WEEK 7

[ReactJs]

Exercise 1: Create a React Application named “cricketapp” with the following components

**Steps:**

1.npm create-react-app cricketapp

2. The files we should create are:

* src/ListofPlayers.js
* src/Scorebelow70.js
* src/IndianPlayers.js

3.source code for **“ListofPlayers.js”**

import React from 'react';

function ListofPlayers(props) {

    return (

        <div>

            {props.players.map((item) => (

                <div key={item.name}>

                    <li>

                        Mr. {item.name} <span>{item.score}</span>

                    </li>

                </div>

            ))}

        </div>

    );

}

export default ListofPlayers;

**4.Scorebelow70.js:**

Source code:

import React from 'react';

function Scorebelow70(props) {

    // First, filter the players who have a score less than 70

    const players70 = props.players.filter(item => item.score < 70);

    // Then, map over the new filtered array to display them

    return (

        <div>

            {players70.map((item) => (

                <div key={item.name}>

                    <li>

                        Mr. {item.name} <span>{item.score}</span>

                    </li>

                </div>

            ))}

        </div>

    );

}

export default Scorebelow70;

**5.IndianPlayers.js**

import React from 'react';

const T20Players = ['First Player', 'Second Player', 'Third Player'];

const RanjiTrophyPlayers = ['Fourth Player', 'Fifth Player', 'Sixth Player'];

export const IndianPlayers = [...T20Players, ...RanjiTrophyPlayers];

export const IndianTeam = {

    first: 'Sachin1',

    second: 'Dhoni2',

    third: 'Virat3',

    fourth: 'Rohit4',

    fifth: 'Yuvaraj5',

    sixth: 'Raina6'

};

export function OddPlayers({ first, third, fifth }) {

    return (

        <div>

            <li>First : {first}</li>

            <li>Third : {third}</li>

            <li>Fifth : {fifth}</li>

        </div>

    )

}

export function EvenPlayers({ second, fourth, sixth }) {

    return (

        <div>

            <li>Second : {second}</li>

            <li>Fourth : {fourth}</li>

            <li>Sixth : {sixth}</li>

        </div>

    )

}

export function ListofindianPlayers({ IndianPlayers }) {

    return (

        <div>

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

                <li key={index}>Mr. {player}</li>

            ))}

        </div>

    )

}

**6.App.js**

import './App.css';

import ListofPlayers from './ListofPlayers';

import Scorebelow70 from './Scorebelow70';

import { IndianPlayers, IndianTeam, OddPlayers, EvenPlayers, ListofindianPlayers } from './IndianPlayers';

function App() {

  // Data for the first part of the assignment

  const players = [

    { name: 'Jack', score: 50 },

    { name: 'Michael', score: 70 },

    { name: 'John', score: 40 },

    { name: 'Amy', score: 25 },

    { name: 'Elisabeth', score: 61 },

    { name: 'Sachin', score: 95 },

    { name: 'Dhoni', score: 100 },

    { name: 'Virat', score: 84 },

    { name: 'Jadeja', score: 64 },

    { name: 'Rama', score: 75 },

    { name: 'Rohit', score: 80 },

  ];

  // Set the flag here. Change to 'false' to see the other output.

  var flag = false;

  if (flag === true) {

    return (

      <div className="App">

        <h1>List of Players</h1>

        <ListofPlayers players={players} />

        <hr />

        <h1>List of Players having Scores Less than 70</h1>

        <Scorebelow70 players={players} />

      </div>

    );

  } else {

    return (

      <div className="App">

        <div>

          <h1>Indian Team</h1>

          <h2>Odd Players</h2>

          <OddPlayers {...IndianTeam} />

          <hr />

          <h2>Even Players</h2>

          <EvenPlayers {...IndianTeam} />

        </div>

        <hr />

        <div>

          <h1>List of Indian Players Merged:</h1>

          <ListofindianPlayers IndianPlayers={IndianPlayers} />

        </div>

      </div>

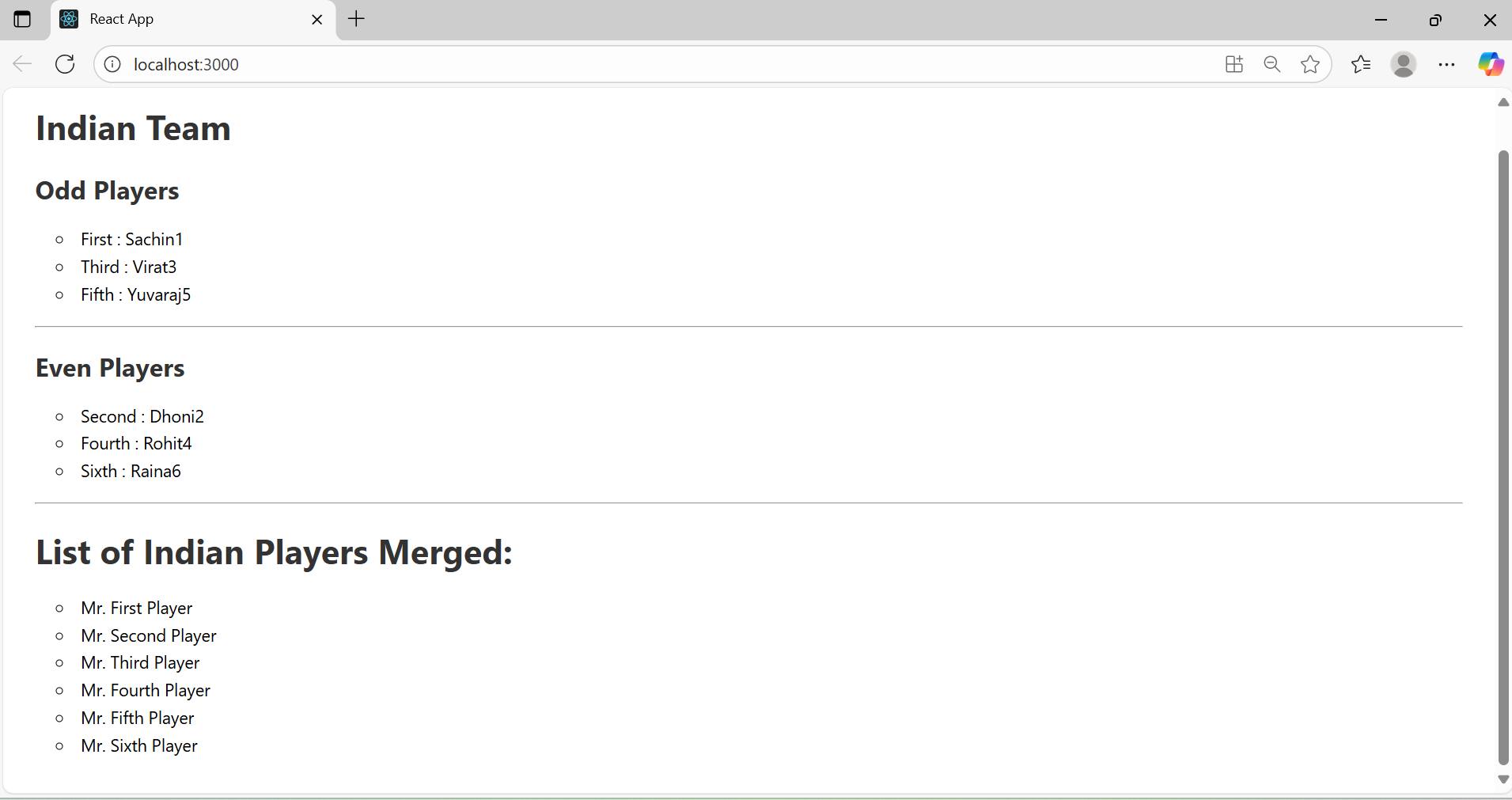
    );

