**WEEK-5**

**REACTJS**

**Superset ID: 6430194**

**Name : Sidharth K**

**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.**

**CODE:**

npm install -g create-react-app

npx create-react-app project

cd project

function App() {

return (

<div>

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

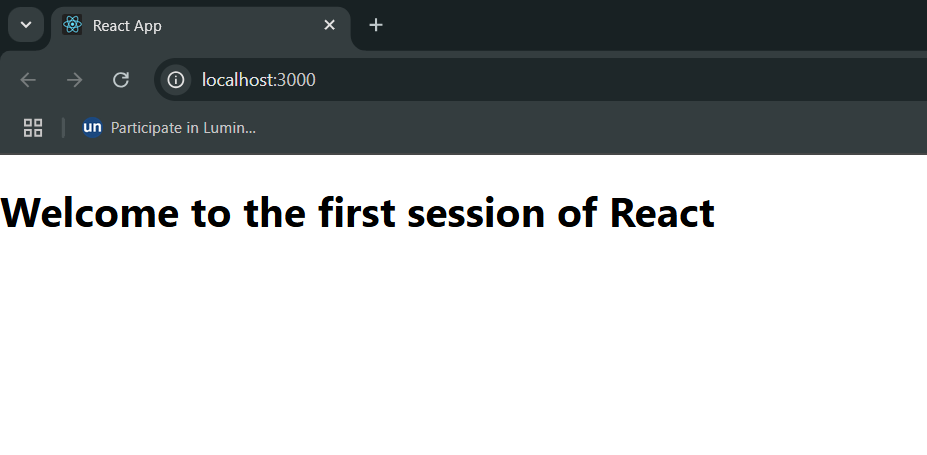
</div>

);

}

export default App;

**OUTPUT:**

****

**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”.**

**CODE:**

**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>

<Home />

<About />

<Contact />

</div>

);

}

export default App;

**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;

**Components.js:**

import React from 'react';

function Contact() {

return (

<div>

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

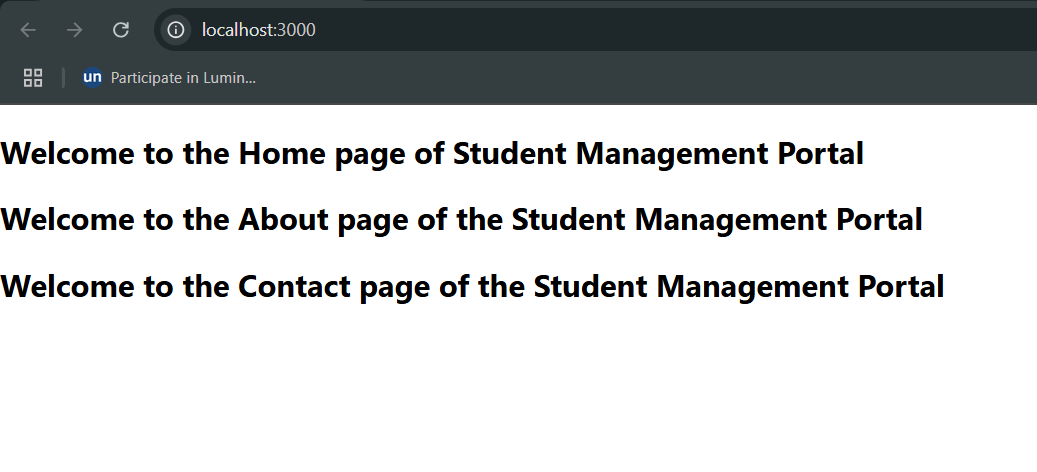
</div>

);

}

export default Contact;

**OUTPUT:**

****

**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.**

**CODE:**

**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 handleSubmit = (e) => {

e.preventDefault();

const avg = (parseFloat(total) / parseFloat(goal)).toFixed(2);

setAverage(avg);

};

return (

<div className="form-container">

<h2>Student Score Calculator</h2>

<form onSubmit={handleSubmit}>

<input type="text" placeholder="Name" onChange={(e) => setName(e.target.value)} required />

