**REACT HOL 9**

Create a React Application named “cricketapp” with the following components:

1. ListofPlayers

* Declare an array with 11 players and store details of their names and scores using the map feature of ES6

**App.js**

import logo from './logo.svg';

import './App.css';

function App() {

**const flag = true;**

const players = [

{ name: "Mr. Jack", score: 50 },

{ name: "Mr. Michael", score: 70 },

{ name: "Mr. John", score: 40 },

{ name: "Mr. Ann", score: 61 },

{ name: "Mr. Elisabeth", score: 61 },

{ name: "Mr. Sachin", score: 95 },

{ name: "Mr. Dhoni", score: 100 },

{ name: "Mr. Virat", score: 84 },

{ name: "Mr. Jadeja", score: 64 },

{ name: "Mr. Raina", score: 75 },

{ name: "Mr. Rohit", score: 80 }

];

const lowScorers = players.filter(player => player.score < 70);

const teamPlayers = [

"Sachin1", "Dhoni2", "Virat3", "Rohit4", "Yuvaraj5", "Raina6"

];

const oddPlayers = teamPlayers.filter((\_, index) => index % 2 === 0);

const evenPlayers = teamPlayers.filter((\_, index) => index % 2 !== 0);

const T20players = ["Mr. First Player", "Mr. Second Player", "Mr. Third Player"];

const RanjiPlayers = ["Mr. Fourth Player", "Mr. Fifth Player", "Mr. Sixth Player"];

const mergedPlayers = [...T20players, ...RanjiPlayers];

return (

<div className="App">

{flag ? (

<div>

<h2>List of Players</h2>

<ul>

{players.map((player, index) => (

<li key={index}>{player.name} {player.score}</li>

))}

</ul>

<hr />

<h2>List of Players having Scores Less than 70</h2>

<ul>

{lowScorers.map((player, index) => (

<li key={index}>{player.name} {player.score}</li>

))}

</ul>

</div>

) : (

<div>

<h2>Odd Players</h2>

<ul>

{oddPlayers.map((player, i) => (

<li key={i}>{["First", "Third", "Fifth"][i]} : {player}</li>

))}

</ul>

<h2>Even Players</h2>

<ul>

{evenPlayers.map((player, i) => (

<li key={i}>{["Second", "Fourth", "Sixth"][i]} : {player}</li>

))}

</ul>

<h2>List of Indian Players Merged:</h2>

<ul>

{mergedPlayers.map((player, i) => <li key={i}>{player}</li>)}

</ul>

</div>

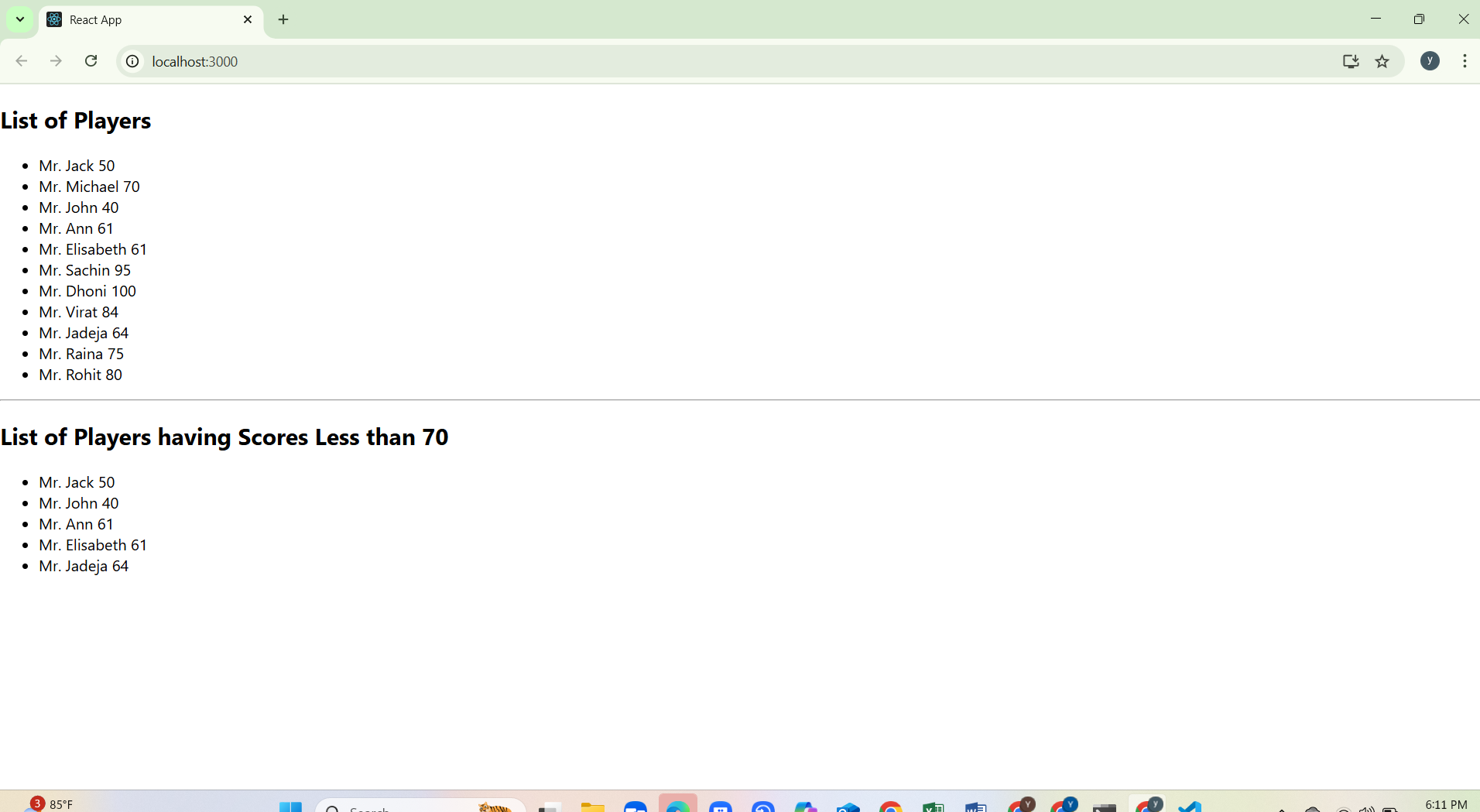
)}

</div>

);

}

export default App;



import logo from './logo.svg';

import './App.css';

function App() {

**const flag = false;**

const players = [

{ name: "Mr. Jack", score: 50 },

{ name: "Mr. Michael", score: 70 },

{ name: "Mr. John", score: 40 },

{ name: "Mr. Ann", score: 61 },

{ name: "Mr. Elisabeth", score: 61 },

{ name: "Mr. Sachin", score: 95 },

{ name: "Mr. Dhoni", score: 100 },

{ name: "Mr. Virat", score: 84 },

{ name: "Mr. Jadeja", score: 64 },

{ name: "Mr. Raina", score: 75 },

{ name: "Mr. Rohit", score: 80 }

];

const lowScorers = players.filter(player => player.score < 70);

const teamPlayers = [

"Sachin1", "Dhoni2", "Virat3", "Rohit4", "Yuvaraj5", "Raina6"

];

const oddPlayers = teamPlayers.filter((\_, index) => index % 2 === 0);

const evenPlayers = teamPlayers.filter((\_, index) => index % 2 !== 0);

const T20players = ["Mr. First Player", "Mr. Second Player", "Mr. Third Player"];

const RanjiPlayers = ["Mr. Fourth Player", "Mr. Fifth Player", "Mr. Sixth Player"];

const mergedPlayers = [...T20players, ...RanjiPlayers];

