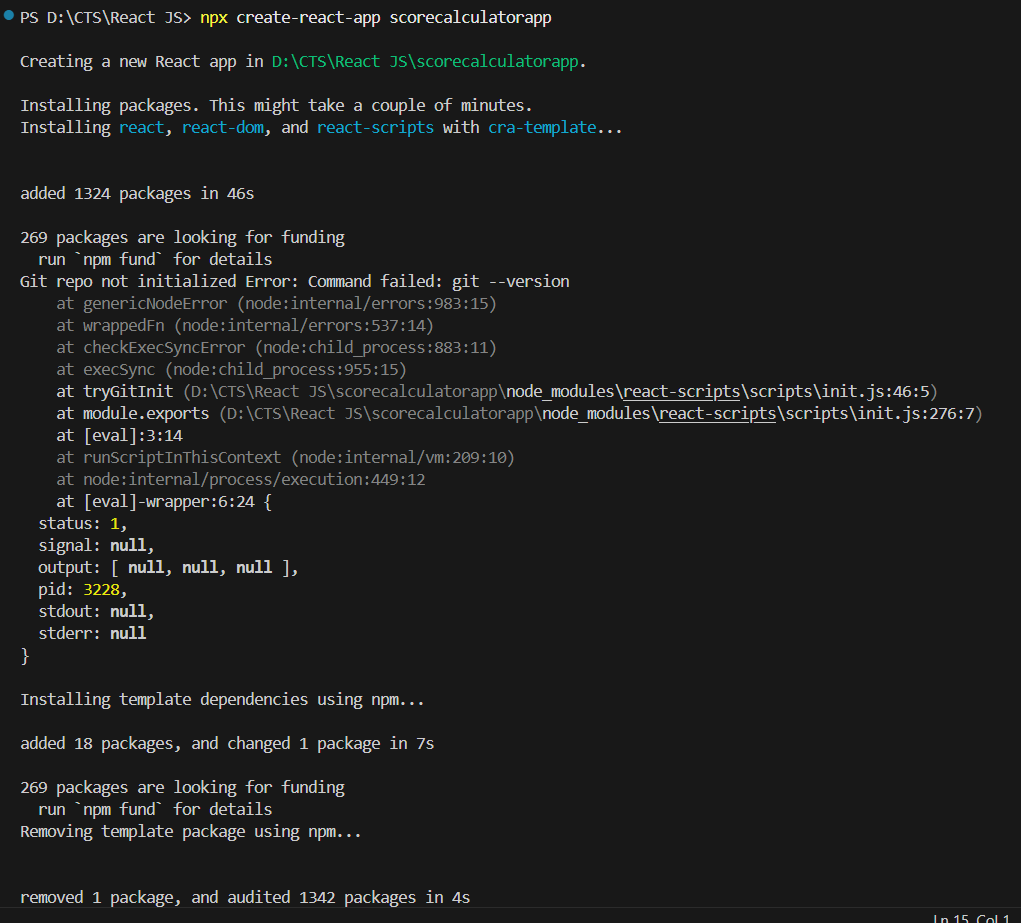


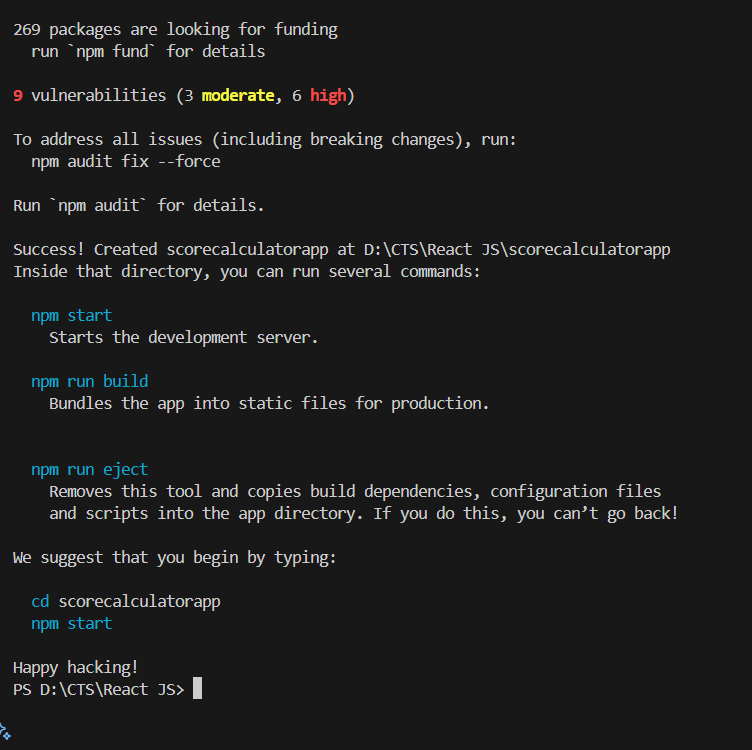
**Exercise 3:**

**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.**

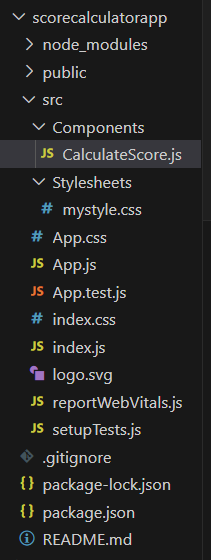
* Create a React project named “scorecalculatorapp” type the following command in terminal of Visual studio:

**npx create-react-app scorecalculatorapp**





* Create a new folder under Src folder with the name “Components”. Add a new file named “CalculateScore.js”



* Type the following code in CalculateScore.js

**D:\CTS\React JS\scorecalculatorapp\src\Components\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 }) => {

    return (

        <div className="formatestyle">

            <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>

            </div>

            <div className="Score">

                <b> Score: </b>

                <span>{calcScore(total, goal)}</span>

            </div>

        </div>

    );

}

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

**D:\CTS\React JS\scorecalculatorapp\src\Stylesheets\mystyle.css**

 .Name {

     font-weight: 300;

     color: blue;

 }

 .School {

     color: crimson;

 }

 .Total {

     color: darkmagenta;

 }

 .formatstyle {

     text-align: center;

     font-size: large;

 }

 .Score {

     color: forestgreen;

 }

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

**D:\CTS\React JS\scorecalculatorapp\src\App.js**

import logo from './logo.svg';

import './App.css';

import { CalculateScore } from './Components/CalculateScore';

function App() {

  return (

    <center>

      <div>

        <CalculateScore

          Name={"Steeve"}

          School={"DNV Public School"}

          total={284}

          goal={3}

        />

      </div>

    </center>

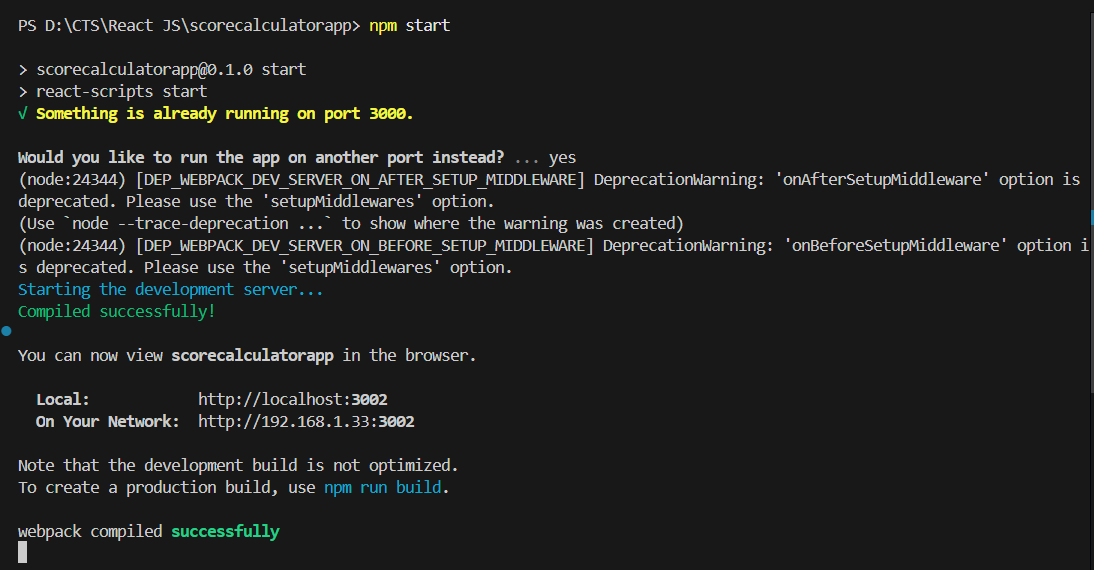
  );