<input type="text" placeholder="School" onChange={(e) => setSchool(e.target.value)} required />

<input type="number" placeholder="Total Score" onChange={(e) => setTotal(e.target.value)} required />

<input type="number" placeholder="Goal" onChange={(e) => setGoal(e.target.value)} required />

<button type="submit">Calculate Average</button>

</form>

{average && (

<div className="result">

<p>Name: {name}</p>

<p>School: {school}</p>

<p>Average Score: {average}</p>

</div>

)}

</div>

);

}

export default CalculateScore;

**mystyle.css:**

.form-container {

width: 300px;

margin: 30px auto;

padding: 20px;

border: 2px solid #ddd;

border-radius: 8px;

background-color: #f2f2f2;

}

input {

display: block;

width: 100%;

padding: 8px;

margin: 10px 0;

}

button {

padding: 8px 16px;

background-color: #007bff;

color: white;

border: none;

cursor: pointer;

border-radius: 4px;

}

button:hover {

background-color: #0056b3;

}

.result {

margin-top: 20px;

padding: 10px;

background-color: #e6ffe6;

border: 1px solid #ccc;

border-radius: 5px;

}

**App.js:**

import React from 'react';

import CalculateScore from './Components/CalculateScore';

function App() {

return (

<div className="App">

<CalculateScore />

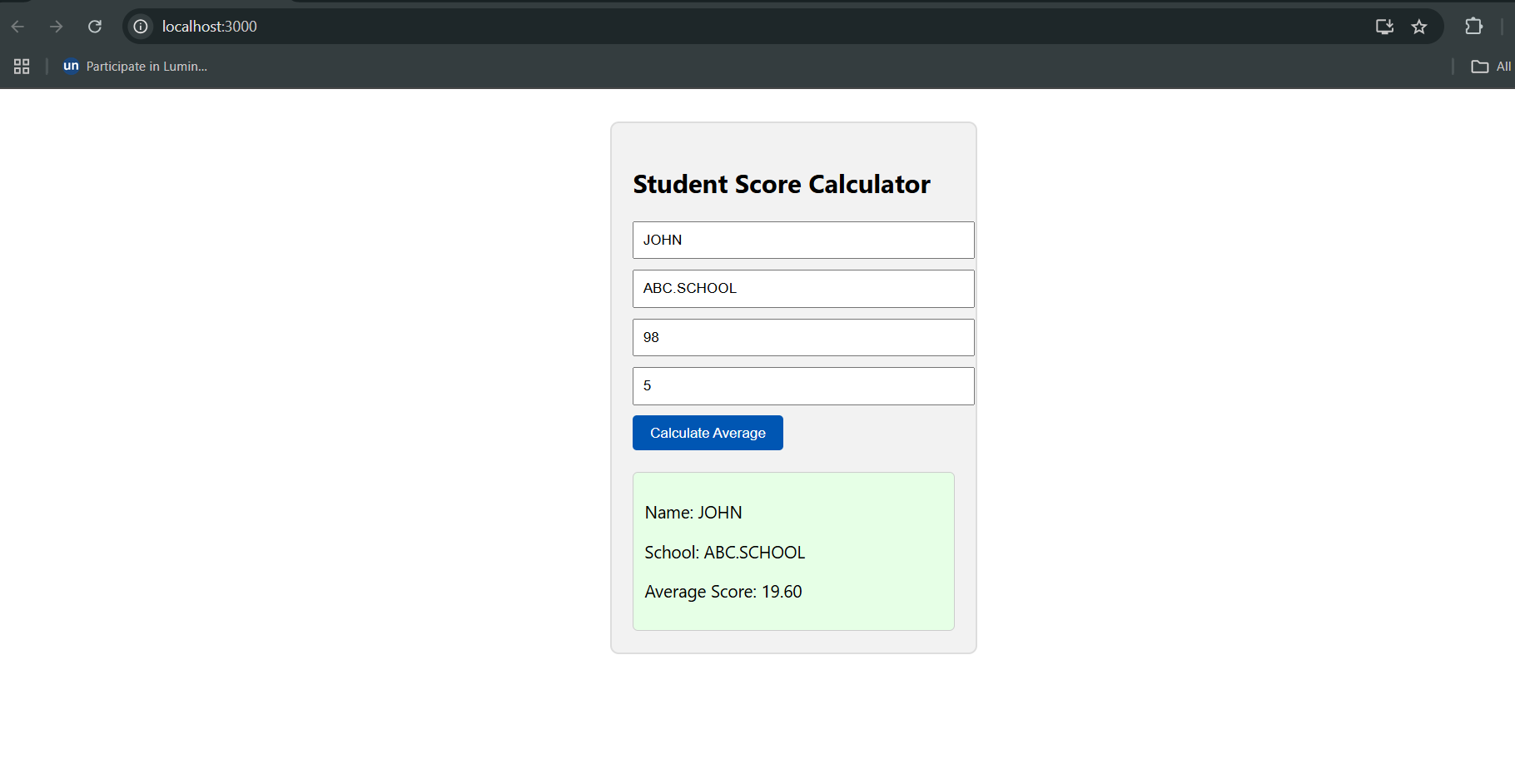
</div>

);

}

export default App;

**OUTPUT:**



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

**CODE:**

**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: [],

error: null

};

}

loadPosts() {

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

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

.then(data => {

const postObjects = data.map(

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

);

this.setState({ posts: postObjects });

})

.catch(error => {

this.setState({ error });

});

}

componentDidMount() {

this.loadPosts();

}

componentDidCatch(error, info) {

alert('An error occurred in Posts component.');