return (

<div className="App">

{flag ? (

<div>

<h2>List of Players</h2>

<ul>

{players.map((player, index) => (

<li key={index}>{player.name} {player.score}</li>

))}

</ul>

<hr />

<h2>List of Players having Scores Less than 70</h2>

<ul>

{lowScorers.map((player, index) => (

<li key={index}>{player.name} {player.score}</li>

))}

</ul>

</div>

) : (

<div>

<h2>Odd Players</h2>

<ul>

{oddPlayers.map((player, i) => (

<li key={i}>{["First", "Third", "Fifth"][i]} : {player}</li>

))}

</ul>

<h2>Even Players</h2>

<ul>

{evenPlayers.map((player, i) => (

<li key={i}>{["Second", "Fourth", "Sixth"][i]} : {player}</li>

))}

</ul>

<h2>List of Indian Players Merged:</h2>

<ul>

{mergedPlayers.map((player, i) => <li key={i}>{player}</li>)}

</ul>

</div>

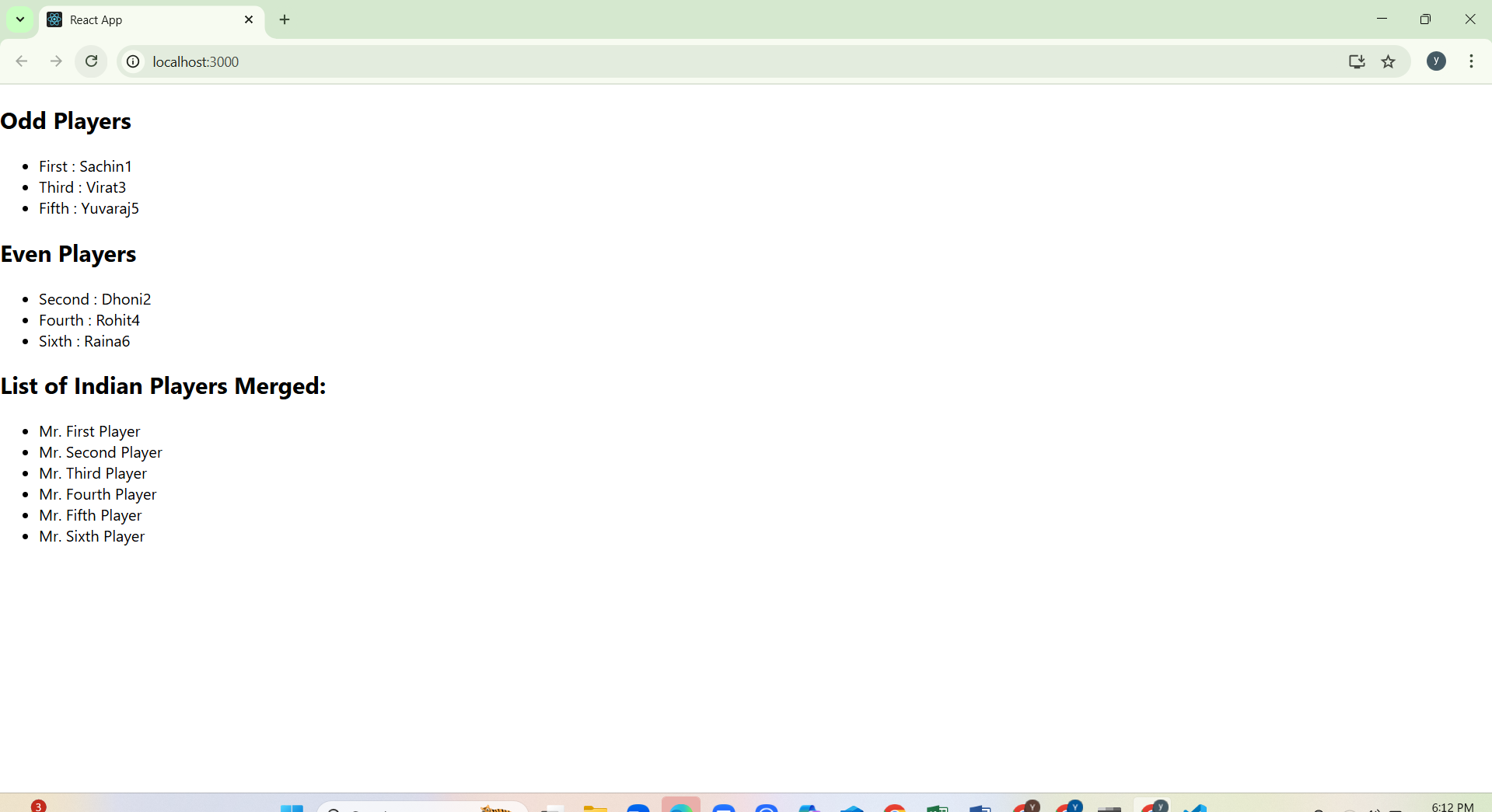
)}

</div>

);

}

export default App;



**REACT HOL 10**

Create a React Application named “officespacerentalapp” which uses React JSX to create elements, attributes and renders DOM to display the page.

**App.js**

import logo from './logo.svg';

import './App.css';

import React from "react";

function App() {

const office = {

name: "DBS",

rent: 50000,

address: "Chennai",

image:

"https://cdn-klbmd.nitrocdn.com/oflAEolSqrPhKMSrYpDBcfHfpcFckcly/assets/images/optimized/rev-1bcb9bb/www.dbsindia.com/images/locations-pages/chennai.webp"

};

return (

<div style={{ textAlign: "center", padding: "20px", fontFamily: "Arial" }}>

<h1>Office Space, at Affordable Range</h1>

<img

src={office.image}

alt="Office Space"

style={{ width: "400px", height: "250px", marginBottom: "20px" }}

/>

<h2>Name: {office.name}</h2>

<p

style={{

color: office.rent < 60000 ? "red" : "green",

fontWeight: "bold"

}}

>

Rent: Rs. {office.rent}

</p>

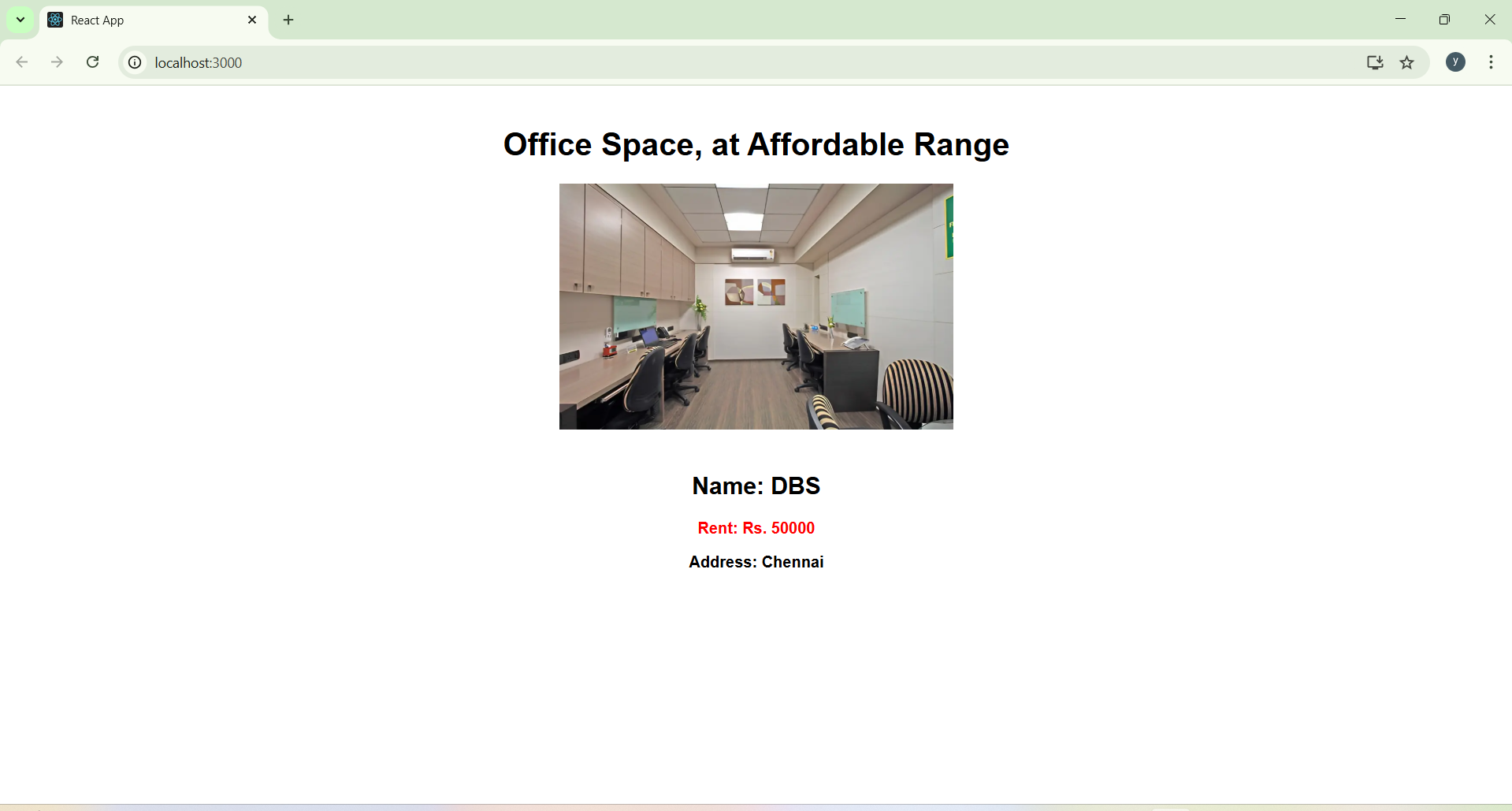
<p style={{ fontWeight: "bold" }}>Address: {office.address}</p>

</div>

);

}

export default App;



**REACT HOL 11**

Create a React Application “eventexamplesapp” to handle various events of the form elements in HTML.

**App.js**

import React, { useState } from 'react';

function App() {

const [count, setCount] = useState(1);

const [amount, setAmount] = useState('');

const [currency, setCurrency] = useState('Euro');

function handleConvert(e) {

e.preventDefault();

const RATE = 80;

const result = Number(amount) \* RATE;

alert(`Converting to ${currency} Amount is ${result}`);

}

function sayMessage(msg) {

alert(msg);

}

return (

<div style={{ fontFamily: 'Arial, sans-serif', margin: 40 }}>

<div>

<span style={{ fontSize: 18 }}>{count}</span>

<br /><br />

<button onClick={() => setCount(count + 1)}>Increment</button>

<br /><br />

<button onClick={() => setCount(count - 1)}>Decrement</button>

<br /><br />

<button onClick={() => sayMessage('welcome')}>Say welcome</button>

<br /><br />

<button onClick={() => sayMessage('You clicked on me!')}>Click on me</button>

</div>

<br /><br />

<div>

<div style={{ color: 'green', fontWeight: 'bold', fontSize: 32 }}>

Currency Convertor!!!