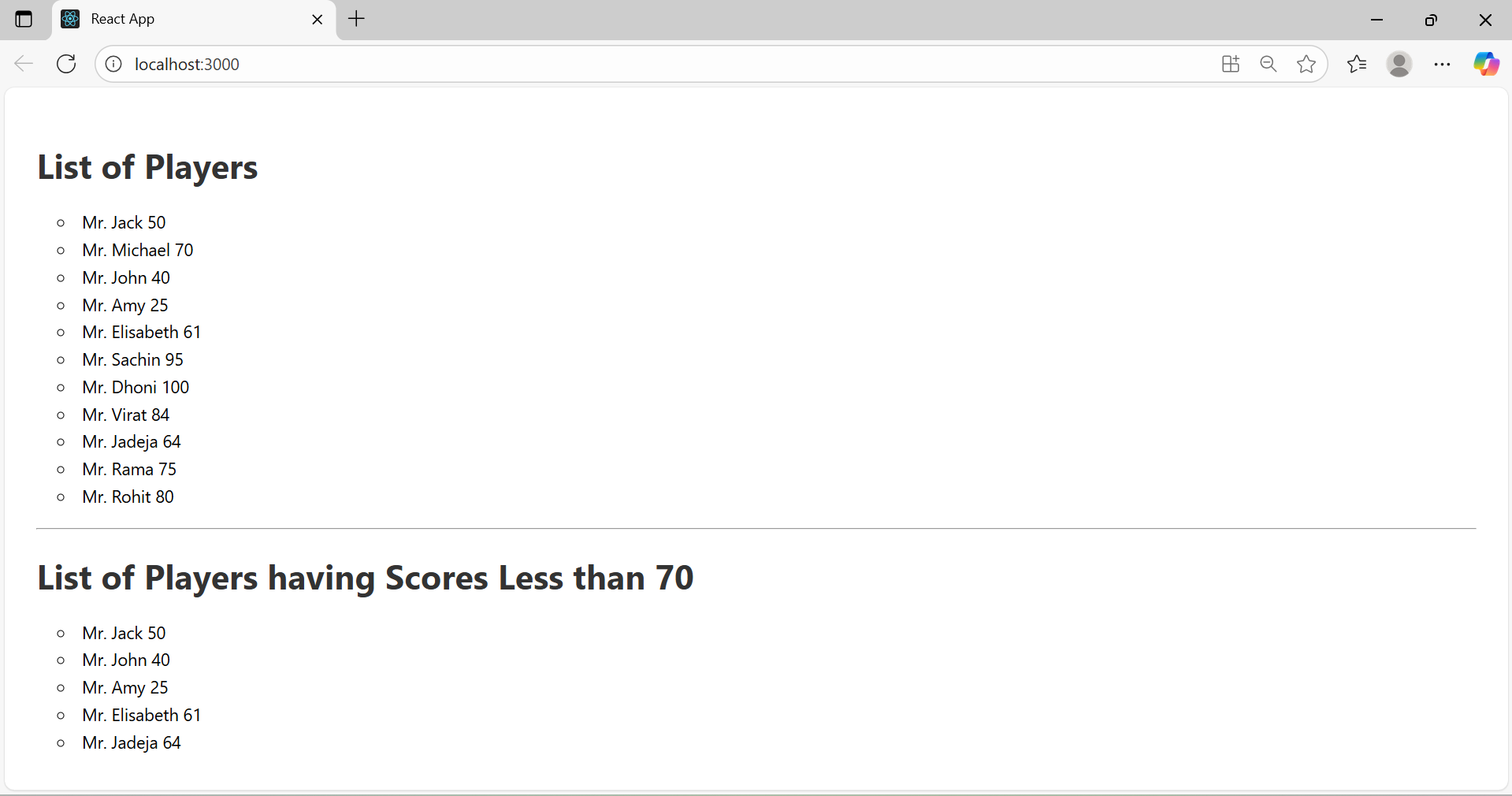
  }

}

export default App;

**7.Output:**





Exercise 2: Create a React Application named “officespacerentalapp” which uses React JSX to create elements, attributes and renders DOM to display the page.

1.npx create-react-app officespacerentalapp

2.App.js

Source code:

import './App.css';

// You can find an image online or use one from your project's public folder

function App() {

  // Create a list of Objects to loop through

  const officeSpaces = [

    {

      name: 'DBS',

      rent: 50000,

      address: 'Chennai',

      image: 'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSwXpmEP7v01sVej2QVE0227YSrHi-39OhkZg&s'

    },

    {

      name: 'WeWork',

      rent: 75000,

      address: 'Bengaluru',

      image: 'https://media.istockphoto.com/id/184962061/photo/business-towers.jpg?s=612x612&w=0&k=20&c=gLQLQ9lnfW6OnJVe39r516vbZYupOoEPl7P\_22Un6EM='

    },

    {

      name: 'CoWrks',

      rent: 45000,

      address: 'Hyderabad',

      image: 'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSblZ9YNv6gwVyunqGwUw2mrZq9G1Gu8FhRfQ&s'

    }

  ];

  const pageHeading = 'Office Space, at Affordable Range';

  return (

    <div className="App">

      {/\* Loop through the office space items to display more data \*/}

      {officeSpaces.map((item, index) => {

        // Create an element to display the image attribute

        const imageElement = <img src={item.image} alt={item.name} className="office-image" />;