}

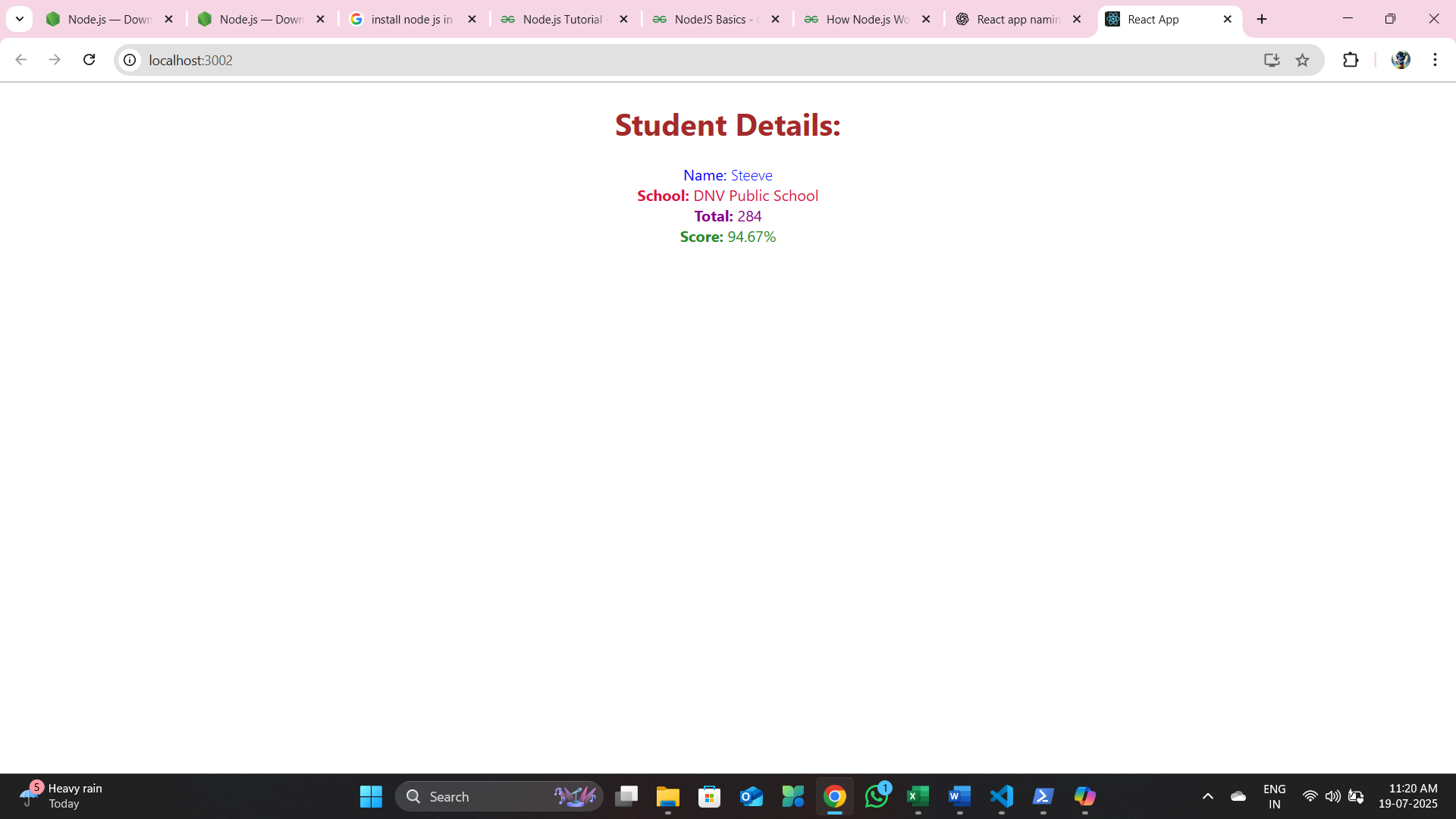
export default App;

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

**npm start**



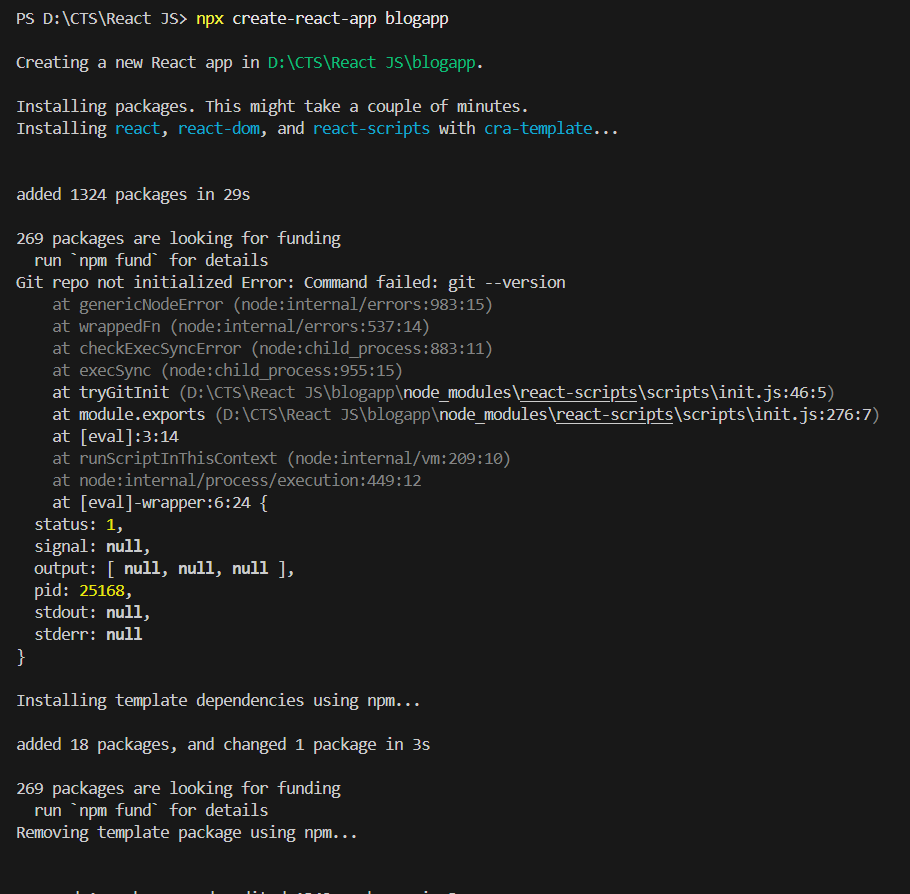
* Open browser and type “localhost:3002” in the address bar:

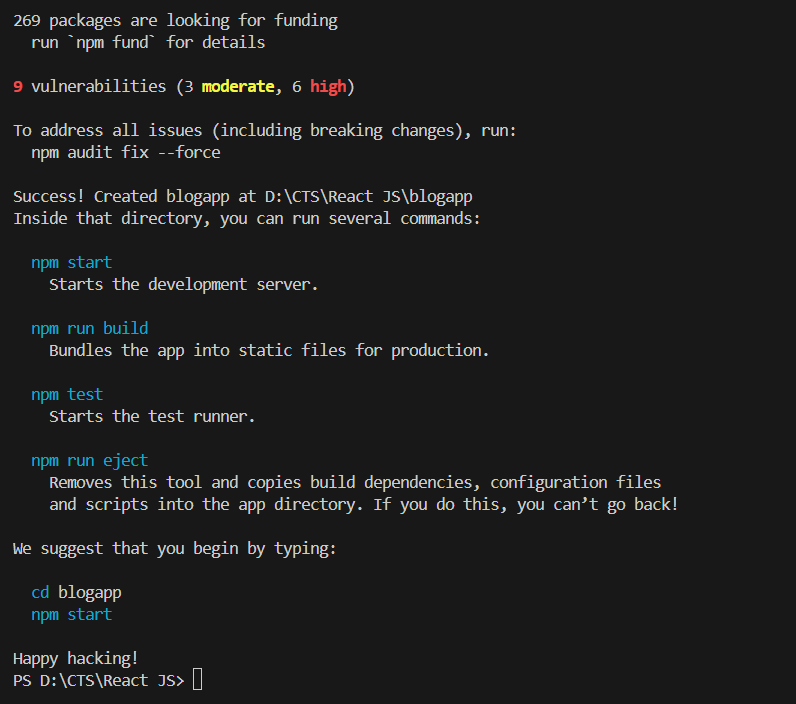


**Exercise 4:**

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

**npx create-react-app blogapp**

****

****

* Open the application using VS Code
* Create a new file named as **Post.js** in **src folder** with following properties

**D:\CTS\React JS\blogapp\src\Post.js**

class Post {

    constructor(title, content) {

        this.id = this.id;

        this.title = title;

        this.body = this.body;

    }

}

* Create a new class based component named as **Posts** inside **Posts.js** file
* Initialize the component with a list of Post in state of the component using the constructor
* 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)
* Implement the **componentDidMount()** hook to make calls to **loadPosts()** which will fetch the posts
* Implement the **render()** which will display the title and post of posts in html page using heading and paragraphs respectively.
* Define a **componentDidCatch()** method which will be responsible for displaying any error happing in the component as alert messages.