</div>

<br />

<form onSubmit={handleConvert}>

<div>

<label>

<span style={{ fontWeight: 'bold' }}>Amount:</span>&nbsp;

<input

type="number"

value={amount}

onChange={e => setAmount(e.target.value)}

style={{ width: 150, fontSize: 16 }}

required

/>

</label>

</div>

<br />

<div>

<label>

<span style={{ fontWeight: 'bold' }}>Currency:</span>&nbsp;

<input

type="text"

value={currency}

onChange={e => setCurrency(e.target.value)}

style={{ width: 150, fontSize: 16 }}

required

/>

</label>

</div>

<br />

<button type="submit">Submit</button>

</form>

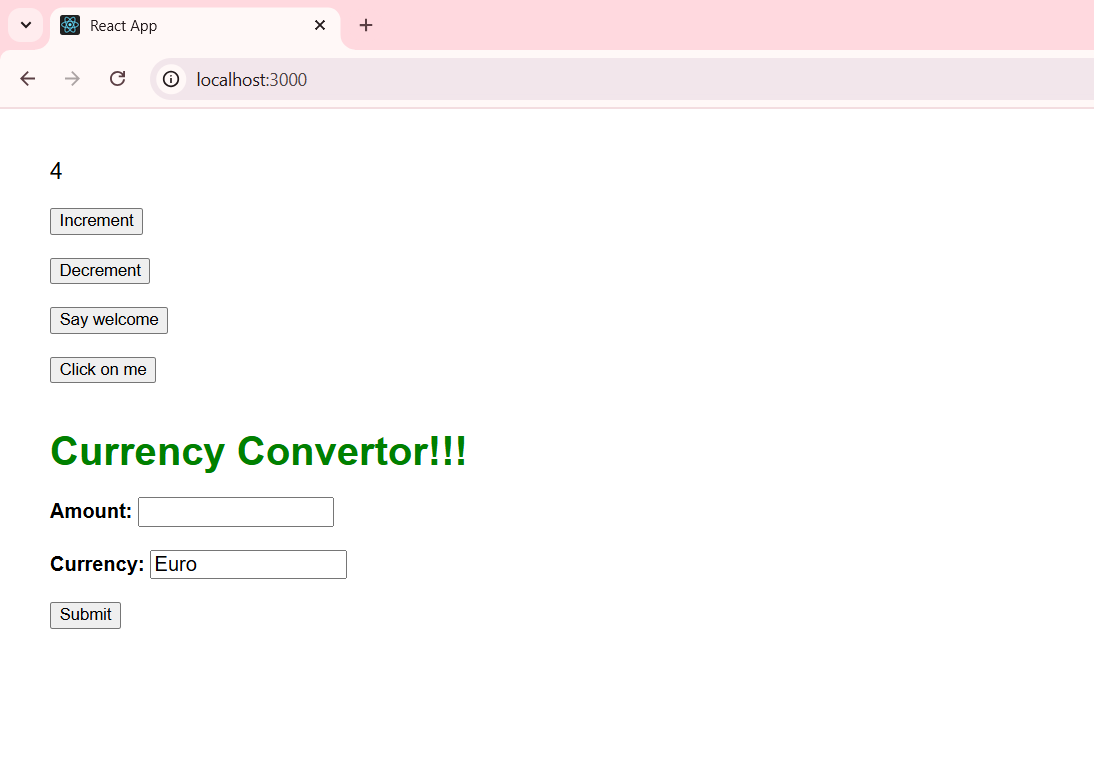
</div>

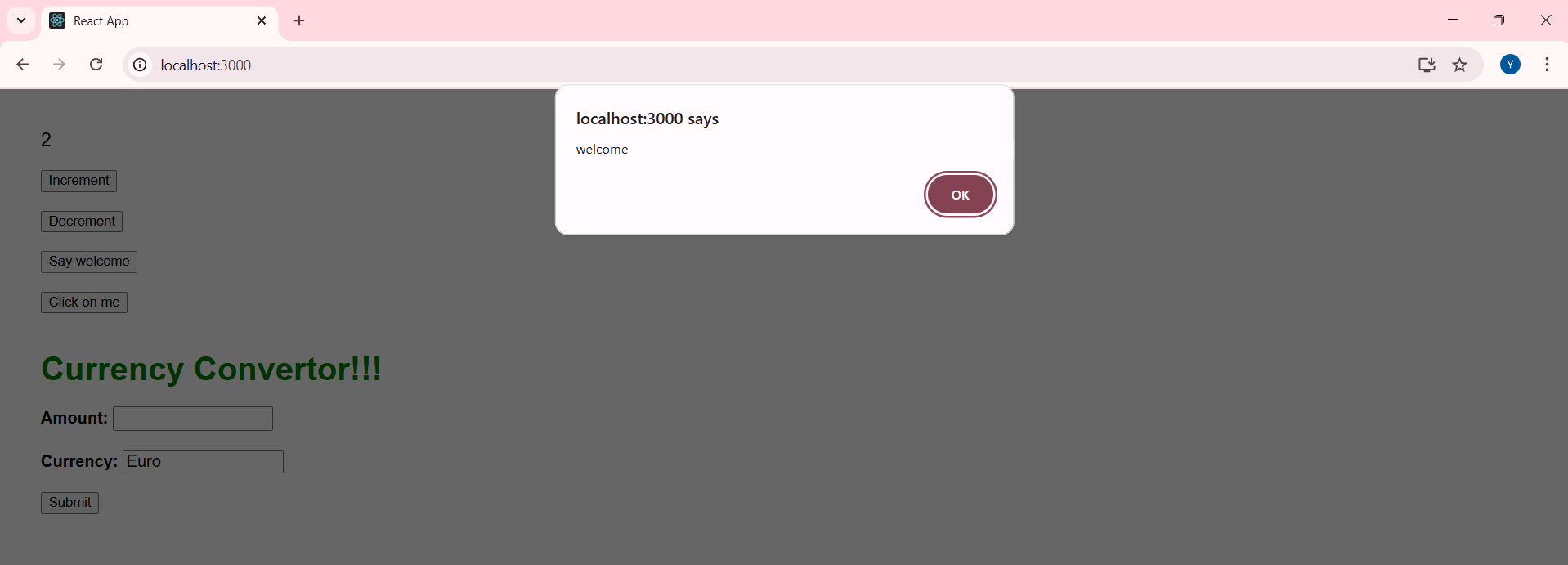
</div>

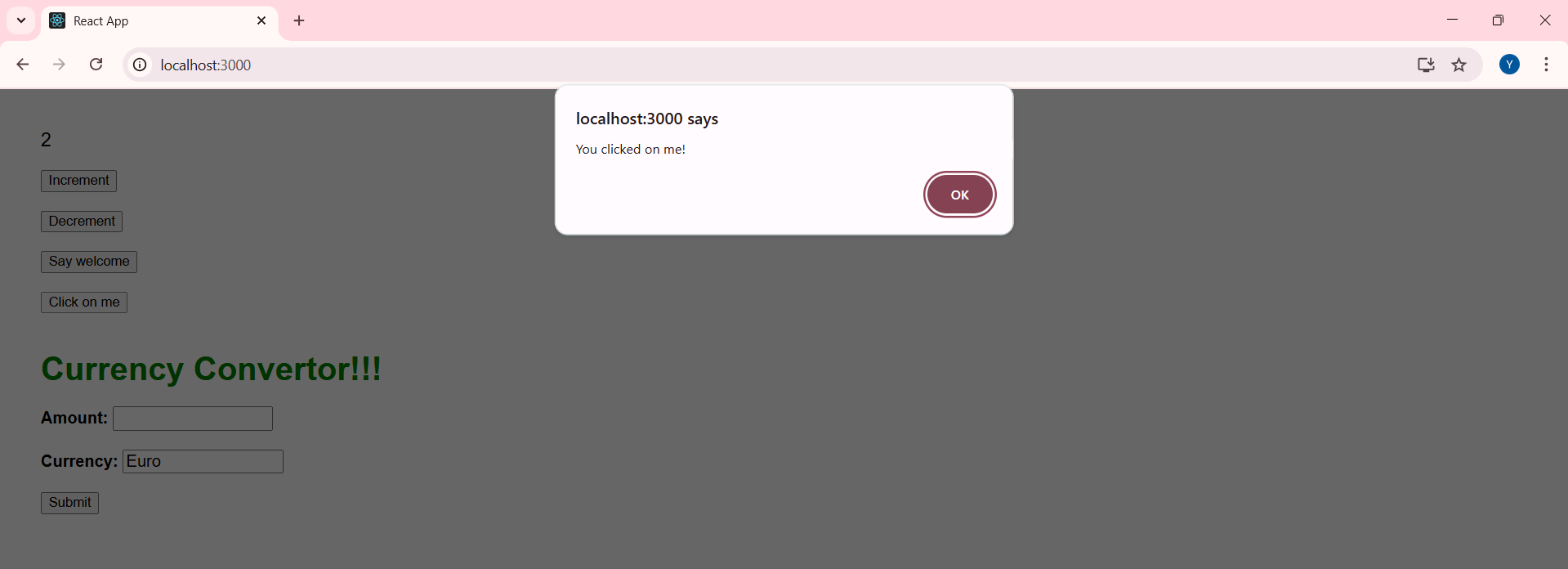
);

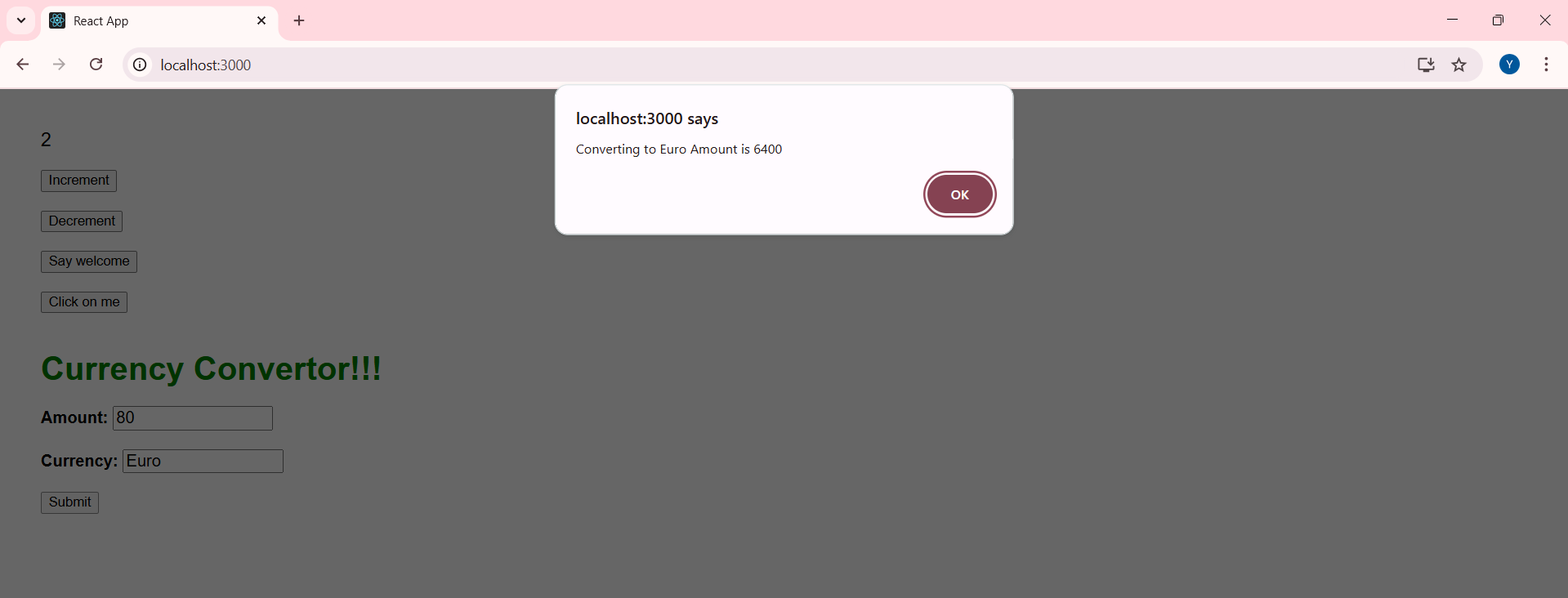
}

export default App;









**REACT HOL 12**

Create a React Application named “ticketbookingapp” where the guest user can browse the page where the flight details are displayed whereas the logged in user only can book tickets.