        // Determine the CSS class for the rent color

        const rentColorClass = item.rent < 60000 ? 'rent-red' : 'rent-green';

        return (

          // Use a key for each item in the list for React's optimization

          <div className="office-card" key={index}>

            {/\* Create an element to display the heading of the page \*/}

            <h1>{pageHeading}</h1>

            {/\* Attribute to display the image of the office space \*/}

            {imageElement}

            {/\* Create an object of office to display the details \*/}

            <h3>Name: {item.name}</h3>

            {/\* Apply CSS to display the color of the Rent \*/}

            <h3 className={rentColorClass}>Rent: Rs. {item.rent}</h3>

            <h3>Address: {item.address}</h3>

          </div>

        );

      })}

    </div>

  );

}

export default App;

**2.App.css**

**Source code:**

.App {

  /\* Make the container a flexbox \*/

  display: flex;

  /\* Stack the cards vertically \*/

  flex-direction: column;

  /\* Center the cards horizontally \*/

  align-items: center;

  /\* Add some padding at the top and bottom \*/

  padding: 40px 20px;

  /\* Optional: Make sure the app takes at least the full screen height \*/

  min-height: 100vh;

  font-family: sans-serif;

  background-color: #f7f7f7;

  /\* Added a light gray background \*/

}

.office-card {

  border: 1px solid #ddd;

  border-radius: 8px;

  padding: 20px;

  margin-bottom: 20px;

  max-width: 500px;

  width: 100%;

  /\* Ensure it takes up the max-width \*/

  background-color: white;

  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

  /\* Add a subtle shadow \*/

  text-align: left;

  /\* Keep text inside the card aligned left \*/

}

.office-image {

  width: 100%;

  height: auto;

  border-radius: 4px;

  margin-bottom: 15px;

  /\* Add space below the image \*/

}

/\* To apply Css, Display the color of the Rent \*/

.rent-red {

  color: red;

  font-weight: bold;