}

render() {

const { posts } = this.state;

return (

<div>

<h1>Blog Posts</h1>

{posts.map(post => (

<div key={post.id}>

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

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

<hr />

</div>

))}

</div>

);

}

}

export default Posts;

**App.js:**

import React from 'react';

import Posts from './Posts';

function App() {

return (

<div className="App">

<Posts />

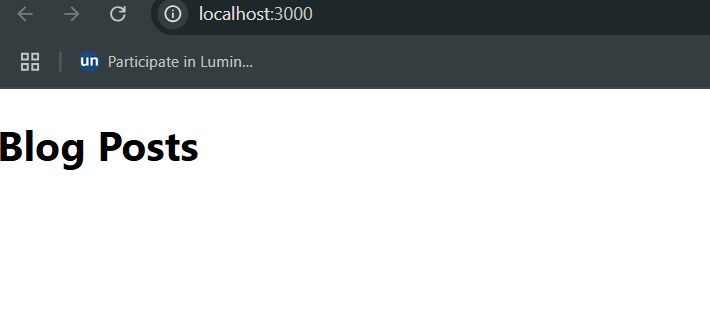
</div>

);

}

export default App;

**OUTPUT:**



**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.**

**CODE:**

**App.js:**

import React from 'react';

import CohortDetails from './coho/CohortDetails';

const cohorts = [

{

id: 'INTADMDF10',

tech: '.NET FSD',

startDate: '22-Feb-2022',

status: 'Scheduled',

coach: 'Aathma',

trainer: 'Jojo Jose',

},

{

id: 'ADM21JF014',

tech: 'Java FSD',

startDate: '10-Sep-2021',

status: 'Ongoing',

coach: 'Apoorv',

trainer: 'Elisa Smith',

},

{

id: 'CDBJF21025',

tech: 'Java FSD',

startDate: '24-Dec-2021',

status: 'Ongoing',

coach: 'Aathma',

trainer: 'John Doe',

}

];

function App() {

return (

<div>

<h1>Cohorts Details</h1>

{cohorts.map((cohort, index) => (

<CohortDetails key={index} cohort={cohort} />

))}

</div>

);

}

export default App;

**CohortDetails.js:**

import React from 'react';

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

function CohortDetails({ cohort }) {

const titleStyle = {

color: cohort.status === 'Ongoing' ? 'green' : 'blue',

};

return (

<div className={styles.box}>

<h3 style={titleStyle}>{cohort.id} - {cohort.tech}</h3>

<dl>

<dt>Started On</dt>

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

<dt>Current Status</dt>

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

<dt>Coach</dt>

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

<dt>Trainer</dt>

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

</dl>

</div>

);