The Login and Logout buttons should accordingly display different pages. Once the user is logged in the User page should be displayed. When the user clicks on Logout, the Guest page should be displayed.

**App.js**

import React, { useState } from "react";

function LoginButton(props) {

return (

<button onClick={props.onClick}>

Login

</button>

);

}

function LogoutButton(props) {

return (

<button onClick={props.onClick}>

Logout

</button>

);

}

function Greeting(props) {

if (props.isLoggedIn) {

return <h1>Welcome back</h1>;

}

return <h1>Please sign up.</h1>;

}

export default function App() {

const [isLoggedIn, setIsLoggedIn] = useState(false);

const handleLoginClick = () => setIsLoggedIn(true);

const handleLogoutClick = () => setIsLoggedIn(false);

return (

<div style={{ margin: "100px" }}>

<Greeting isLoggedIn={isLoggedIn} />

{isLoggedIn ? (

<LogoutButton onClick={handleLogoutClick} />

) : (

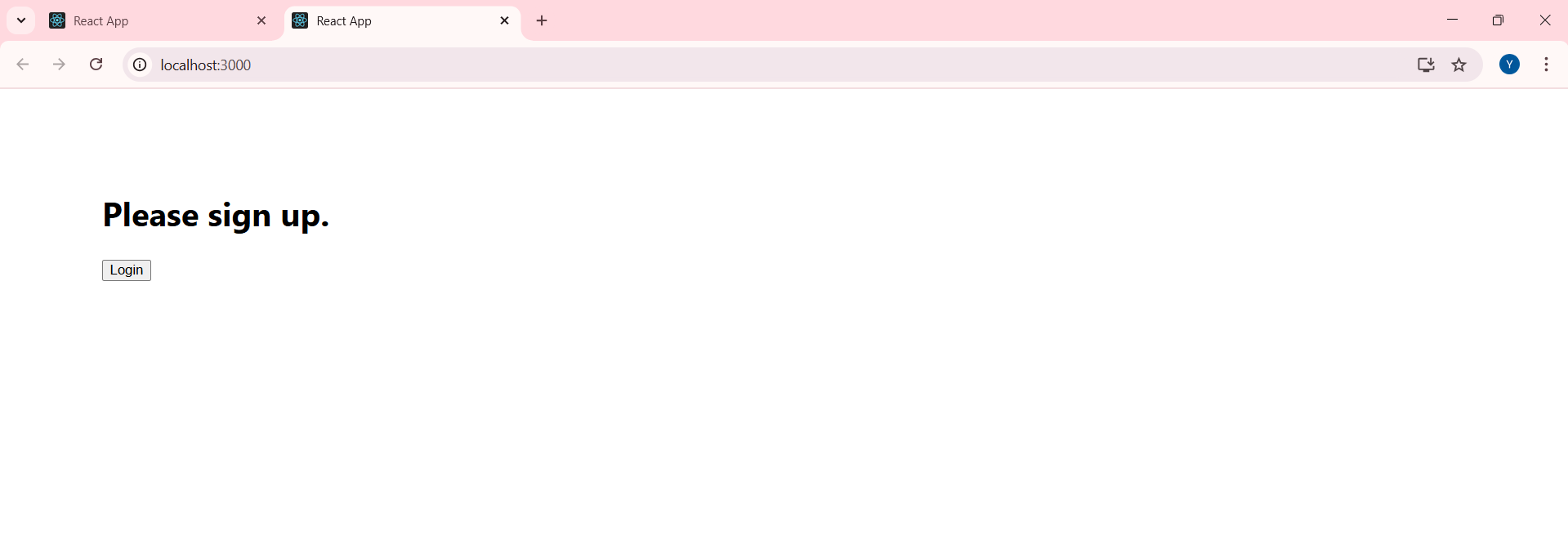
<LoginButton onClick={handleLoginClick} />

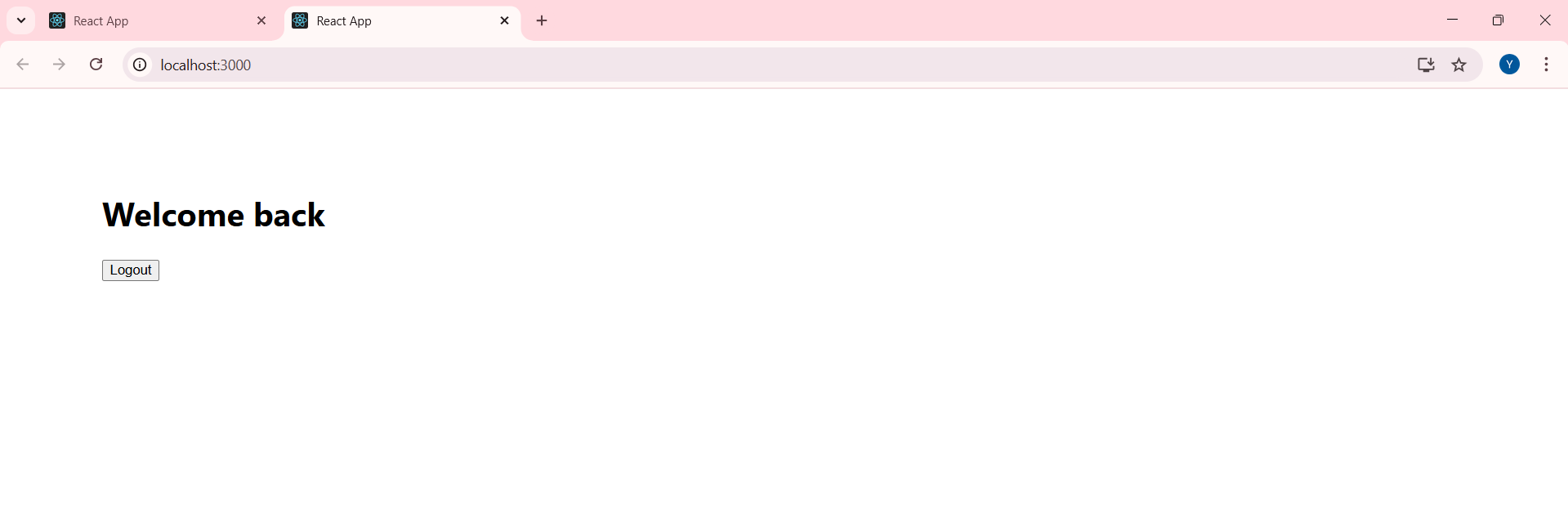
)}

</div>

);

}





**REACT HOL 13**

Create a React App named “bloggerapp” in with 3 components.

1. Book Details
2. Blog Details
3. Course Details

**App.js**

import React from "react";

import "./App.css";

const books = [

{ id: 101, bname: "Master React", price: 670 },

{ id: 102, bname: "Deep Dive into Angular 11", price: 800 },

{ id: 103, bname: "Mongo Essentials", price: 450 },

];

const courses = [

{ id: 1, name: "Angular", date: "4/5/2021" },

{ id: 2, name: "React", date: "6/3/2021" },

];

const blogs = [

{

title: "React Learning",

author: "Stephen Biz",

content: "Welcome to learning React!"

},

{

title: "Installation",

author: "Schwezdenier",

content: "You can install React from npm."