**D:\CTS\React JS\blogapp\src\Posts.js**

class Posts extends React.Component {

    constructor(props) {

        super(props);

    }

    loadPosts() {

        fetch('https://jsonplaceholder. import React from 'react';

class Posts extends React.Component {

    constructor(props) {

        super(props);

        this.state = {

            posts: []

        };

    }

    loadPosts() {

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

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

            .then(data => {

                this.setState({ posts: data });

            })

            .catch(error => {

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

            });

    }

    componentDidMount() {

        this.loadPosts();

    }

    render() {

        return (

            <div>

                <h1>Posts</h1>

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

                    <div key={post.id}>

                        <h2>{post.title}</h2>

                        <p>{post.body}</p>

                    </div>

                ))}

            </div>

        );

    }

    componentDidCatch(error, info) {

        alert('An error occurred: ' + error.toString());

    }

}

export default Posts;

* Add the Posts component to App component.

**D:\CTS\React JS\blogapp\src\App.js**

import React from 'react';

import Posts from './Posts';

function App() {

  return (

    <div>

      <Posts />

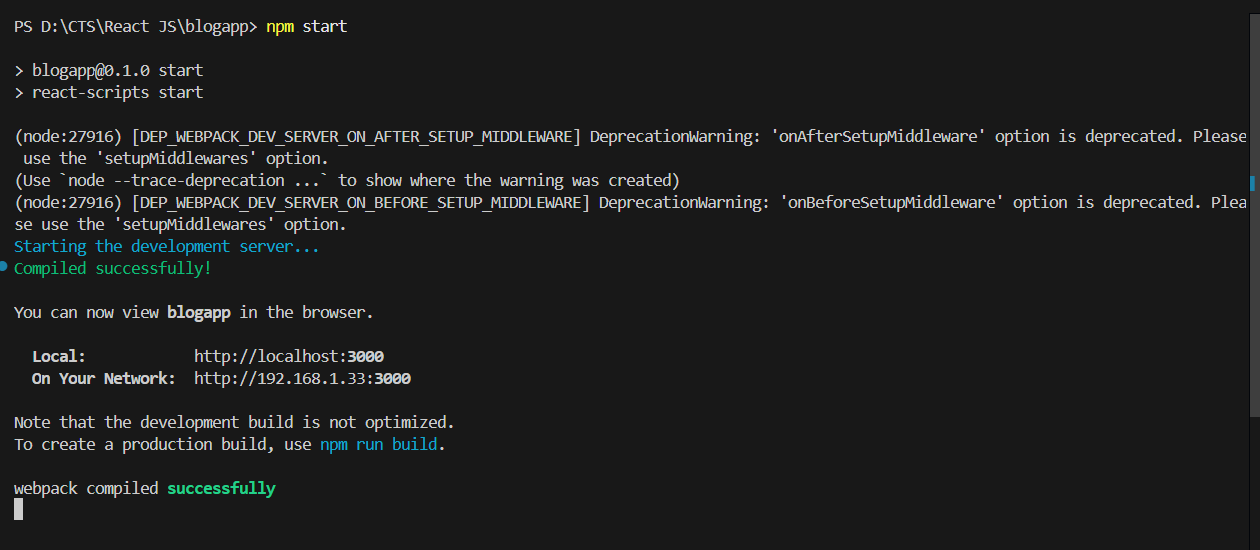
    </div>

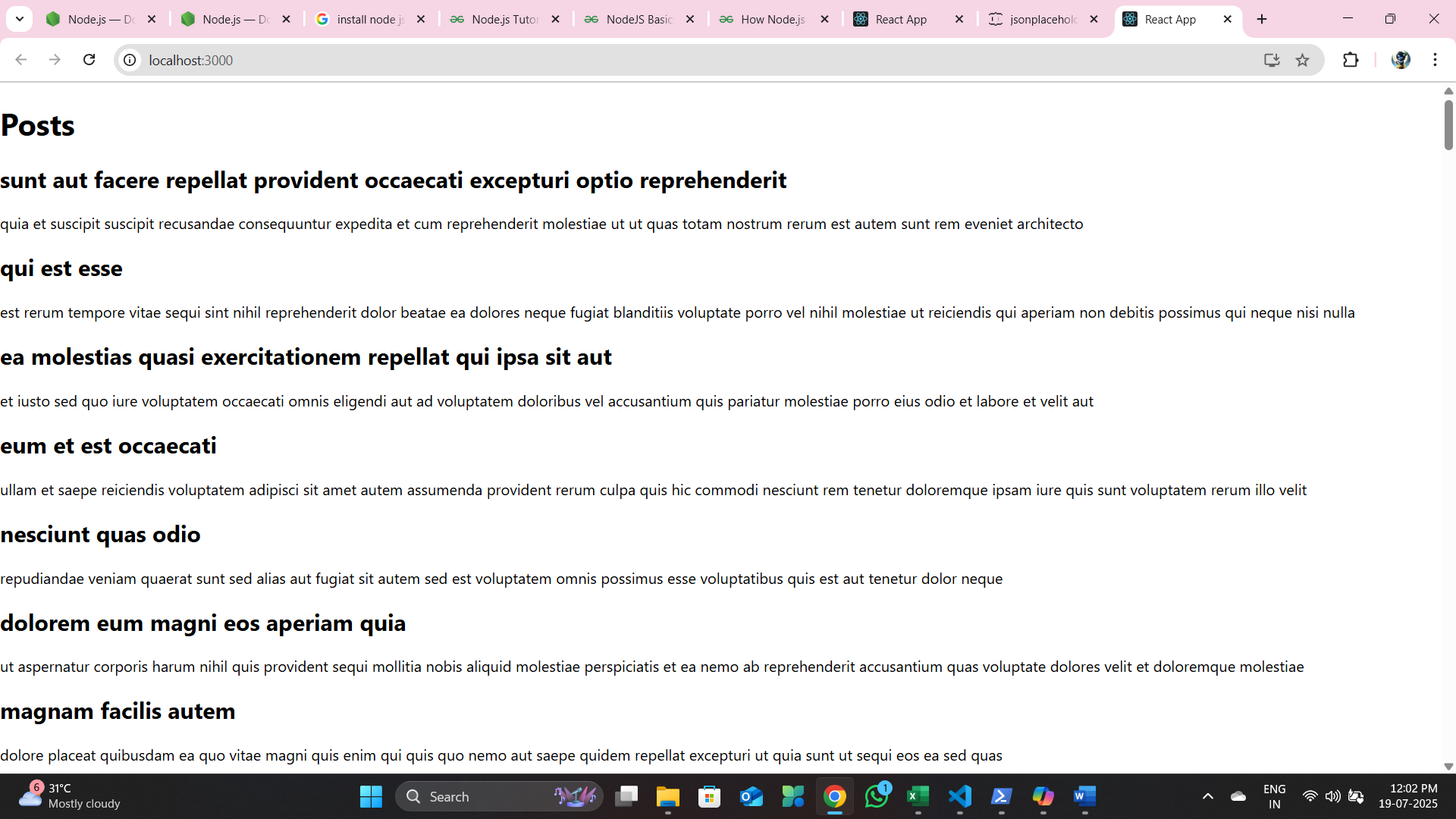
  );

}

export default App;

* Build and Run the application using *npm start* command.