}

export default CohortDetails;

**CohortDetails.module.css:**

.box {

width: 300px;

display: inline-block;

margin: 10px;

padding: 10px 20px;

border: 1px solid black;

border-radius: 10px;

vertical-align: top;

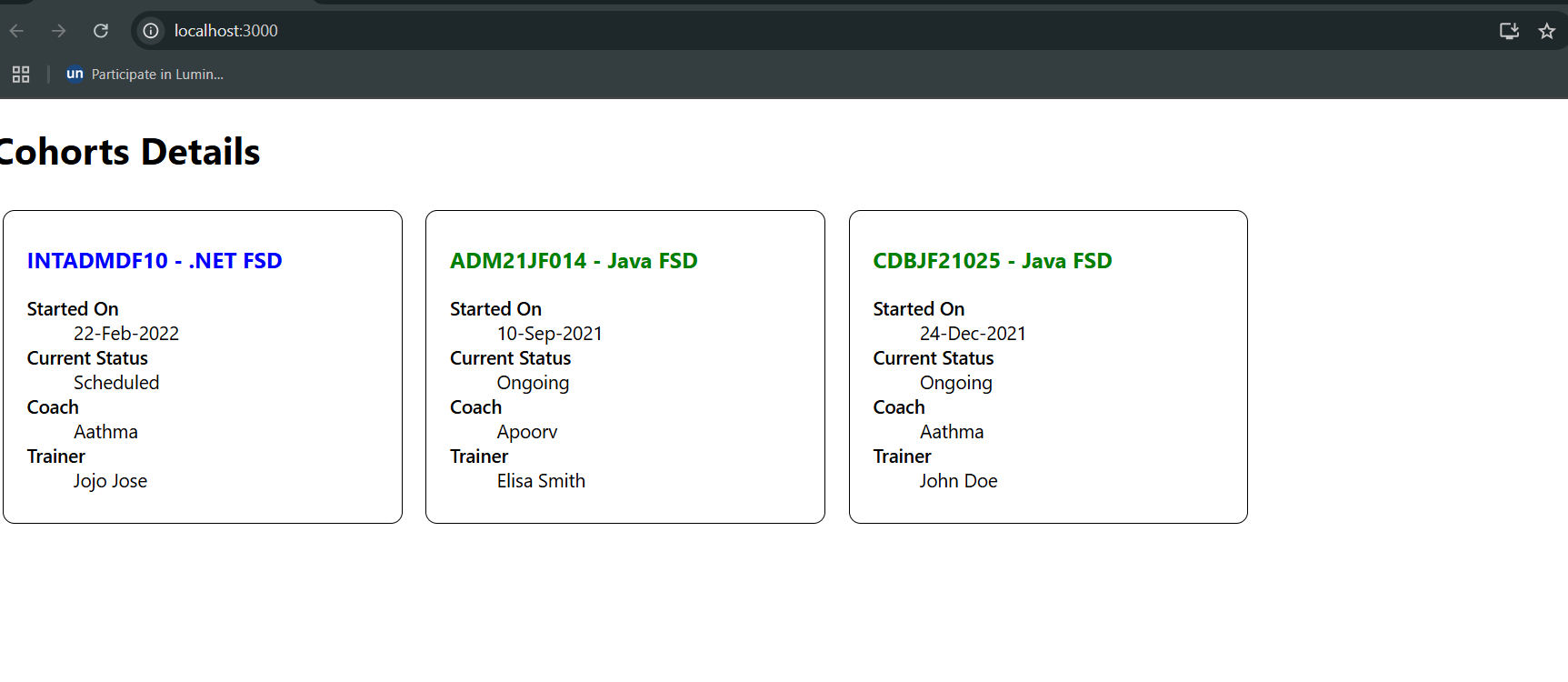
}

dt {

font-weight: 500;

}

**OUTPUT:**



**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.**

**CODE:**

**App.js:**

import React from 'react';

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

import Home from './Home';

import TrainersList from './TrainersList';

import TrainerDetails from './TrainerDetails';

function App() {

return (

<Router>

<div>

<h1>My Academy Trainers App</h1>

<nav>

<Link to="/">Home</Link> | <Link to="/trainers">Show Trainers</Link>

</nav>

<hr />

<Routes>

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

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

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

</Routes>

</div>

</Router>

);