}

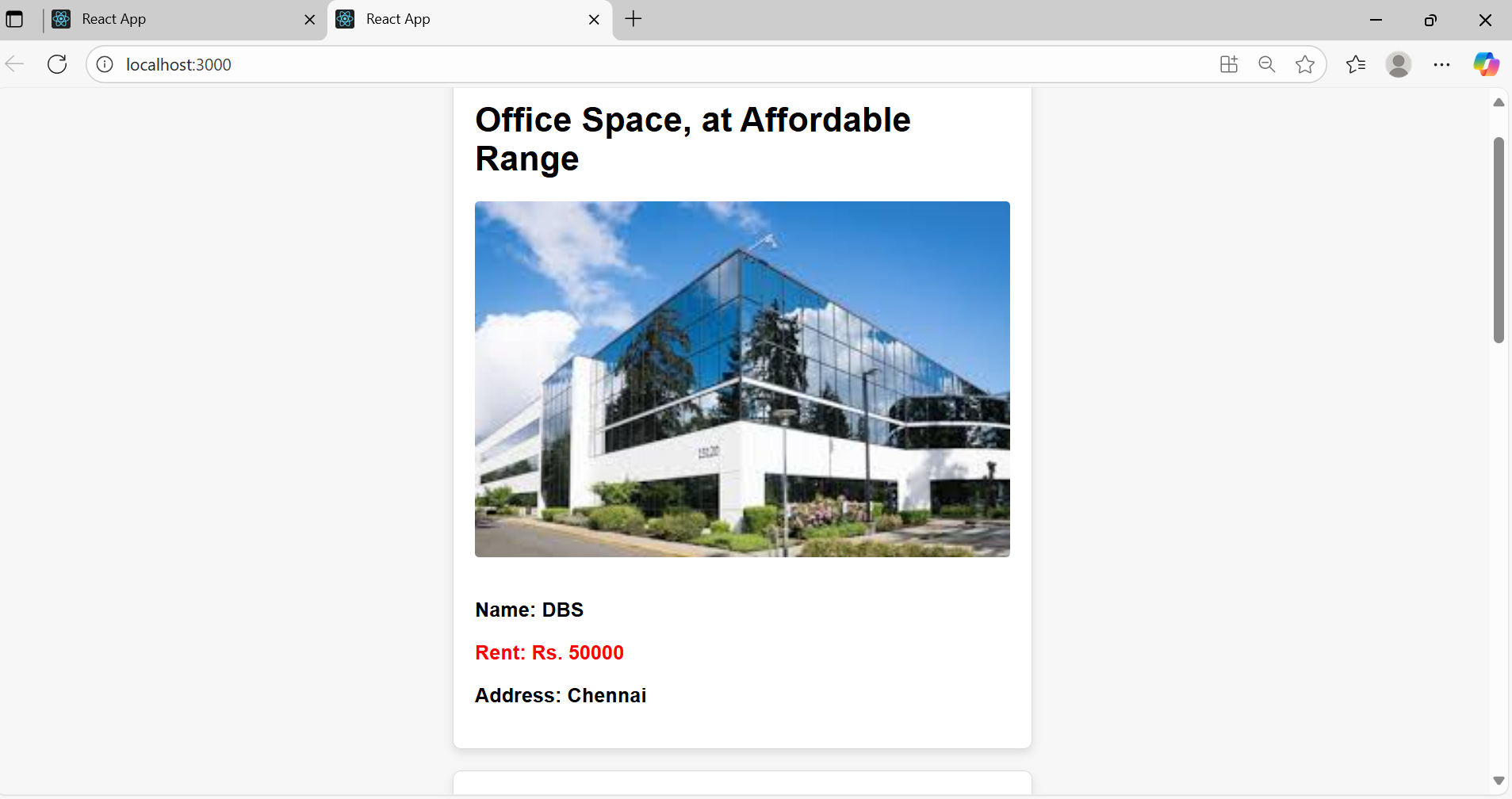
.rent-green {

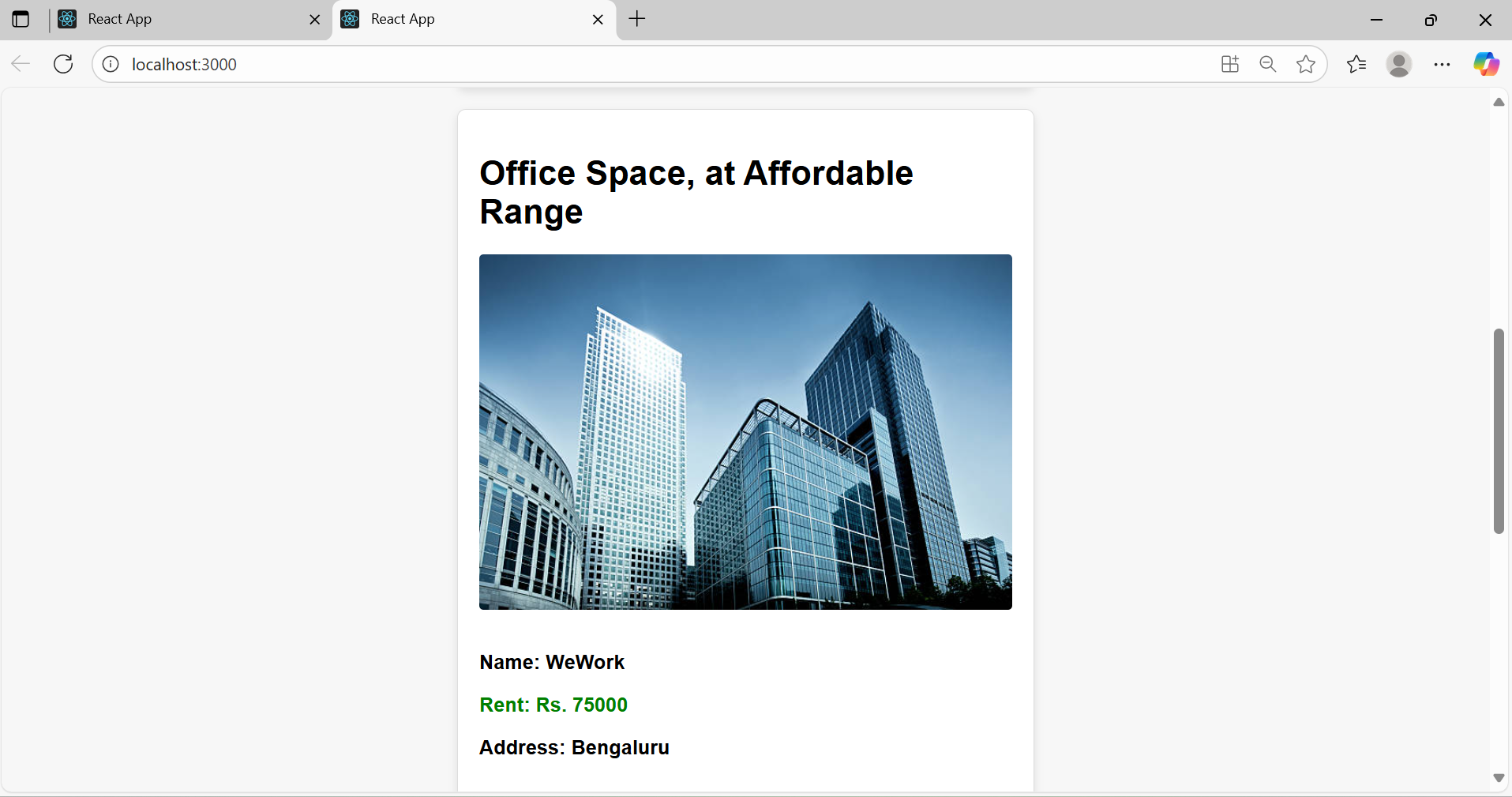
  color: green;

  font-weight: bold;

}

**3.Output:**





Exercise 3: Create a React Application “eventexamplesapp” to handle various events of the form elements in HTML.

1. Create “Increment” button to increase the value of the counter and “Decrement” button to decrease the value of the counter. The “Increase” button should invoke multiple methods.
   1. To increment the value
   2. Say Hello followed by a static message.

🡪 npx create-react-app eventexamplesapp

**1.App.js**

import React, { useState } from 'react';

import './App.css';

import CurrencyConvertor from './CurrencyConvertor'; // We will create this next

function App() {

// Use the useState hook to manage the counter's state

const [count, setCount] = useState(5); // Initial value set to 5 as in the screenshot

// 1a. Method to increment the value

const increment = () => {

setCount(prevCount => prevCount + 1);

};

// 1a. Method to say hello

const sayHello = () => {

alert('hello Member1');

};

// Method to handle the "Increment" button click, invoking multiple functions

const handleIncrementClick = () => {

increment();

sayHello();

};

// Method to decrement the value

const decrement = () => {

setCount(prevCount => prevCount - 1);

};

// 2. Function that takes an argument

const sayWelcome = (message) => {

alert(message);

};

// 3. Function for the synthetic event

const handlePress = (event) => {

// 'event' is the synthetic event object provided by React

console.log(event); // You can inspect the event in the browser console

alert('I was clicked');

};

return (

<div className="App">

<div className="counter-section">

<h2>Counter</h2>

{/\* Display the current count \*/}

<h1>{count}</h1>

{/\* The onClick event handlers \*/}

<button onClick={handleIncrementClick}>Increment</button>

<button onClick={decrement}>Decrement</button>

{/\* We use an arrow function to pass an argument to our event handler \*/}

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

<button onClick={handlePress}>Click on me</button>

</div>

<hr />

{/\* We will render the CurrencyConvertor component here \*/}

<CurrencyConvertor />

</div>

);

}