**npm start  
  
**

**Output:**

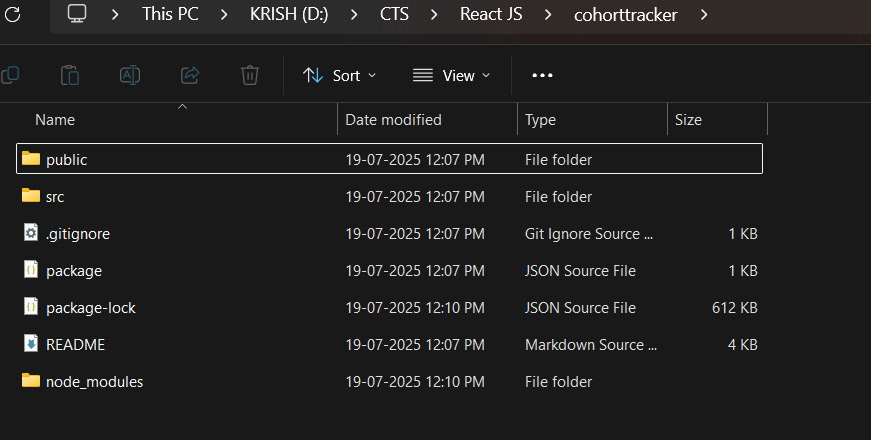
**Exercise 5:**

**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.



* Unzip the react application in a folder



* Open command prompt and switch to the react application folder

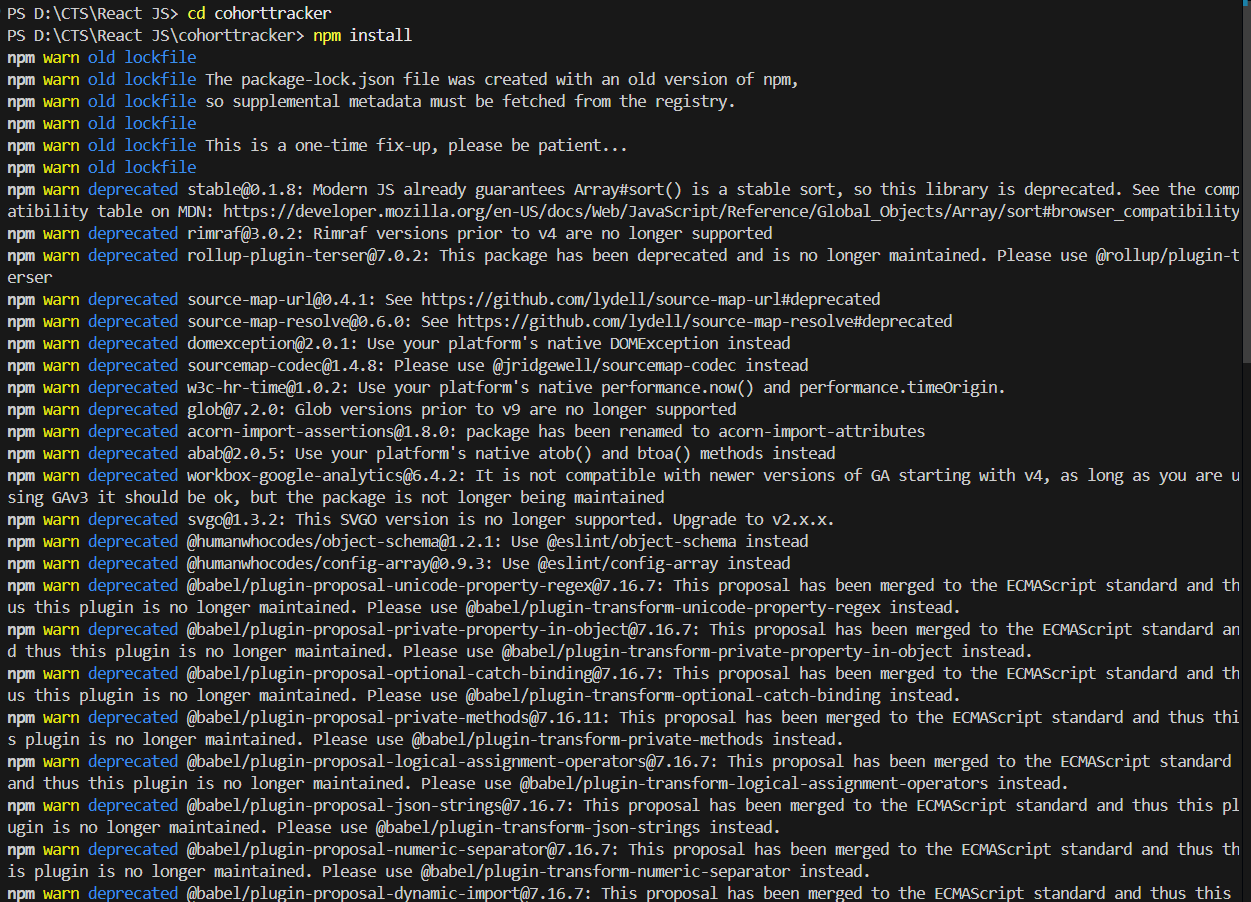
**cd cohorttracker**

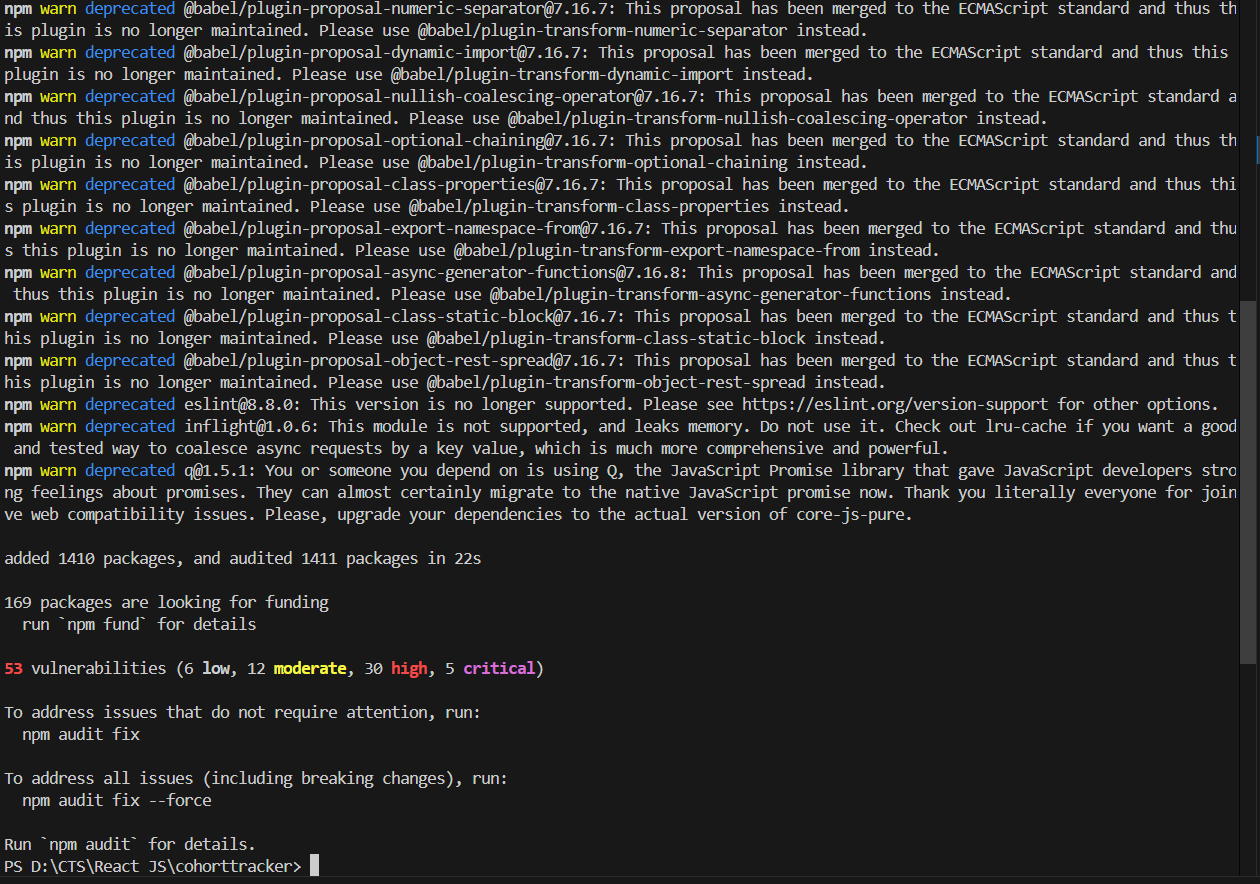
* Restore the node packages using the following commands

**npm install**



Figure 1: Restore packages





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

* Define a css style for html <dt> element using tag selector. Set the font weight to 500.

**D:\CTS\React JS\cohorttracker\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: bold;

}

* Open the cohort details component and import the CSS Module
* Apply the box class to the container div
* Define the style for <h3> element to use “green” color font when cohort status is “ongoing” and “blue” color in all other scenarios.