}

];

const BookDetails = ({ books }) => (

<ul>

{books.map((book) => (

<li key={book.id}>

<h3>{book.bname}</h3>

<h4>{book.price}</h4>

</li>

))}

</ul>

);

const CourseDetails = ({ courses }) => (

<ul>

{courses.map((course) => (

<li key={course.id}>

<h2>{course.name}</h2>

<h4>{course.date}</h4>

</li>

))}

</ul>

);

const BlogDetails = ({ blogs }) => (

<ul>

{blogs.map((blog, idx) => (

<li key={idx}>

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

<h4>{blog.author}</h4>

<span>{blog.content}</span>

</li>

))}

</ul>

);

function App() {

return (

<div className="main-container">

<div className="content-row">

<div className="column">

<h1>Course Details</h1>

<CourseDetails courses={courses} />

</div>

<div className="v1"></div>

<div className="column">

<h1>Book Details</h1>

<BookDetails books={books} />

</div>

<div className="v1"></div>

<div className="column">

<h1>Blog Details</h1>

<BlogDetails blogs={blogs} />

</div>

</div>

</div>

);

}

export default App;

**App.css**

.App {

text-align: center;

}

.App-logo {

height: 40vmin;

pointer-events: none;

}

@media (prefers-reduced-motion: no-preference) {

.App-logo {

animation: App-logo-spin infinite 20s linear;

}

}

.App-header {

background-color: #282c34;

min-height: 100vh;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

font-size: calc(10px + 2vmin);

color: white;

}

.App-link {

color: #61dafb;

}

.main-container {

width: 100vw;

min-height: 100vh;

padding-top: 32px;

background: #fff;

}

.content-row {

display: flex;

justify-content: center;

align-items: flex-start;

gap: 36px;

max-width: 1200px;

margin: 0 auto;

}

.column {

flex: 1 1 0;

min-width: 260px;

padding: 0 20px;

box-sizing: border-box;

}

.column h1 {

font-size: 2rem;

font-weight: bold;

margin-bottom: 32px;

color: #111;

}

ul {

list-style: none;

margin: 0;

padding: 0;

}

li {

margin-bottom: 30px;

}

.column h2 {

font-size: 1.5rem;

font-weight: 700;

margin: 0 0 7px 0;

color: #222;

}

.column h3,

.column h4 {

font-size: 1.2rem;

margin: 0 0 7px 0;

}

.column h4 {

font-size: 1rem;

font-weight: bold;

margin-bottom: 7px;

color: #222;

}

.column span {

font-size: 1rem;

color: #222;

}

.v1 {

border-left: 4px solid green;

height: 380px;

margin: 0 18px;

align-self: stretch;

}

@media (max-width: 900px) {

.content-row {

flex-direction: column;

gap: 0;

max-width: 95vw;

}

.v1 {

display: none;

}

.column {

margin-bottom: 36px;

padding: 0;

}

}

@keyframes App-logo-spin {

from {

transform: rotate(0deg);

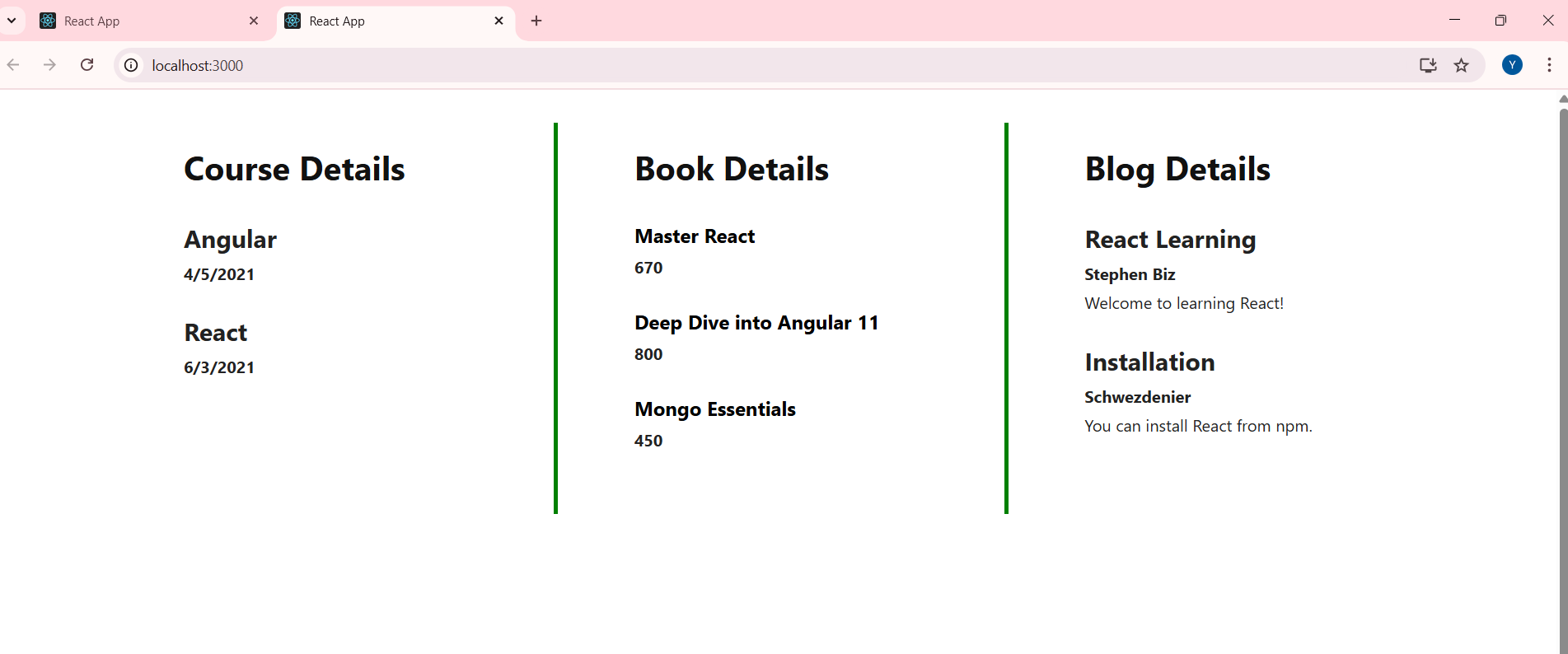
}

to {

transform: rotate(360deg);

}

}



**REACT HOL 14**

Developers of Apps Centric Solutions have created an employee management application which supports light and dark themes for the buttons. The current solution uses the react state and props to provide the theme name to be used from App component to Employee List component and from there to Employee Card component. Quality assurance team analyzed the solutions and found the technique being used to be a substandard one. React architect suggested to use the react context API to share the theme name with nested child components instead of passing them down using props from the parent component.

You are assigned the task of converting the application form props only to React Context API.

**ThemeContext.js**

import { createContext } from "react";

const ThemeContext = createContext("light");

export default ThemeContext;

**Employee.js**

class Employee {

constructor(id, name, email, phone){

this.id=id;

this.name=name;

this.email=email;

this.phone=phone;

}

}

const EmployeesData=[

new Employee(101,'Jojo','jojo@congizant.com','98238971234'),

new Employee(102,'Sam','sam@congizant.com', '9981184126'),

new Employee(103,'Elisa','elisa@cognizant.com','9989389735')

];

export default Employee;

export {EmployeesData};

**EmployeeCard.js**

import React, { useContext } from 'react';

import Styles from './EmployeeCard.module.css';

import ThemeContext from './ThemeContext';

function EmployeeCard(props) {

const theme = useContext(ThemeContext);

return (

<div className={Styles.Card}>

<h3>{props.employee.name}</h3>

<p>{props.employee.email}</p>

<p>{props.employee.phone}</p>

<p>

<a href="#" className={theme}>Edit</a>

<a href="#" className={theme}>Delete</a>

</p>

</div>

);

}

export default EmployeeCard;

**EmployeesList.js**

import EmployeeCard from './EmployeeCard';

function EmployeesList(props) {

return (

<div>

<h1>Employees List</h1>

{

props.employees.map(employee =>

<EmployeeCard employee={employee} key={employee.id} />

)

}

</div>

);

}

export default EmployeesList;

**App.js**

import React, { useState } from 'react';

import EmployeesList from './EmployeesList';

import { EmployeesData } from './Employee';

import ThemeContext from './ThemeContext';

function App() {

const [theme, setTheme] = useState('light');

return (

<ThemeContext.Provider value={theme}>

<div>

<h1>Employee Management</h1>

<EmployeesList employees={EmployeesData} />

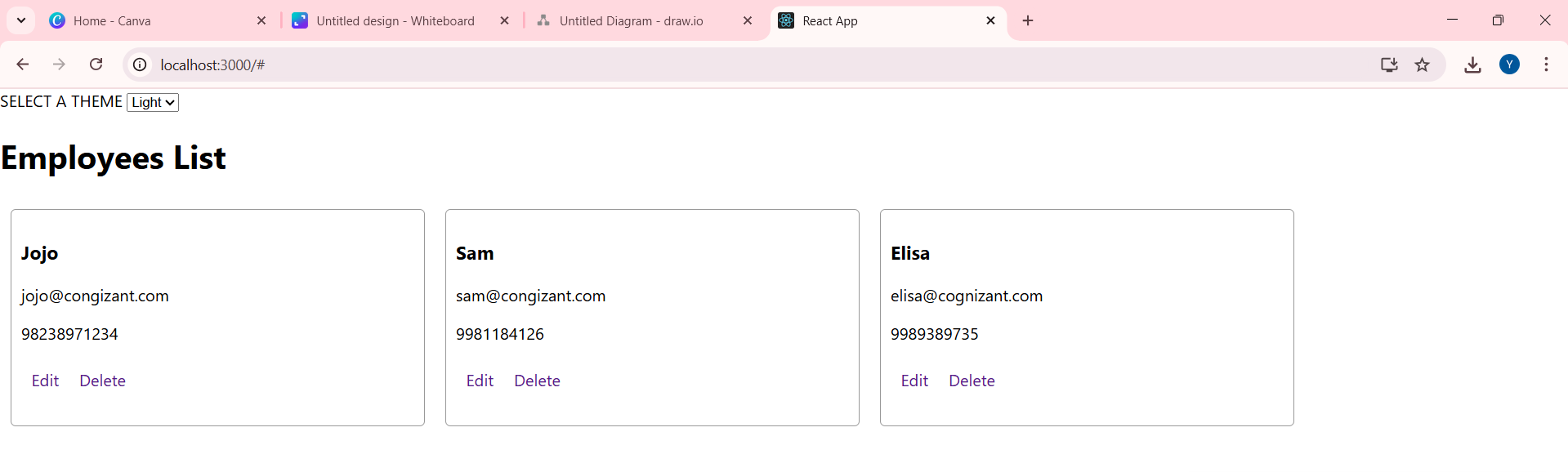
</div>

</ThemeContext.Provider>

);

}

export default App;



**REACT HOL 15**

Create a React App named “ticketraisingapp” which will help to raise a complaint and get it resolved.

Create a component named “ComplaintRegister” with a form containing a textbox to enter the employee name and a textarea to enter the complaint. Use “handleSubmit” event of the button to submit the complaint and generate a Reference number for further follow ups in the alert box.