}

export default App;

**Home.js:**

import React from 'react';

function Home() {

return (

<div>

<h1>My Academy Trainers App</h1>

<p>Welcome to the home page of My Academy Trainers App</p>

</div>

);

}

export default Home;

**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 === id);

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

return (

<div>

<h2>Trainers Details</h2>

<h3>{trainer.name} ({trainer.technology})</h3>

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

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

<ul>

{trainer.skills.map((skill, idx) => (

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

))}

</ul>

</div>

);

}

export default TrainerDetails;

**TrainersList.js:**

import React from 'react';

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

import trainers from './TrainersMock';

function TrainersList() {

return (

<div>

<h2>Trainers List</h2>

<ul>

{trainers.map((trainer) => (

<li key={trainer.trainerId}>

<Link to={`/trainer/${trainer.trainerId}`}>{trainer.name}</Link>

</li>

))}

</ul>

</div>

);

}

export default TrainersList;

**TrainersMock.js:**

import Trainer from "./trainer";

const trainers = [

new Trainer(

"T001",

"Syed Khaleelullah",

"khaleelullah@cognizant.com",

"97676516962",

".NET",

["C#", "SQL Server", "React", ".NET Core"]

),

new Trainer(

"T002",

"Jane Doe",

"jane.doe@cognizant.com",

"9876543210",

"Java",

["Java", "Spring Boot", "Hibernate"]

)

];

export default trainers;

**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;

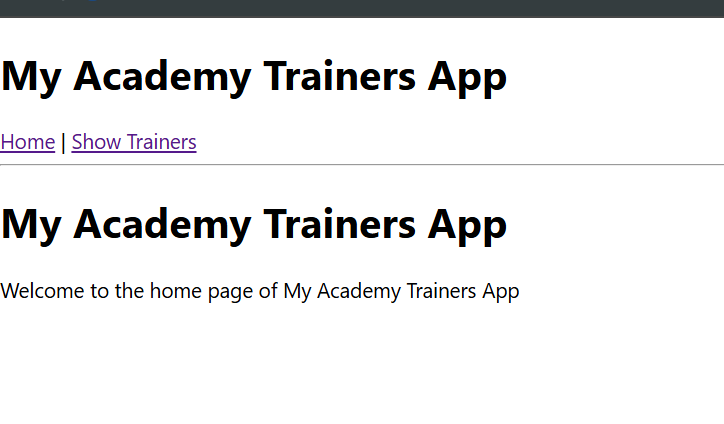
this.skills = skills;

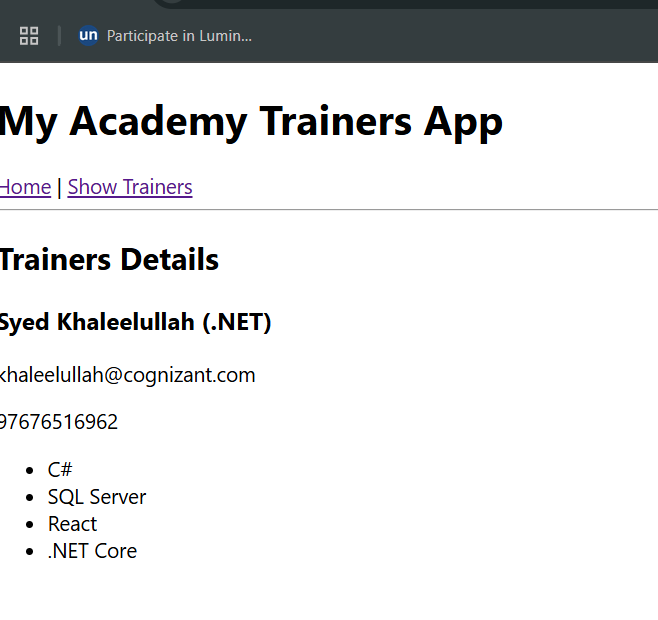
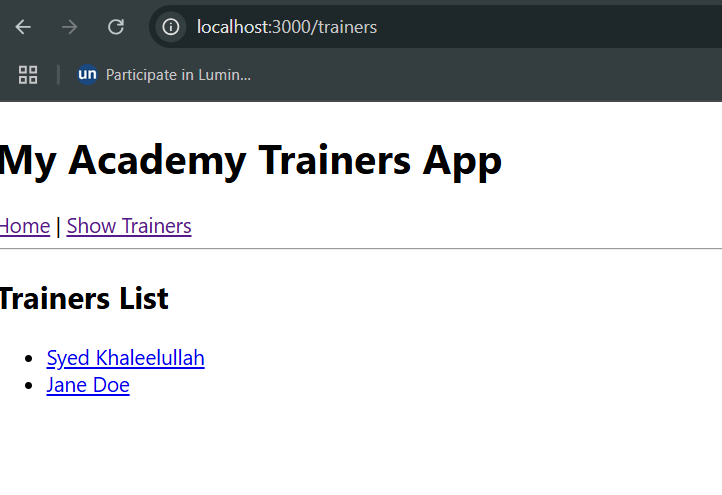
}