**D:\CTS\React JS\cohorttracker\src\CohortDetails.js**

import React from 'react';

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

function CohortDetails(props) {

    return (

        <div className={styles.box}>

            <h3

                className={styles.cohortTitle}

                style={{ color: props.cohort.currentStatus === 'Ongoing' ? 'green' : 'blue' }}

            >

                {props.cohort.cohortCode} -

                <span>{props.cohort.technology}</span>

            </h3>

            <dl>

                <dt>Started On</dt>

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

                <dt>Current Status</dt>

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

                <dt>Coach</dt>

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

                <dt>Trainer</dt>

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

            </dl>

        </div>

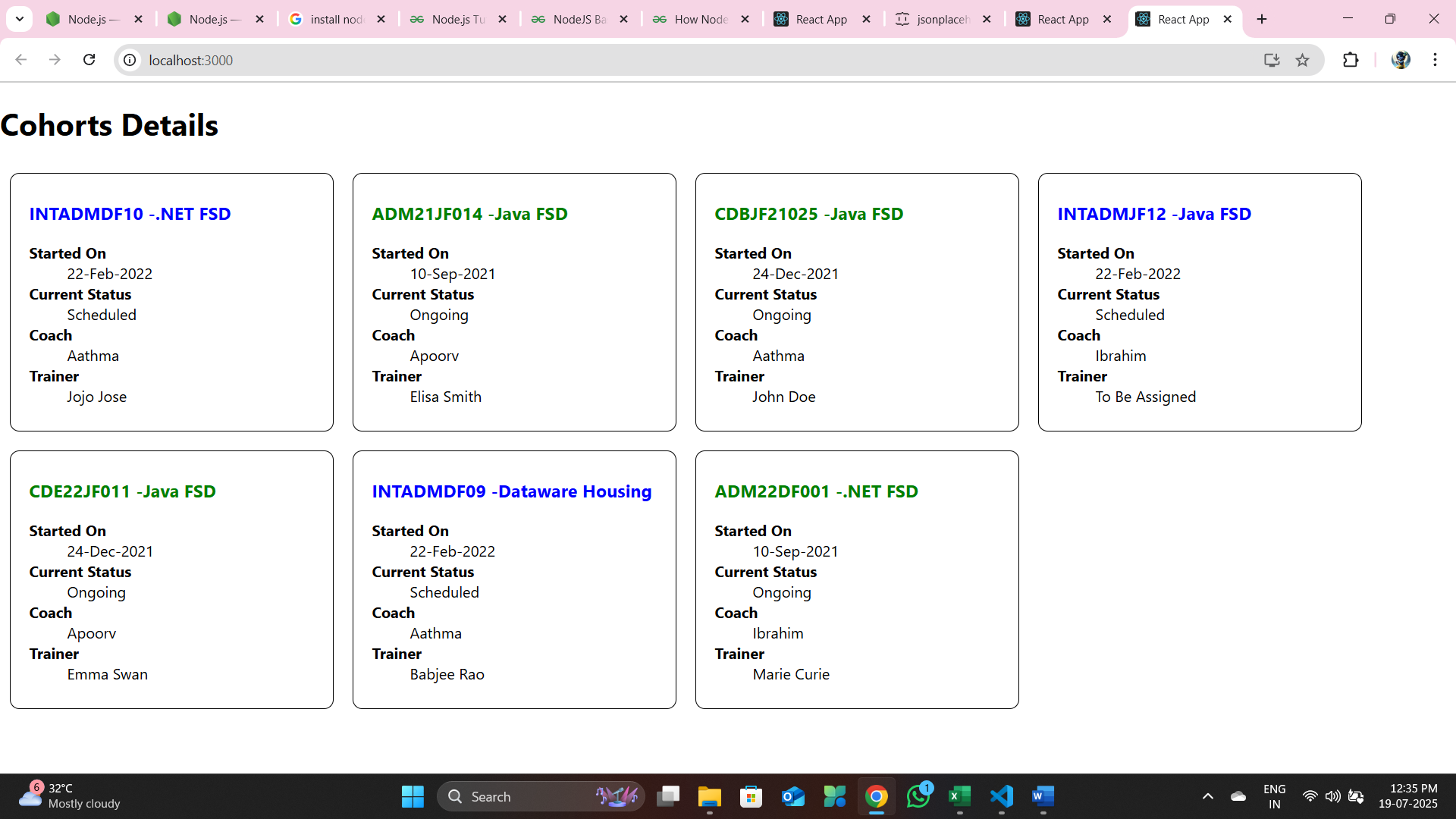
    );

}

export default CohortDetails;

Figure 2: Final Result

**Output:**

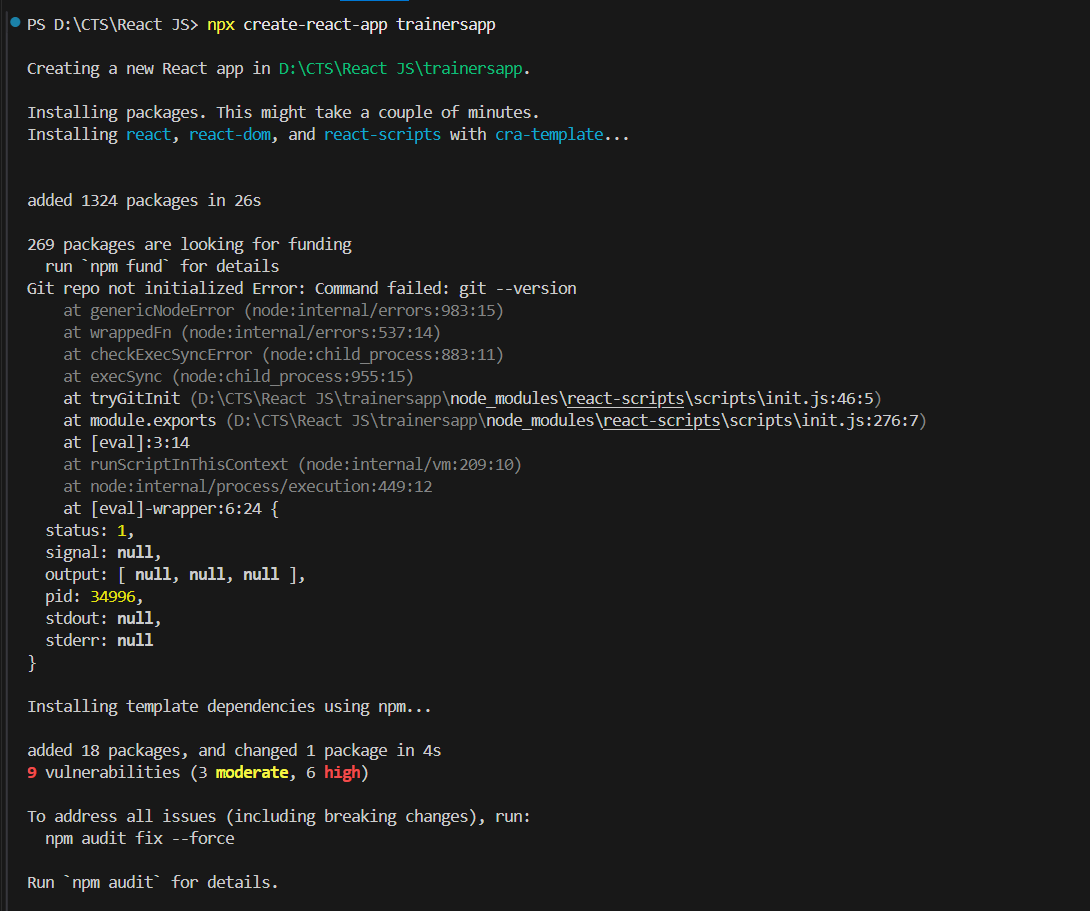


**Exercise 6:**

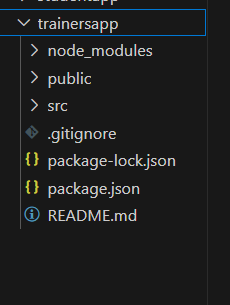
**Cognizant Academy teams want to maintain a list of trainers along with their expertise in a SPA using React as the technology. You are assigned the task of creating this React app.**

**The following trainers’ data application will deal.**

* **T-ID**
* **Name**
* **Phone**
* **Email**
* **Stream**
* **Skills**
* Create a new React app using *create-react-app* tool with the as “TrainersApp”  
  **npx create-react-app trainersapp**



* Open the application using the VS Code



* Add a new file called *trainer.js* inside the **src folder** and define a class named as “Trainer” with the following properties
  + TrainerId
  + Name
  + Email
  + Phone
  + Technology
  + Skills

**D:\CTS\React JS\trainersapp\src\trainer.js**

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;

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

**D:\CTS\React JS\trainersapp\src\TrainersMock.js**

import Trainer from "./trainer";

const trainersMock = [

    new Trainer('t-syed8',

        'Syed Khaleelullah',

        'khaleelullah@cognizant.com',

        '98767561962',

        '.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',

        '987112235',

        'Python',

        ['Python', 'Django', 'Angular'])

];

export default trainersMock;

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

**cd trainersapp**

**npm install react-router-dom@6**

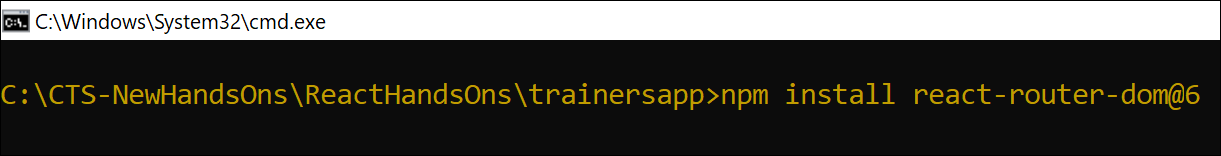


Figure 4: Install React Router

* 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.