export default App;

**2.CurrencyConvertor.js**

import React, { useState } from 'react';

// Assuming a static conversion rate for the assignment

const EURO\_TO\_INR\_RATE = 80;

function CurrencyConvertor() {

// State for the amount and currency input fields

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

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

// Handles the form submission

const handleSubmit = (event) => {

// Prevent the default form submission behavior which reloads the page

event.preventDefault();

// Check if the currency is 'Euro' (case-insensitive)

if (currency.toLowerCase() === 'euro') {

const inputAmount = parseFloat(amount);

if (!isNaN(inputAmount)) {

const convertedAmount = inputAmount \* EURO\_TO\_INR\_RATE;

alert(`Converting to Euro Amount is ${convertedAmount}`);

} else {

alert('Please enter a valid number for the amount.');

}

} else {

alert('This converter only handles Euro to INR conversion.');

}

};

return (

<div className="converter-section">

<h2>Currency Convertor!!!</h2>

{/\* The onSubmit event is handled on the form element \*/}

<form onSubmit={handleSubmit}>

<div>

<label>Amount: </label>

<input

type="text"

value={amount}

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

/>

</div>

<div>

<label>Currency: </label>

<input

type="text"

value={currency}

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

/>

</div>

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

</form>

</div>

);

}

export default CurrencyConvertor;

**3. App.css**

.App {

display: flex;

flex-direction: column;

align-items: center;

text-align: center;

font-family: sans-serif;

padding-top: 40px;

}

.counter-section,

.converter-section {

border: 1px solid #ccc;

border-radius: 8px;

padding: 20px;

margin: 20px;

width: 300px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

}

h2 {

color: #007BFF;

}

button {

margin: 5px;

padding: 10px 15px;

border: none;

border-radius: 4px;

background-color: #007BFF;

color: white;

cursor: pointer;

font-size: 14px;

}

button:hover {

background-color: #0056b3;

}

hr {

width: 80%;

margin: 30px 0;

border: 1px solid #eee;

}

input {

padding: 8px;

margin: 5px 0 10px 0;

border: 1px solid #ccc;

border-radius: 4px;