}

export default Trainer;

**OUTPUT:**

****

****

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

**CODE:**

**App.js:**

import React, { Component } from 'react';

import './App.css';

class Cart extends Component {

render() {

return (

<tr>

<td>{this.props.item.Itemname}</td>

<td>{this.props.item.Price}</td>

</tr>

);

}

}

class OnlineShopping extends Component {

constructor() {

super();

this.items = [

{ Itemname: "Laptop", Price: 80000 },

{ Itemname: "TV", Price: 120000 },

{ Itemname: "Washing Machine", Price: 50000 },

{ Itemname: "Mobile", Price: 30000 },

{ Itemname: "Fridge", Price: 70000 },

];

}

render() {

return (

<div className="App">

<h2>Items Ordered :</h2>

<table>

<thead>

<tr>

<th>Name</th>

<th>Price</th>

</tr>

</thead>

<tbody>

{this.items.map((item, index) => (

<Cart key={index} item={item} />

))}

</tbody>

</table>

</div>

);

}

}

export default OnlineShopping;

**App.css:**

.App {

text-align: center;

margin-top: 50px;

}

h2 {

color: green;

}

table {

margin: auto;

border-collapse: collapse;

font-family: Arial, sans-serif;

}

table, th, td {

border: 1px solid gray;

padding: 10px 20px;

}

th {

background-color: #d0f0d0;

color: green;