Figure 5: TrainersList Component

**D:\CTS\React JS\trainersapp\src\Trainerlist.js**

import React from 'react';

function TrainersList({ trainers }) {

    return (

        <div>

            <h2>Trainers List</h2>

            <ul>

                {trainers.map(trainer => (

                    <li key={trainer.trainerid}>

                        <a href={`mailto:${trainer.email}`} style={{ color: 'blue', textDecoration: 'underline', cursor: 'pointer' }}>

                            {trainer.name}

                        </a>

                    </li>

                ))}

            </ul>

        </div>

    );

}

export default TrainersList;

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



Figure 6: Home Component

import React from 'react';

function Home() {

    return (

        <div>

            <h1>Welcome to Trainers Portal</h1>

        </div>

    );

}

export default Home;

* 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

* / - must redirect to home component
* /trainers – must redirect to trainers list component.

The layout of the page must be similar to the following



Figure 7: App Component

**D:\CTS\React JS\trainersapp\src\App.js**

import React from 'react';

import { BrowserRouter, Routes, Route, Link } from 'react-router-dom';

import Home from './Home';

import TrainersList from './Trainerlist';

import TrainerDetail from './TrainerDetails';

const trainers = [

  {

    trainerid: 1,

    name: 'Syed Khaleelullah',

    email: 'khaleelullah@cognizant.com',

    phone: '97676516962',

    technology: '.NET',

    skills: ['C#', 'SQL Server', 'React', '.NET Core']

  },

  {

    trainerid: 2,

    name: 'Jojo Jose',

    email: 'jojo@email.com',

    phone: '9876543210',

    technology: 'Java',

    skills: ['Java', 'Spring', 'Hibernate']

  },

  {

    trainerid: 3,

    name: 'Elisa Jones',

    email: 'elisa@email.com',

    phone: '9123456780',

    technology: 'Python',

    skills: ['Python', 'Django', 'Flask']

  }

];

function App() {

  return (

    <BrowserRouter>

      <div style={{ padding: '20px' }}>

        <h1 style={{ fontSize: '2.5rem', marginBottom: '10px' }}>My Academy Trainers App</h1>

        <nav style={{ marginBottom: '20px', fontSize: '1.1rem' }}>

          <Link to="/" style={{ marginRight: '10px' }}>Home</Link>

          |

          <Link to="/trainers" style={{ marginLeft: '10px' }}>Show Trainers</Link>

        </nav>

        <Routes>

          <Route path="/" element={<Home />} />

          <Route path="/trainers" element={<TrainersList trainers={trainers} />} />

          <Route path="/trainer/:id" element={<TrainerDetail />} />

        </Routes>

      </div>

    </BrowserRouter>

  );

}

export default App;

* 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.

**D:\CTS\React JS\trainersapp\src\TrainerDetails.js**

import React from 'react';

import { useParams } from 'react-router-dom';

const trainers = [

    {

        trainerid: 1,

        name: 'Syed Khaleelullah',

        email: 'khaleelullah@cognizant.com',

        phone: '97676516962',

        technology: '.NET',

        skills: ['C#', 'SQL Server', 'React', '.NET Core']

    },

    {

        trainerid: 2,

        name: 'Jojo Jose',

        email: 'jojo@email.com',

        phone: '9876543210',

        technology: 'Java',

        skills: ['Java', 'Spring', 'Hibernate']

    },

    {

        trainerid: 3,

        name: 'Elisa Jones',

        email: 'elisa@email.com',

        phone: '9123456780',

        technology: 'Python',

        skills: ['Python', 'Django', 'Flask']

    }

];

function TrainerDetail() {

    const { id } = useParams();

    const trainer = trainers.find(t => t.trainerid === parseInt(id));

    if (!trainer) {

        return <div>Trainer not found.</div>;

    }

    return (

        <div>

            <h2>Trainers Details</h2>

            <h3 style={{ fontWeight: 'bold' }}>

                {trainer.name} ({trainer.technology})

            </h3>

            <div>{trainer.email}</div>

            <div>{trainer.phone}</div>

            <ul>

                {trainer.skills.map(skill => (

                    <li key={skill}>{skill}</li>

                ))}

            </ul>

        </div>

    );

}

export default TrainerDetail;