}

label {

margin-right: 10px;

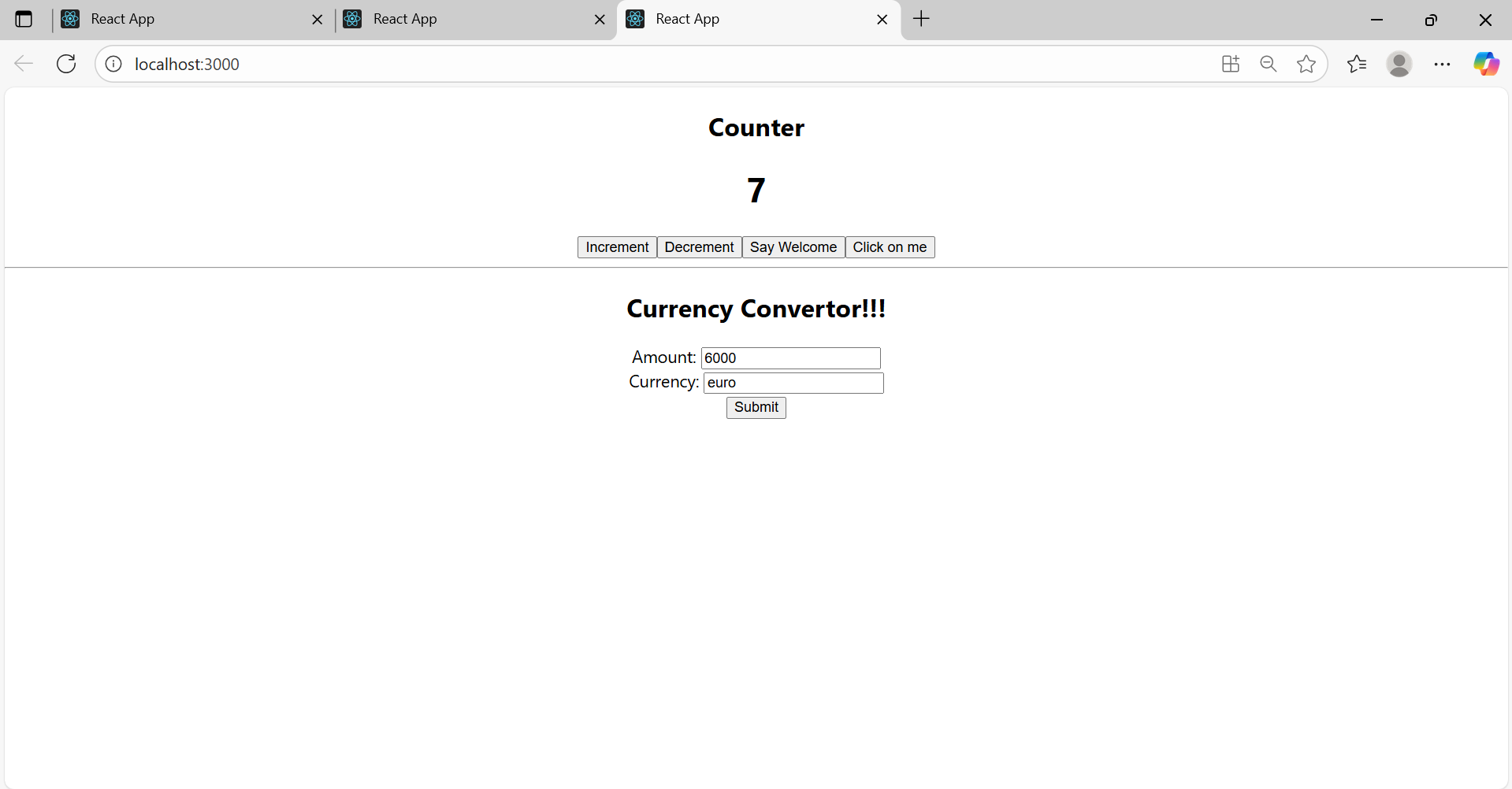
}

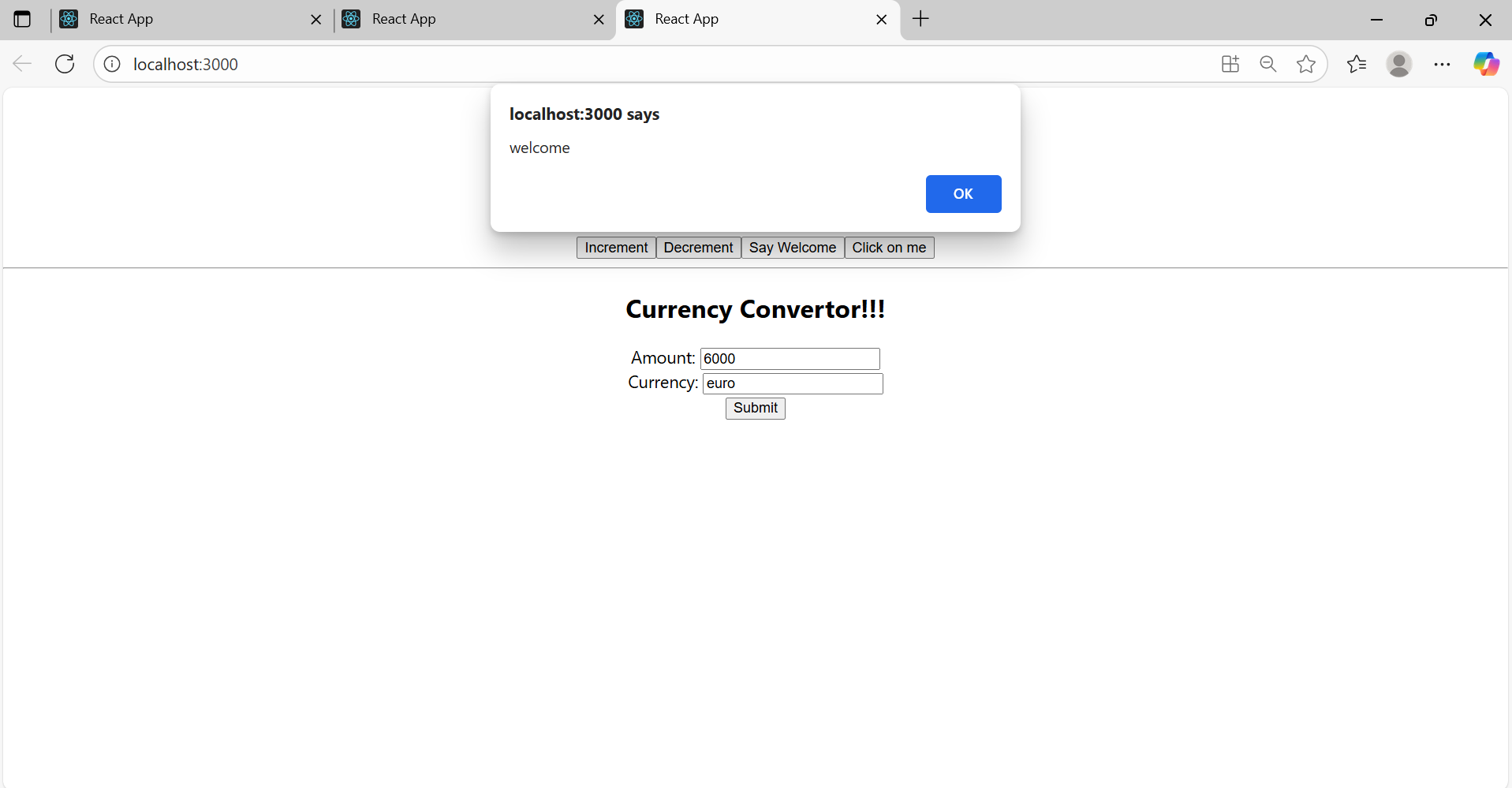
form div {

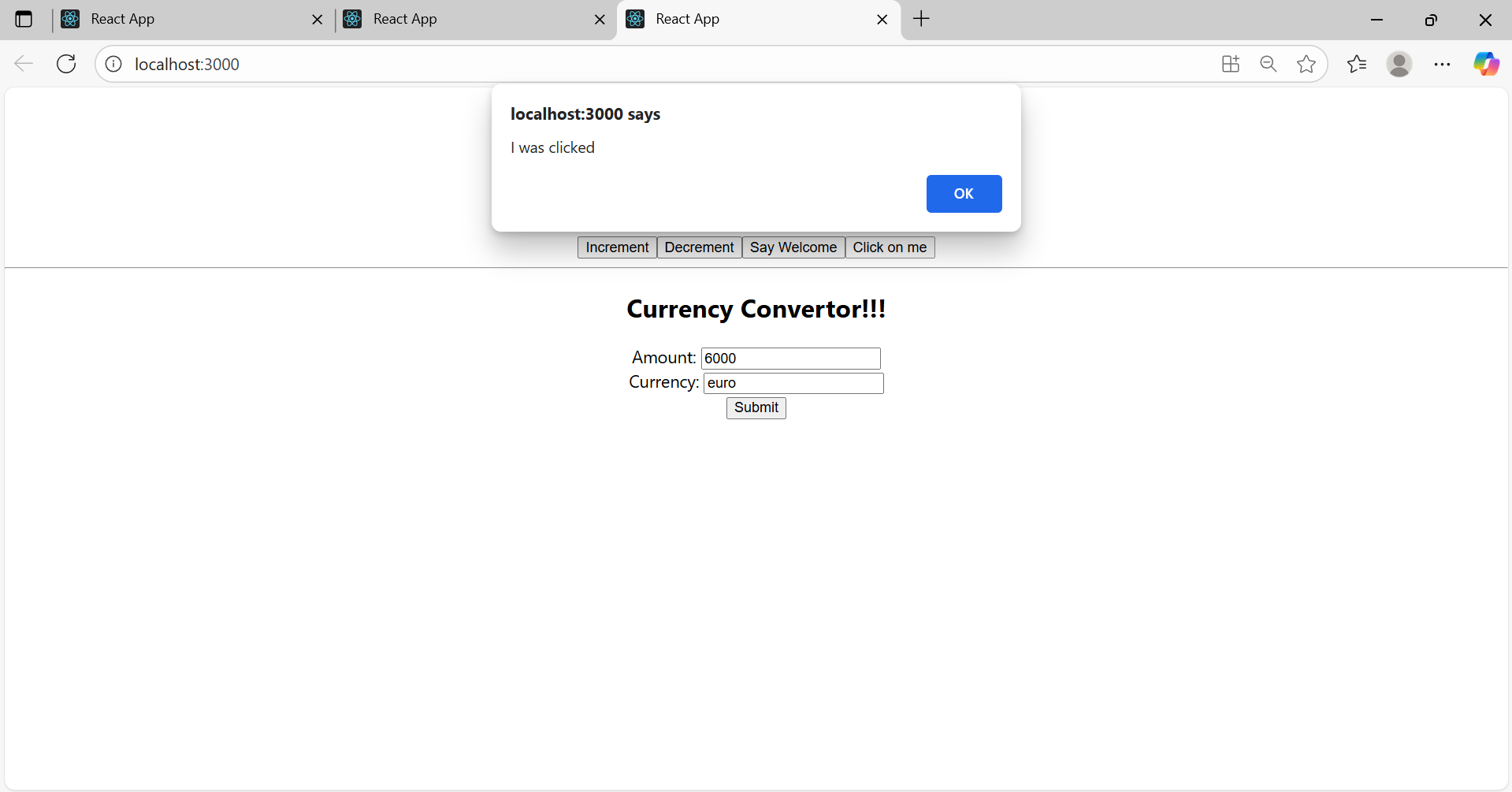
margin-bottom: 10px;