**ComplaintRegister.js**

import React, { useState } from 'react';

const ComplaintRegister = () => {

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

const [complaint, setComplaint] = useState('');

const handleSubmit = (e) => {

e.preventDefault();

const transactionId = Math.floor(Math.random() \* 100) + 1;

alert(`Thanks ${name}!\nYour Complaint was Submitted.\nTransaction ID is: ${transactionId}`);

setName('');

setComplaint('');

};

return (

<div style={{ textAlign: 'center', marginTop: '50px' }}>

<h2 style={{ color: 'red', fontWeight: 'bold' }}>Register your complaints here!!!</h2>

<form onSubmit={handleSubmit}>

<div style={{ margin: '10px' }}>

<label style={{ marginRight: '10px' }}>Name:</label>

<input

type="text"

value={name}

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

required

/>

</div>

<div style={{ margin: '10px' }}>

<label style={{ marginRight: '10px' }}>Complaint:</label>

<textarea

value={complaint}

onChange={(e) => setComplaint(e.target.value)}

required

/>

</div>

<button type="submit">Submit</button>

</form>

</div>

);

};

export default ComplaintRegister;

**App.js**

import React from 'react';

import ComplaintRegister from './ComplaintRegister';

function App() {

return (

<div>

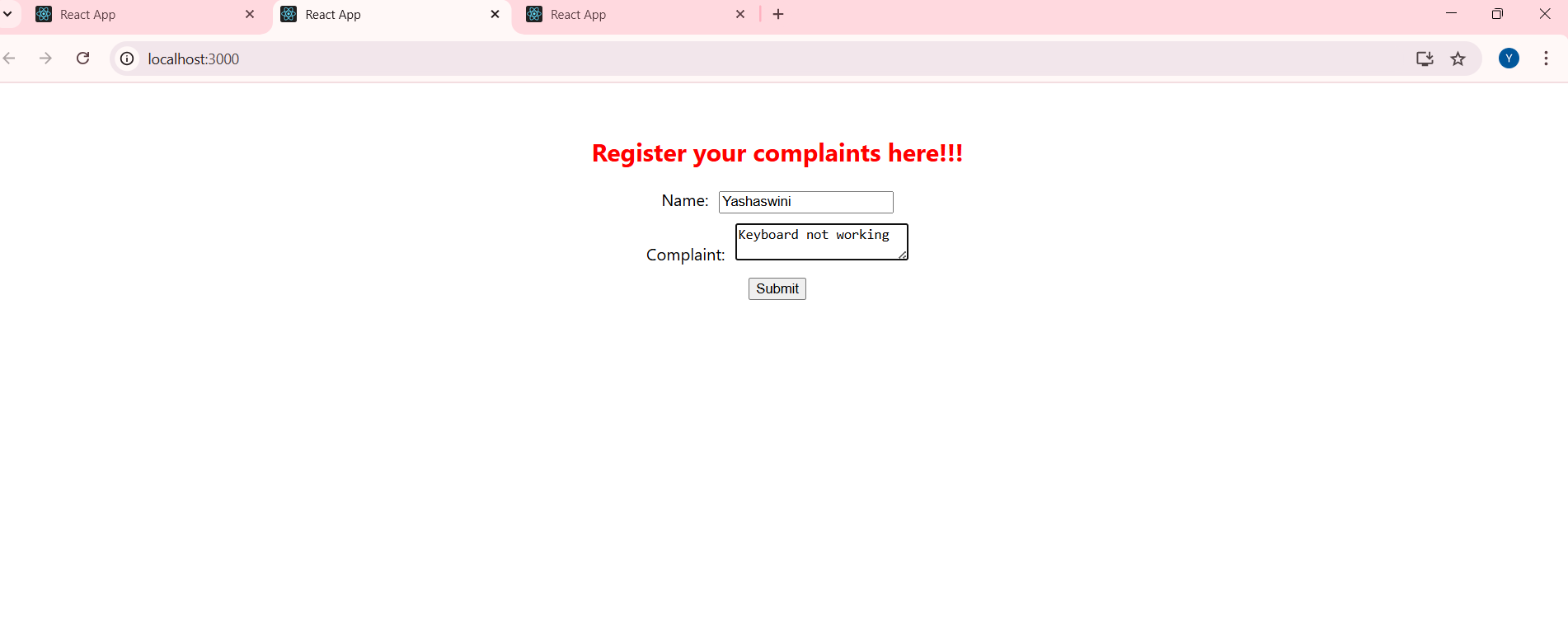
<ComplaintRegister />

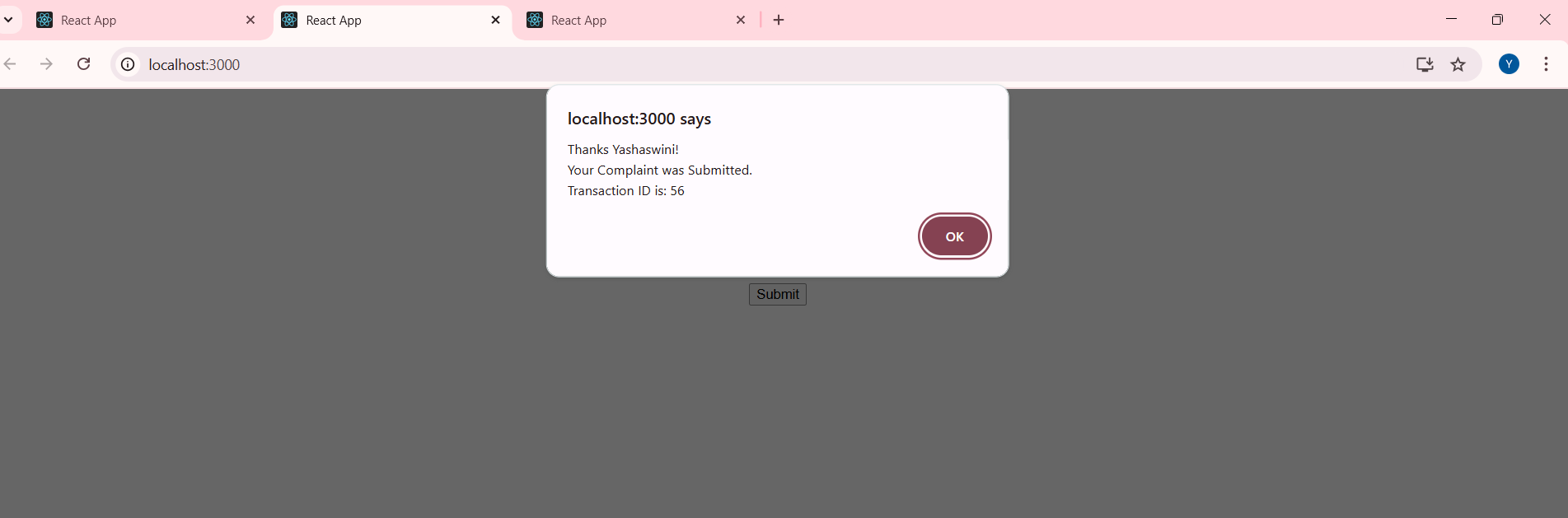
</div>

);

}

export default App;





**REACT HOL 16**

Create a React App named “mailregisterapp” which will have a component named “register.js”. Create a form which accepts the name, email and password and validate the fields as per the following:

1. Name should have atleast 5 characters
2. Email should have @ and .
3. Password should have atleast 8 characters.

Ensure that validations are implemented through eventhandle and eventsubmit of a form.

**Register.js**

import React, { Component } from 'react';

class Register extends Component {

constructor(props) {

super(props);

this.state = {

fullName: '',

email: '',

password: '',

errors: {

fullName: '',

email: '',

password: ''

}

};

}

handleChange = (event) => {

const { name, value } = event.target;

let errors = this.state.errors;

switch (name) {

case 'fullName':

errors.fullName =

value.length < 5 ? 'Full Name must be 5 characters long!' : '';

break;

case 'email':

errors.email =

value.includes('@') && value.includes('.')

? ''

: 'Email is not valid!';

break;

case 'password':

errors.password =

value.length < 8 ? 'Password must be 8 characters long!' : '';

break;

default:

break;

}

this.setState({ errors, [name]: value });

};

validateForm = (errors) => {

return Object.values(errors).every((val) => val === '');

};

handleSubmit = (event) => {

event.preventDefault();

const { errors } = this.state;

if (this.validateForm(errors)) {

alert('Valid Form');

} else {

if (errors.fullName !== '') {

alert(errors.fullName);

}

if (errors.email !== '') {

alert(errors.email);

}

if (errors.password !== '') {

alert(errors.password);