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

**Output:**

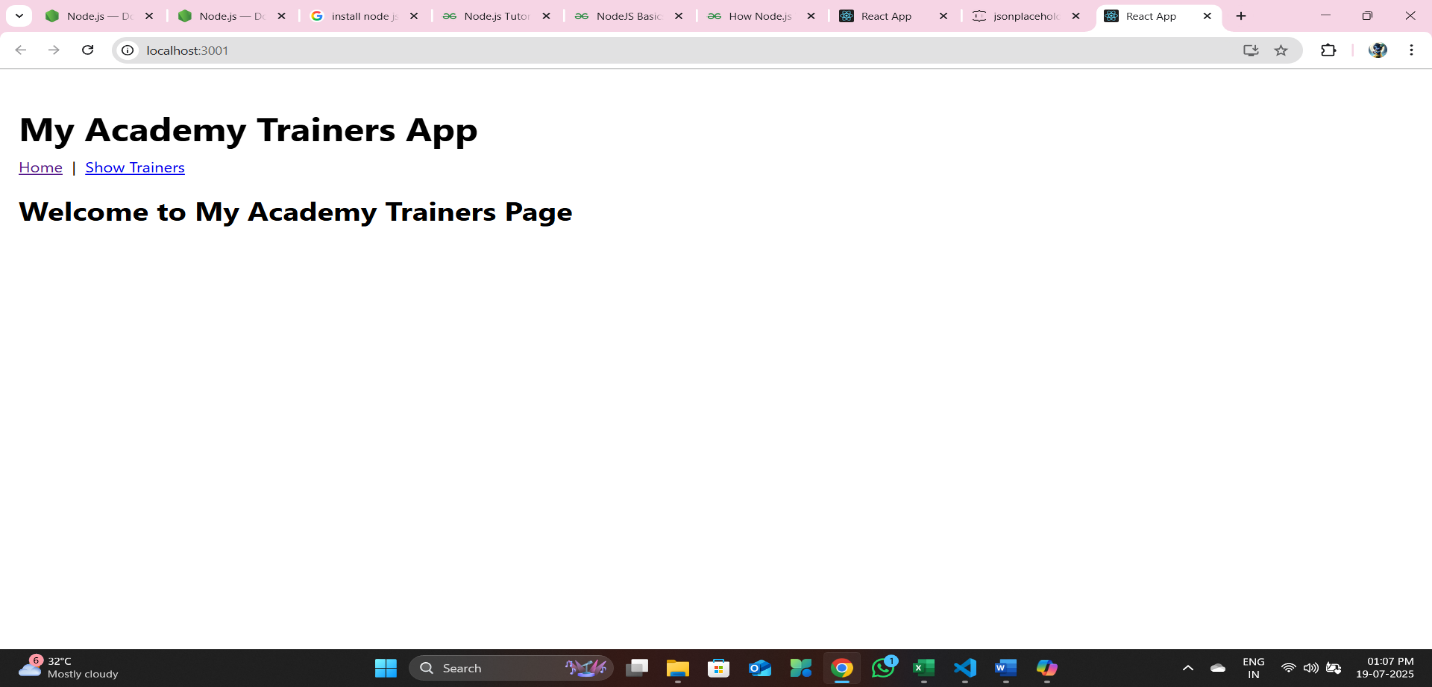


Figure: Home

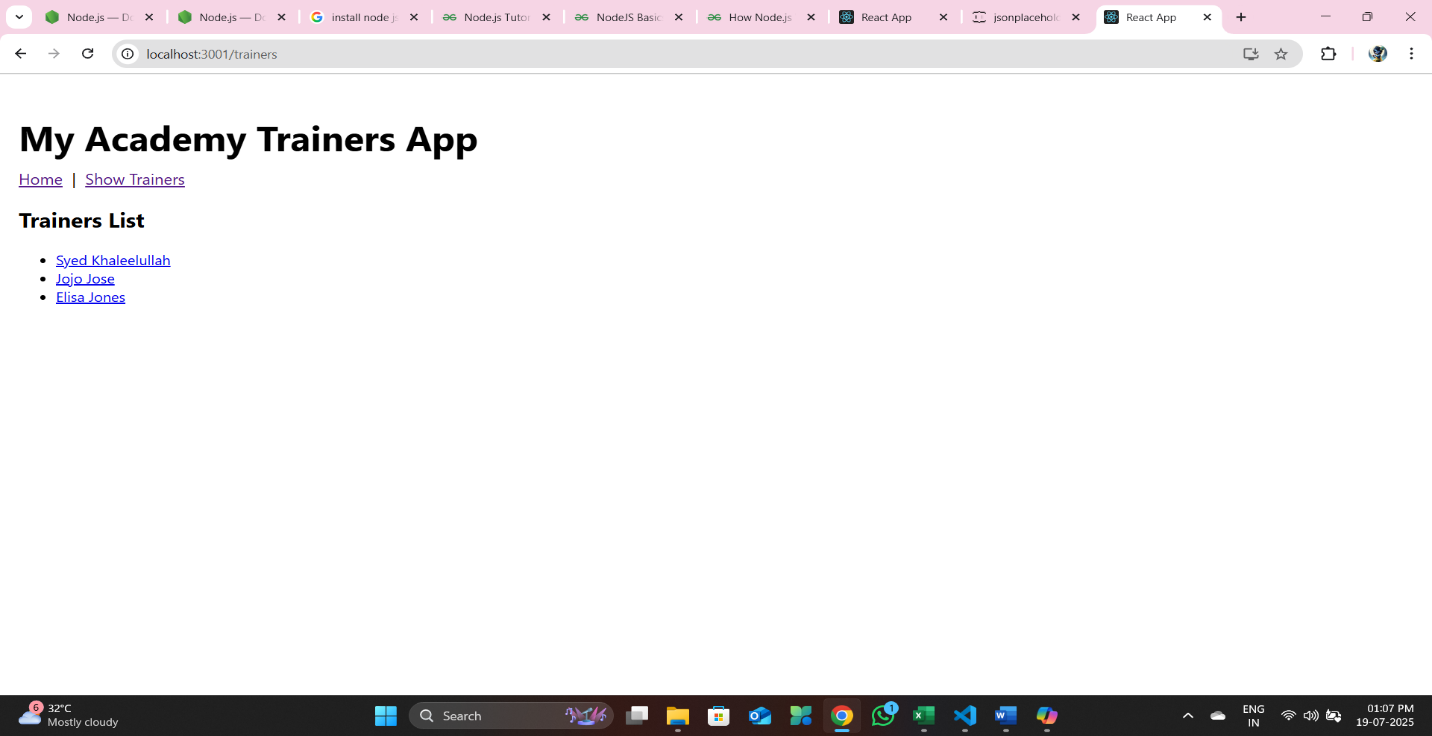


Figure : Trainers List

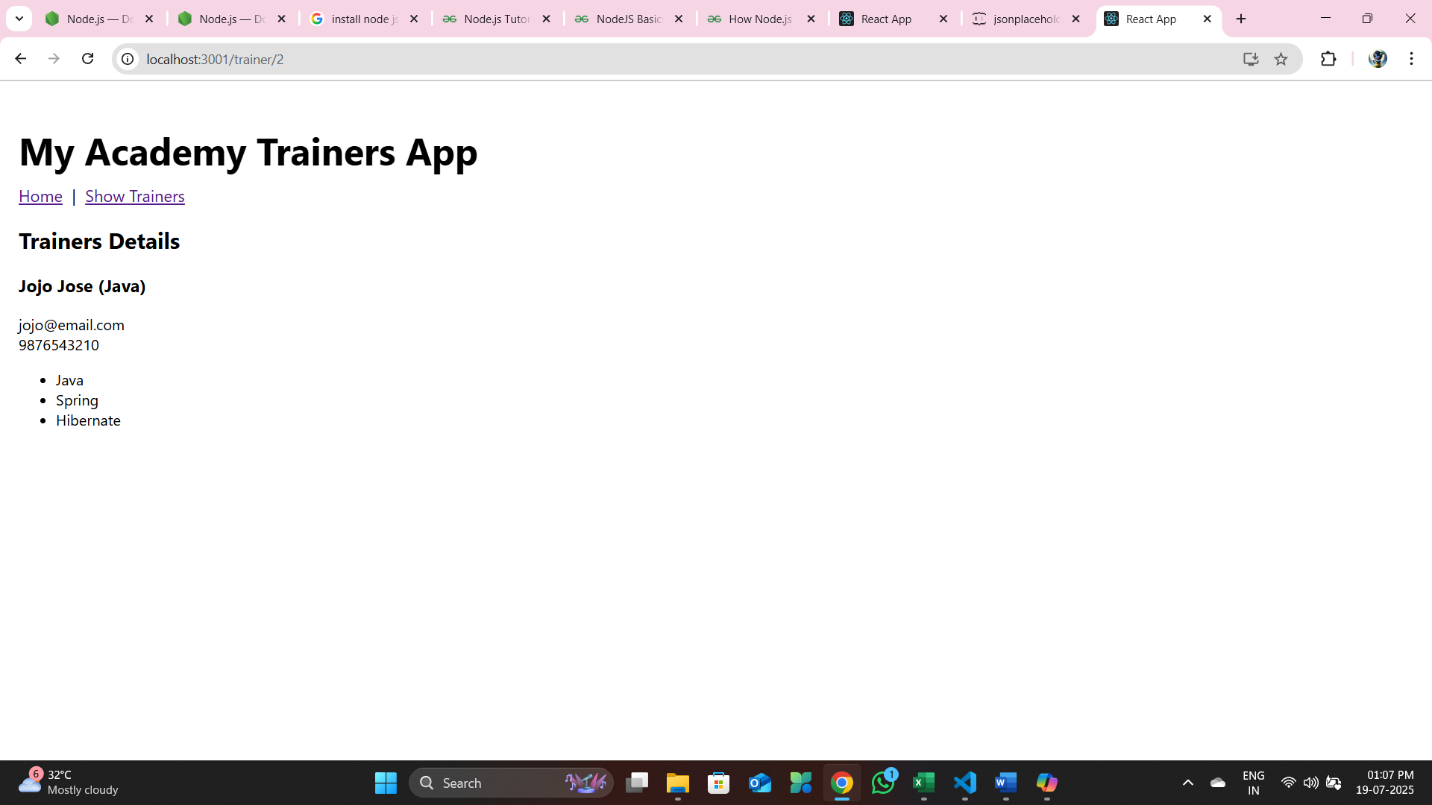
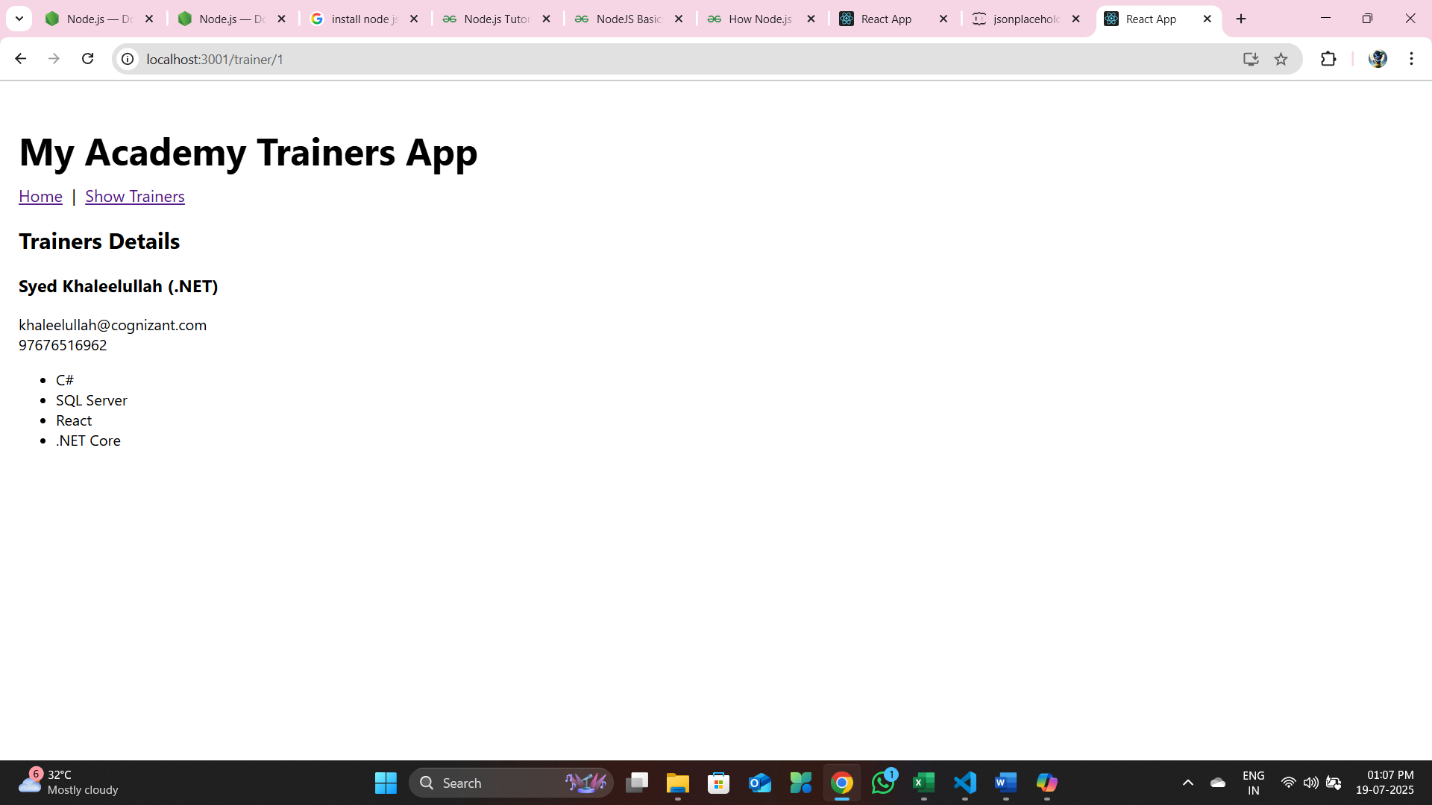