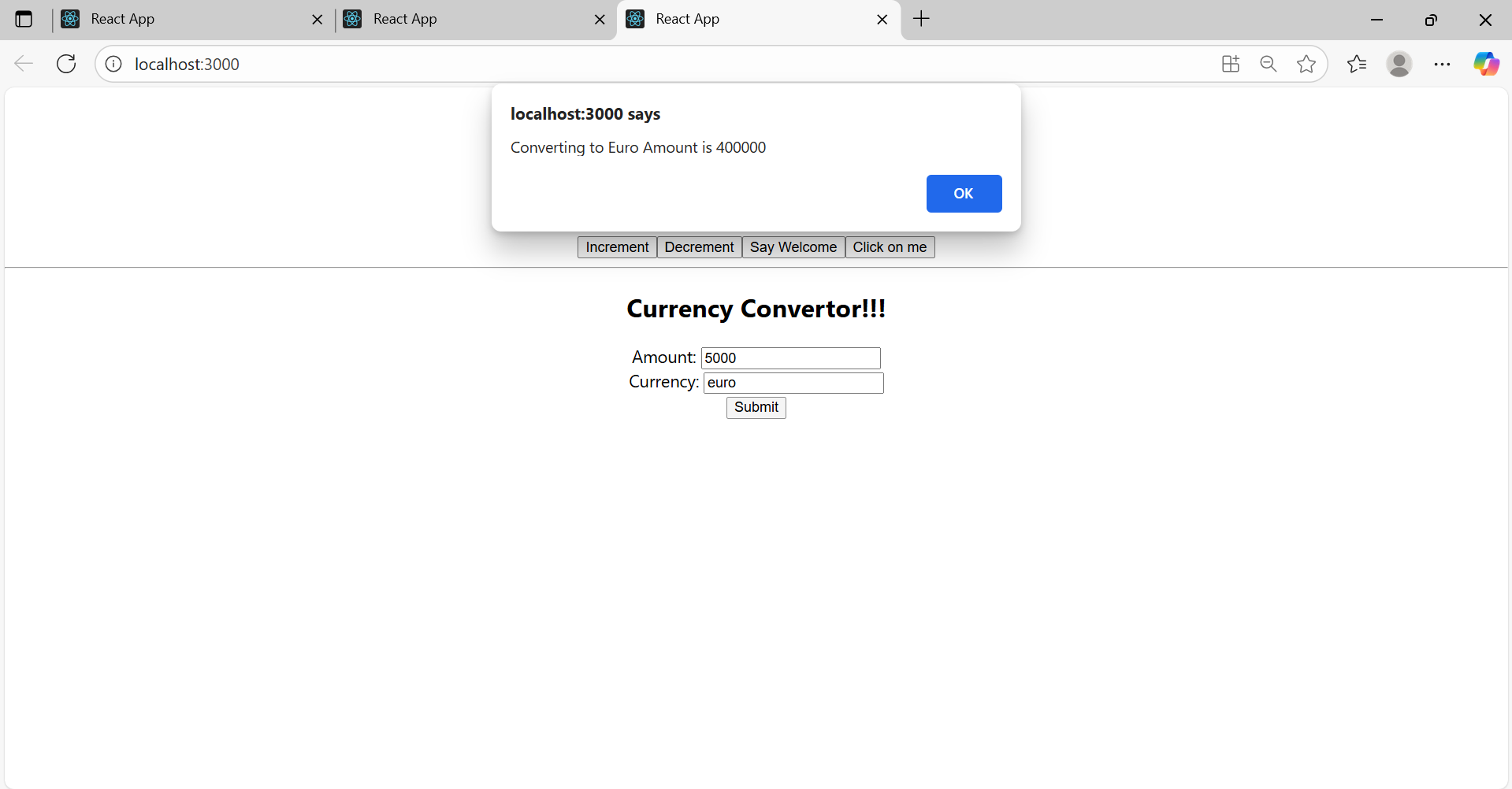
}

**Output:**

****

****

****

****

**Exercise 4:** 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.

**🡪** **npx create-react-app ticketbookingapp**

**1.File: src/Greeting.js**

import React from 'react';

function UserGreeting(props) {

return <h1>Welcome back</h1>;

}

function GuestGreeting(props) {

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

}

function Greeting(props) {

const isLoggedIn = props.isLoggedIn;

if (isLoggedIn) {

return <UserGreeting />;

}

return <GuestGreeting />;

}

export default Greeting;

1. **File: src/UserPage.js**

import React from 'react';

function UserPage() {

return (

<div>

<h2>Available Flights</h2>

<p>Flight 101: Delhi to Mumbai <button>Book Now</button></p>

<p>Flight 202: Chennai to Bengaluru <button>Book Now</button></p>

<p>Flight 303: Kolkata to Hyderabad <button>Book Now</button></p>

</div>

);

}

export default UserPage;

**3.File: src/GuestPage.js**

import React from 'react';

function GuestPage() {

return (

<div>

<h2>Available Flights</h2>

<p>Flight 101: Delhi to Mumbai</p>

<p>Flight 202: Chennai to Bengaluru</p>

<p>Flight 303: Kolkata to Hyderabad</p>

<p style={{ color: 'red', marginTop: '15px' }}>

<i>You must be logged in to book a ticket.</i>

</p>

</div>

);

}

export default GuestPage;

**4.File: src/LoginControl.js**

import React, { useState } from 'react';

import Greeting from './Greeting';

import UserPage from './UserPage';

import GuestPage from './GuestPage';

// Simple button components as shown in your hints

function LoginButton(props) {

return <button onClick={props.onClick}>Login</button>;

}

function LogoutButton(props) {

return <button onClick={props.onClick}>Logout</button>;

}

function LoginControl() {

// The core state: is the user logged in or not?

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

// Event handlers to update the state

const handleLoginClick = () => {

setIsLoggedIn(true);

};

const handleLogoutClick = () => {

setIsLoggedIn(false);

};

// Declare variables for the button and the page content

let button;

let page;

// Use an if/else statement to determine which components to show

if (isLoggedIn) {

button = <LogoutButton onClick={handleLogoutClick} />;

page = <UserPage />;

} else {

button = <LoginButton onClick={handleLoginClick} />;

page = <GuestPage />;

}

return (

<div>

{/\* The Greeting component also depends on the isLoggedIn state \*/}

<Greeting isLoggedIn={isLoggedIn} />

{/\* Render the correct button \*/}

{button}

<hr />

{/\* Render the correct page content \*/}

{page}

</div>

);

}

export default LoginControl;

**5.File: src/App.js**

import React from 'react';

import './App.css';

import LoginControl from './LoginControl';