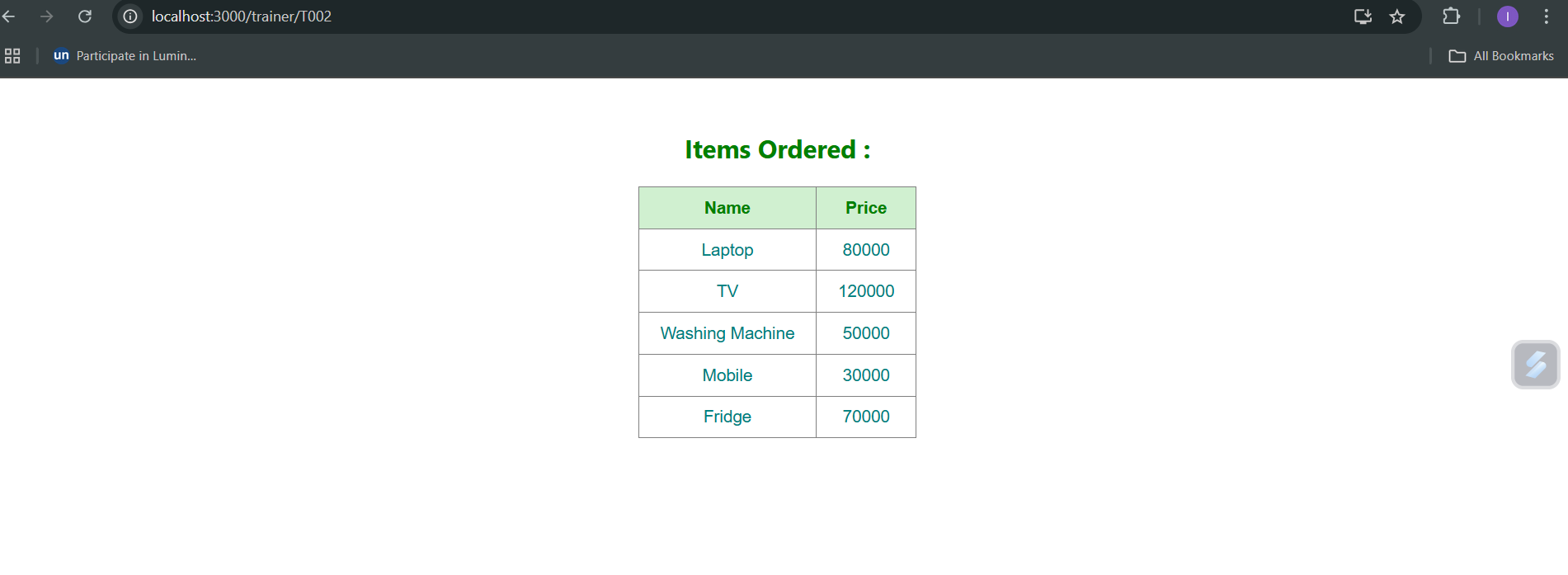
}

td {

color: teal;

}

**OUTPUT:**

****

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

**CODE:**

**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() {

const { entryCount, exitCount } = this.state;

return (

<div style={{ textAlign: 'center', marginTop: '100px' }}>

<button onClick={this.updateEntry} style={{ backgroundColor: 'lightgreen', padding: '10px', marginRight: '10px' }}>

Login

</button>

<span>{entryCount} People Entered!!!</span>

<button onClick={this.updateExit} style={{ backgroundColor: 'lightgreen', padding: '10px', marginLeft: '50px', marginRight: '10px' }}>

Exit

</button>

<span>{exitCount} People Left!!!</span>

</div>

);

}

}

export default CountPeople;

**App.js:**

import React from 'react';

import CountPeople from './CountPeople';

function App() {

return (

<div className="App">

<CountPeople />

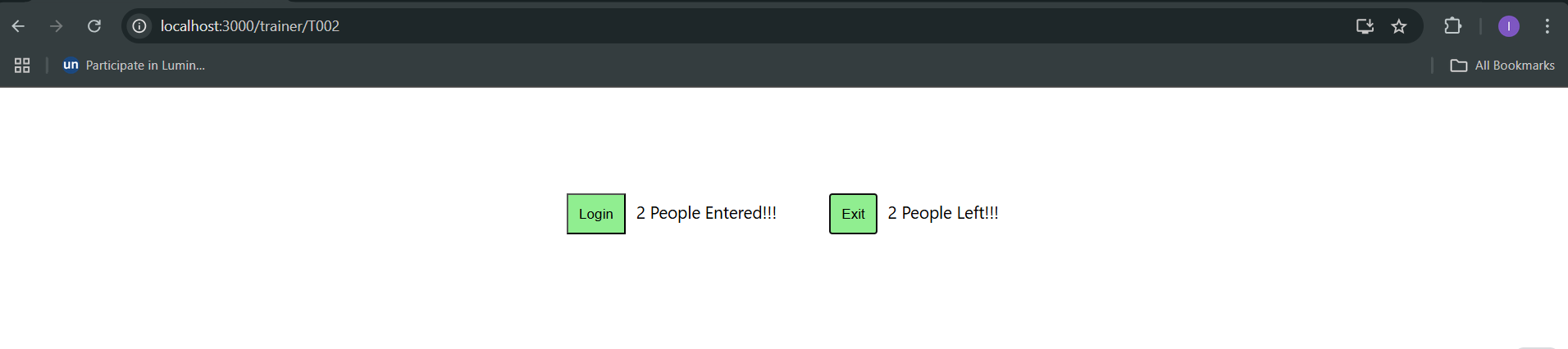
</div>

);

}

export default App;

**OUTPUT:**

****