Figure : Trainer Details

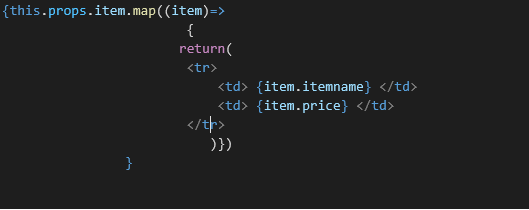
**Exercise 7:**

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

**npx create-react-app shoppingapp**

**cd shoppingapp**

* In Cart class, create 2 properties as mentioned below:
* Itemname
* Price



**D:\CTS\React JS\shoppingapp\src\Cart.js**

import React from 'react';

class Cart extends React.Component {

    render() {

        return (

            <div

                style={{

                    margin: '0 auto',

                    border: '2px solid #888',

                    borderRadius: '10px',

                    padding: '20px',

                    width: '320px',

                    color: 'seagreen',

                    background: '#fff',

                    boxShadow: '0 2px 8px rgba(0,0,0,0.05)'

                }}

            >

                <div style={{ display: 'flex', fontWeight: 'bold', marginBottom: '12px' }}>

                    <div style={{

                        flex: 1,

                        border: '1px solid #888',

                        borderRadius: '6px',

                        padding: '8px',

                        marginRight: '8px',

                        textAlign: 'center',

                        background: '#f9f9f9'

                    }}>Name</div>

                    <div style={{

                        flex: 1,

                        border: '1px solid #888',

                        borderRadius: '6px',

                        padding: '8px',

                        textAlign: 'center',

                        background: '#f9f9f9'

                    }}>Price</div>

                </div>

                {this.props.item.map((item, idx) => (

                    <div key={idx} style={{ display: 'flex', marginBottom: '8px' }}>

                        <div style={{

                            flex: 1,

                            border: '1px solid #888',

                            borderRadius: '6px',

                            padding: '8px',

                            marginRight: '8px',

                            textAlign: 'center',

                            background: '#f9f9f9'

                        }}>

                            {item.itemname}

                        </div>

                        <div style={{

                            flex: 1,

                            border: '1px solid #888',

                            borderRadius: '6px',

                            padding: '8px',

                            textAlign: 'center',

                            background: '#f9f9f9'

                        }}>

                            {item.price}

                        </div>

                    </div>

                ))}

            </div>

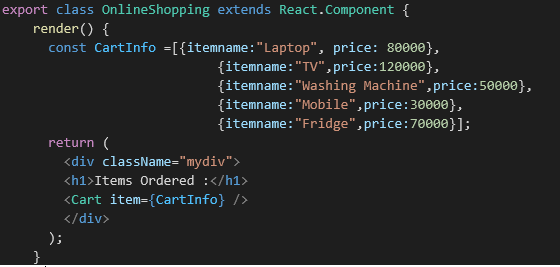
        );

    }

}

export default Cart;

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



**D:\CTS\React JS\shoppingapp\src\OnlineShopping.js**

import React from 'react';

import Cart from './Cart';

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 style={{ color: 'seagreen', textAlign: 'center' }}>Items Ordered :</h1>

                <Cart item={CartInfo} />

            </div>

        );

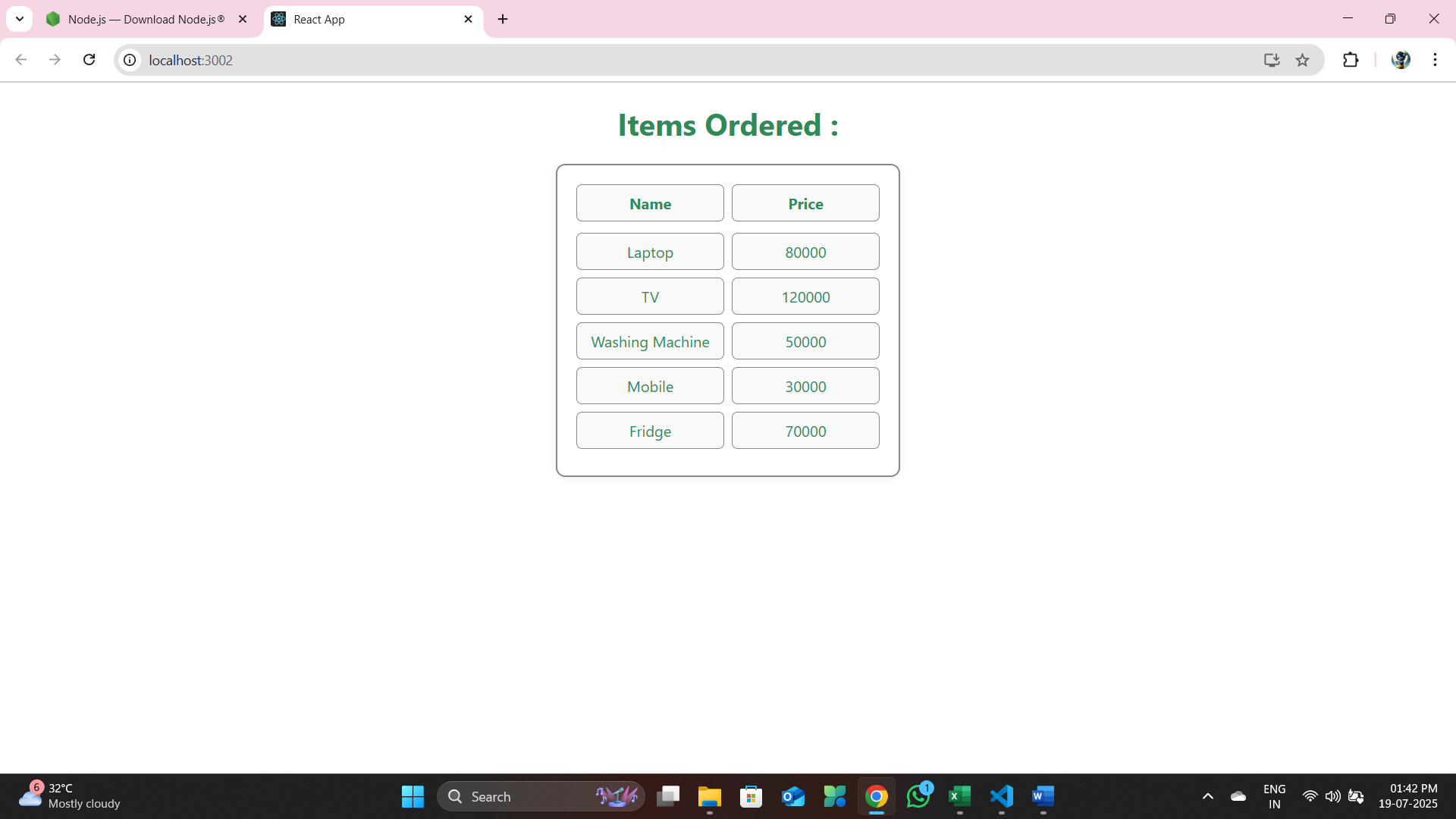
    }

}

export default OnlineShopping;

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

**Output:**



**Exercise 8:**

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

**npx create-react-app counterapp**

**cd counterapp**

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

* Login 🡪 when clicked, the entrycount should get incremented by 1
* Exit 🡪 when clicked, the exitcount should get incremented by 1

**D:\CTS\React JS\counterapp\src\CountPeople.js**

import React, { Component } from 'react';

class CountPeople extends 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 className="counter-container">

                <h2>People Entered : {this.state.entryCount}</h2>

                <button onClick={this.UpdateEntry}>Login</button>

                <h2>People Exited : {this.state.exitCount}</h2>

                <button onClick={this.UpdateExit}>Exit</button>

            </div>

        );

    }

}

export default CountPeople;

**D:\CTS\React JS\counterapp\src\App.js**

import React from 'react';

import CountPeople from './CountPeople';

import './App.css';

function App() {

  return (

    <div className="App">

      <CountPeople />

    </div>

  );

}

export default App;

**D:\CTS\React JS\counterapp\src\App.css**

.App {

  display: flex;

  justify-content: center;

  align-items: center;

  height: 100vh;

}

.counter-container {

  text-align: center;

}

button {

  background-color: green;

  color: white;

  padding: 10px 20px;

  margin: 10px;

  font-size: 16px;

  border: none;

  border-radius: 5px;

  cursor: pointer;

}

button:hover {

  opacity: 0.9;

}

**npm start**

**Output** :