}

}

};

render() {

return (

<div style={{ textAlign: 'center', marginTop: '50px' }}>

<h2 style={{ color: 'red', fontWeight: 'bold' }}>Register Here!!!</h2>

<form onSubmit={this.handleSubmit}>

<div style={{ margin: '10px' }}>

<label style={{ marginRight: '10px' }}>Name:</label>

<input

type="text"

name="fullName"

onChange={this.handleChange}

required

/>

</div>

<div style={{ margin: '10px' }}>

<label style={{ marginRight: '10px' }}>Email:</label>

<input

type="text"

name="email"

onChange={this.handleChange}

required

/>

</div>

<div style={{ margin: '10px' }}>

<label style={{ marginRight: '10px' }}>Password:</label>

<input

type="password"

name="password"

onChange={this.handleChange}

required

/>

</div>

<button type="submit">Submit</button>

</form>

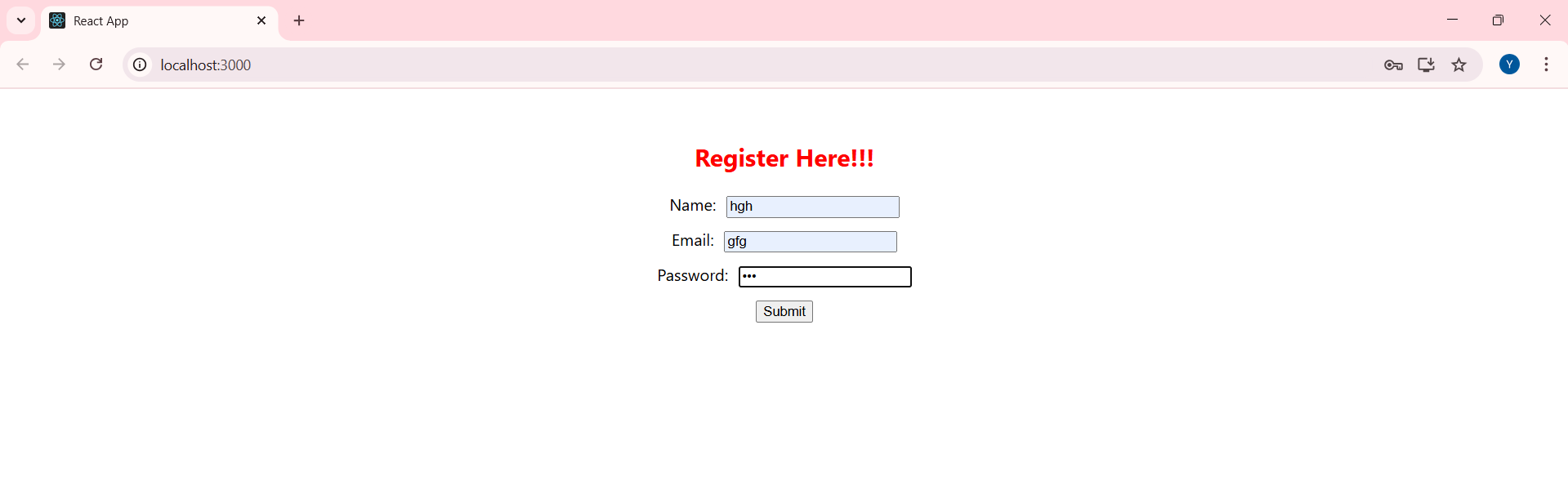
</div>

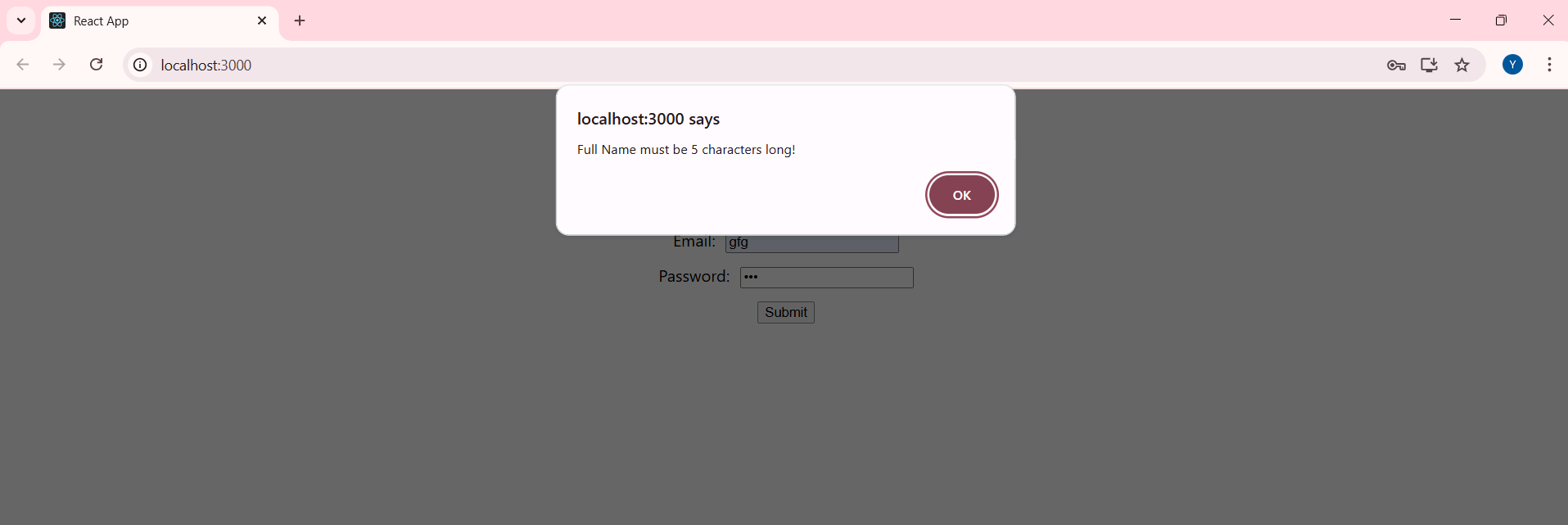
);

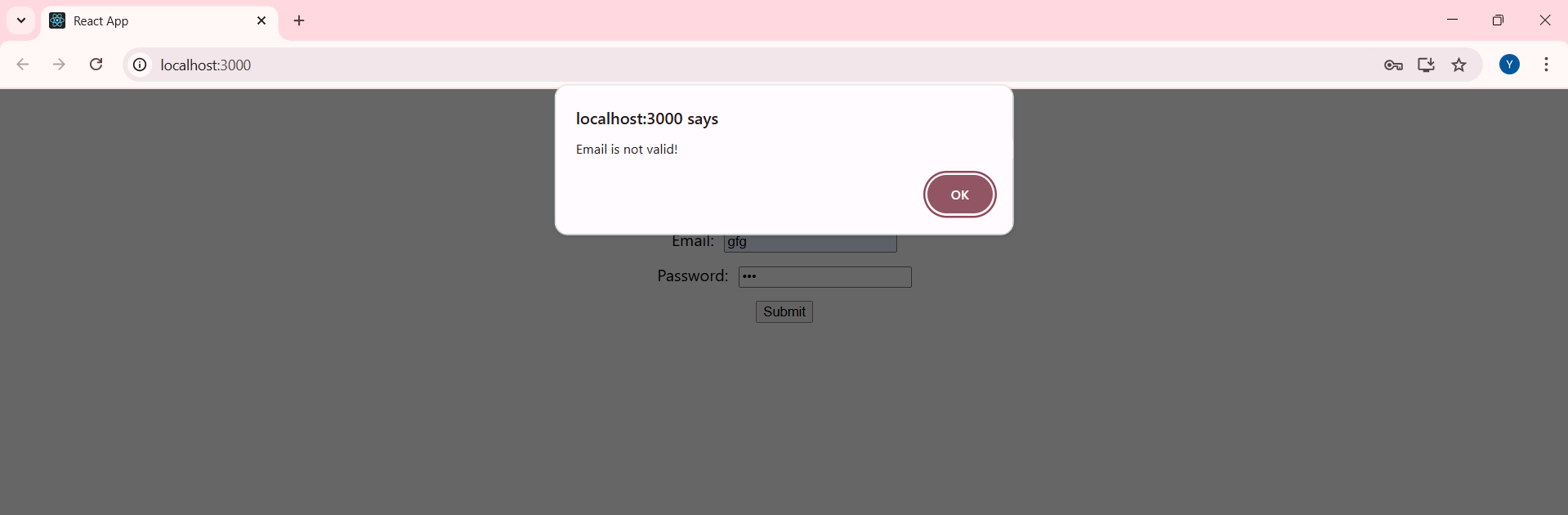
}

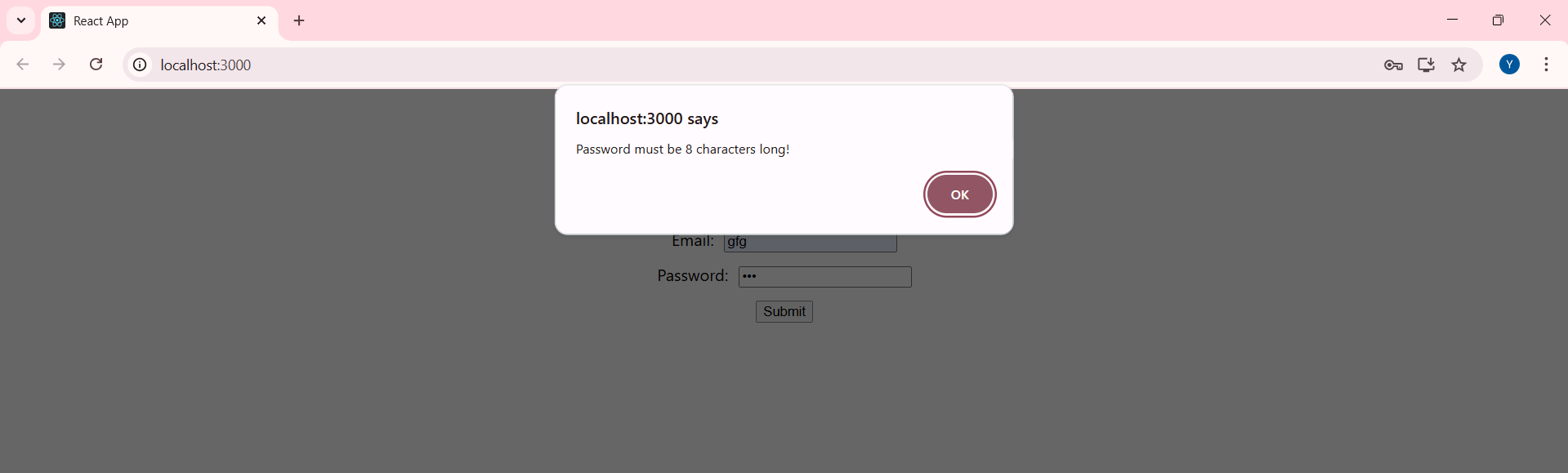
}

export default Register;









**REACT HOL 17**

Create a React Application “fetchuserapp” which will retrieve the user details from <https://api.randomuser.me/> and display the title, firstname and image of a user

**Fetchuserapp.js**

import React, { Component } from 'react';

class Getuser extends Component {

constructor(props) {

super(props);

this.state = {

person: null,

loading: true

};

}

async componentDidMount() {

const url = "https://api.randomuser.me/";

const response = await fetch(url);

const data = await response.json();

this.setState({ person: data.results[0], loading: false });

console.log(data.results[0]);

}

render() {

const { person, loading } = this.state;

if (loading) {

return <h2>Loading...</h2>;

}

return (

<div style={{ textAlign: 'center', marginTop: '50px' }}>

<h2>

{person.name.title} {person.name.first} {person.name.last}

</h2>

<img src={person.picture.large} alt="User" />

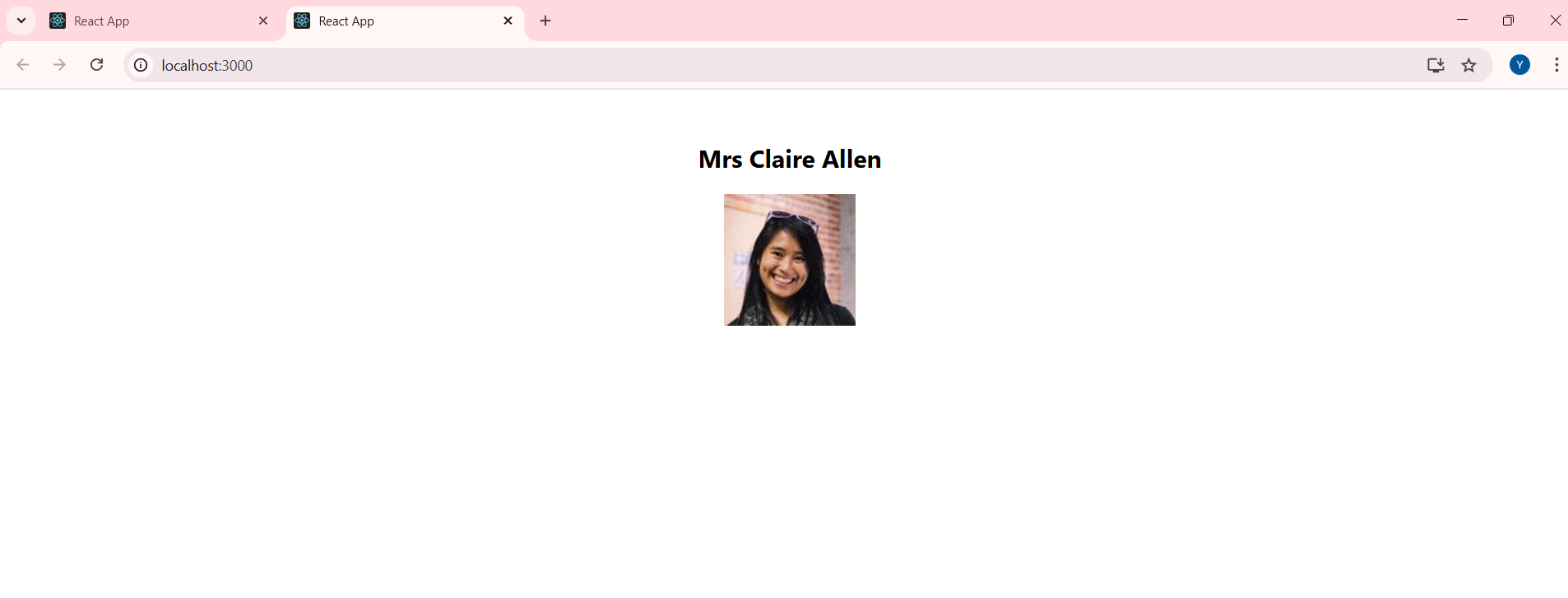
</div>

);

}

}

export default Getuser;



**REACT HOL 18**

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 unit testing the component to ensure it’s free of bugs.

**SetupTest.js**

import { TextEncoder, TextDecoder } from 'util';

if (typeof global.TextEncoder === 'undefined') {

global.TextEncoder = TextEncoder;

}

if (typeof global.TextDecoder === 'undefined') {

global.TextDecoder = TextDecoder;

}

import { configure } from 'enzyme';

import Adapter from 'enzyme-adapter-react-16';

configure({ adapter: new Adapter() });

**CohortDetails.test.js**

import React from 'react';

import { shallow, mount } from 'enzyme';

import CohortDetails from './CohortDetails';

import { CohortData } from './Cohort';

describe("Cohort Details Component", () => {

test("should create the component", () => {

const wrapper = shallow(<CohortDetails />);

expect(wrapper.exists()).toBe(true);

});

test("should initialize the props", () => {

const cohort = CohortData[0];

const wrapper = mount(<CohortDetails cohort={cohort} />);

expect(wrapper.props().cohort).toEqual(cohort);

});

test("should display cohort code in h3", () => {

const cohort = CohortData[0];

const wrapper = mount(<CohortDetails cohort={cohort} />);

expect(wrapper.find("h3").text()).toBe(cohort.cohortCode);

});

test("should always render same html", () => {

const cohort = CohortData[0];

const wrapper = shallow(<CohortDetails cohort={cohort} />);

expect(wrapper).toMatchSnapshot();

});

});

**CohortDetails**

function CohortDetails(props) {

return (

<div>

<h3>

{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;

**REACT HOL 19**

As an intern at OpenAI you are assigned the task of creating and testing a React application which will fetch and display a list of repository names for a given user.

**GitClient.js**

import axios from 'axios';

class GitClient {

static async getRepositories(username) {

try {

const response = await axios.get(`https://api.github.com/users/${username}/repos`);

return response.data.map(repo => repo.name);

} catch (error) {

console.error("Error fetching repositories", error);

return [];

}

}

}

export default GitClient;

App.js

import React, { useEffect, useState } from 'react';

import GitClient from './GitClient';

function App() {

const [repos, setRepos] = useState([]);

useEffect(() => {

async function fetchData() {

const data = await GitClient.getRepositories("techiesyed");

setRepos(data);

}

fetchData();

}, []);

return (

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

<h1>Repositories for techiesyed</h1>

<ul>

{repos.map((repo, index) => (

<li key={index}>{repo}</li>

))}

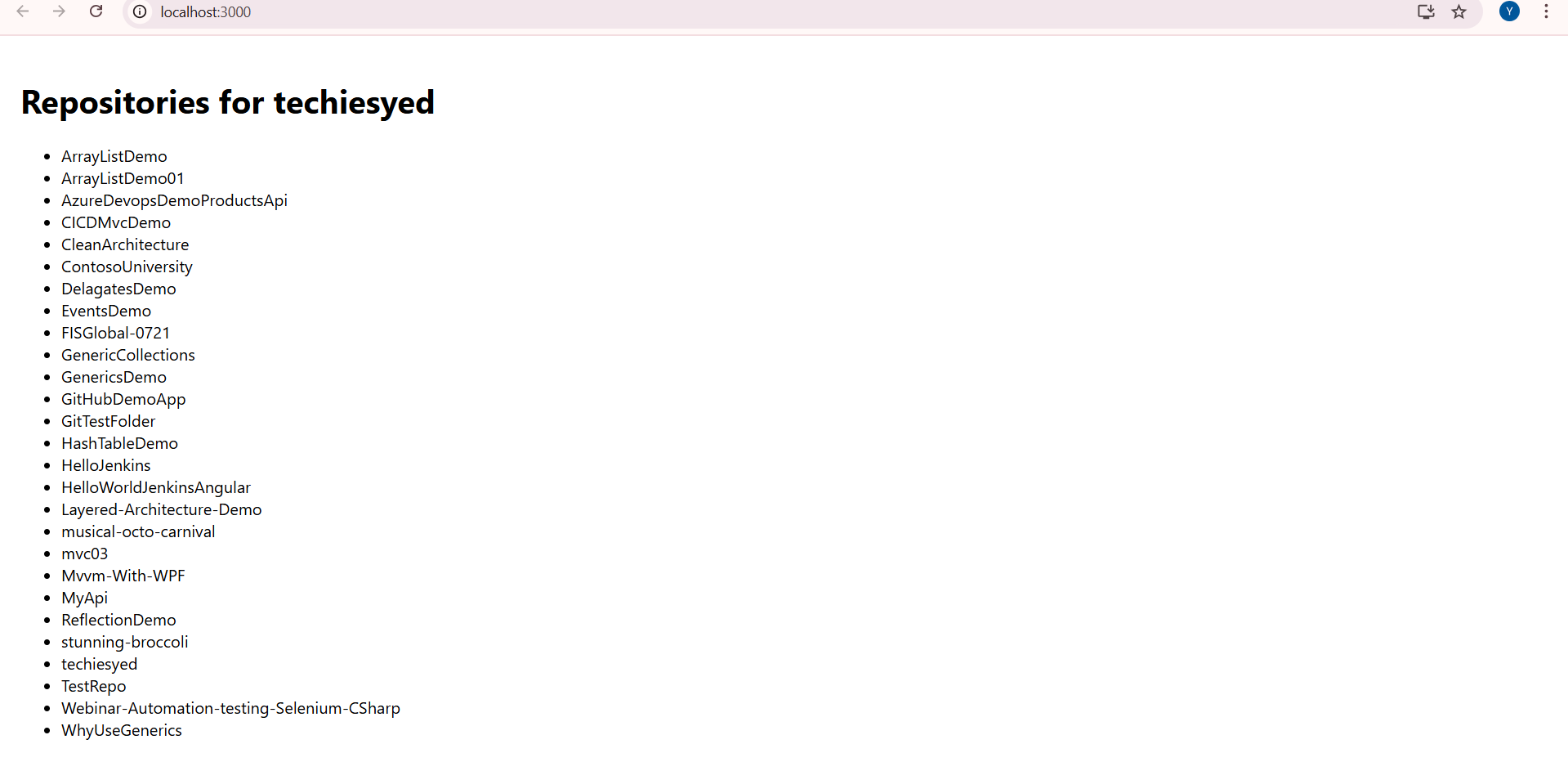
</ul>

</div>

);

}

export default App;



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_