function App() {

return (

<div className="App">

<div className="ticket-container">

<LoginControl />

</div>

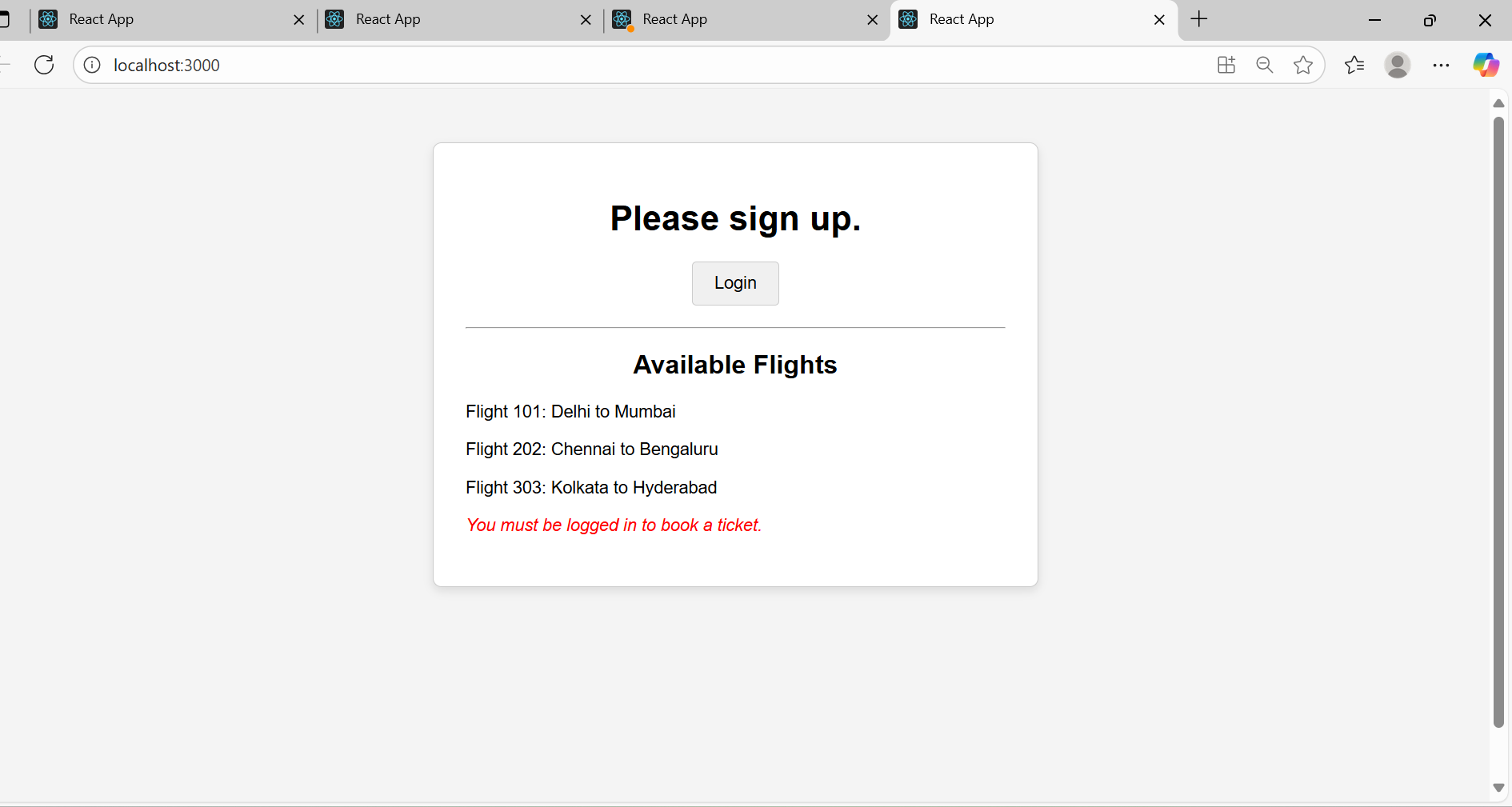
</div>

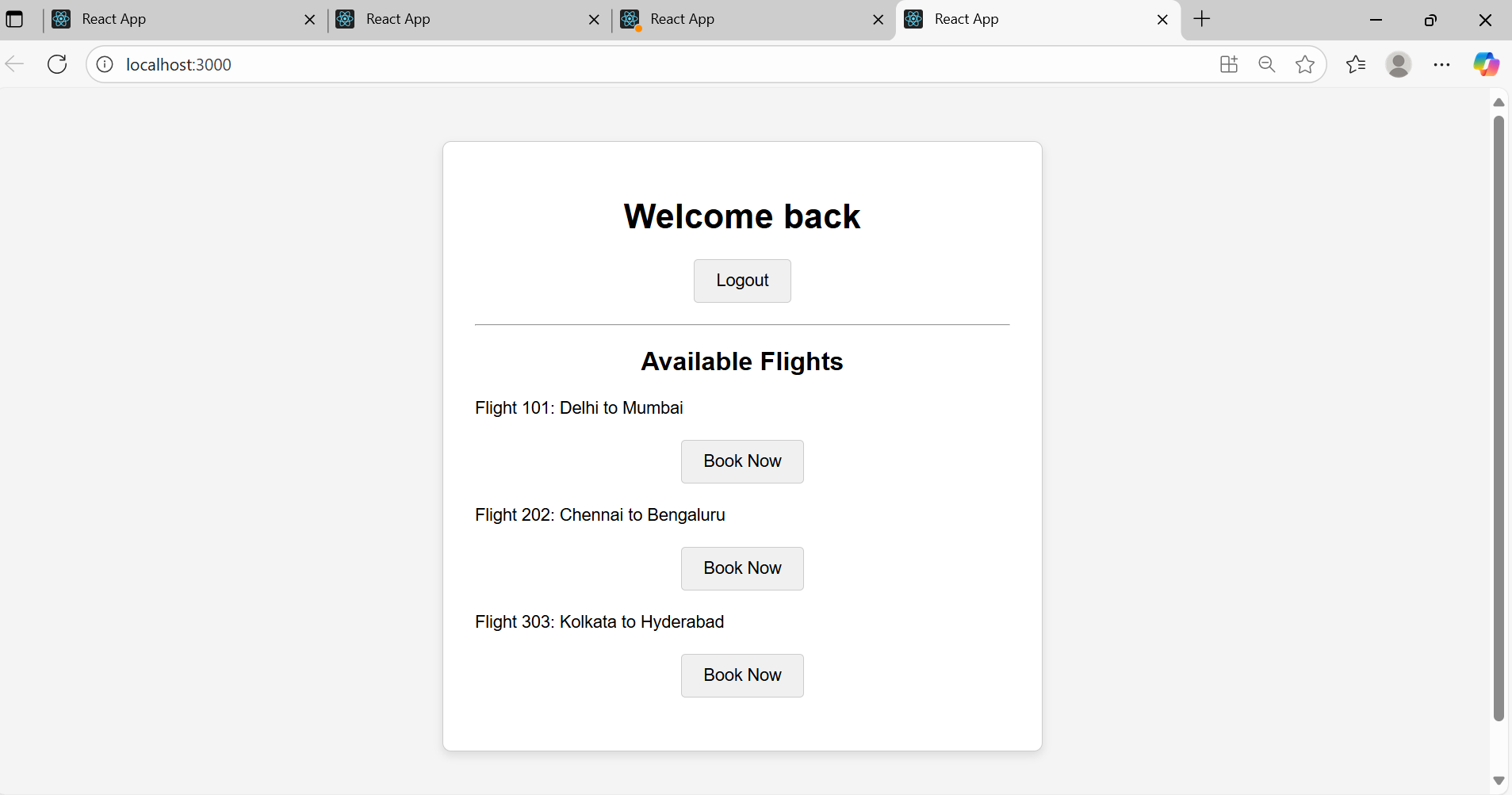
);

}

export default App;

**Output:**

****



**Exercise 5:** Create a React App named “bloggerapp” in with 3 components.

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

Implement this with as many ways possible of Conditional Rendering.

**File: src/BookDetails.js**

import React from 'react';

function BookDetails() {

return (

<div>

<h2>Book Details</h2>

<div>

<h3>Master React</h3>

<p>670</p>

</div>

<div>

<h3>Deep Dive into Angular 11</h3>

<p>800</p>

</div>

<div>

<h3>Mongo Essentials</h3>

<p>450</p>

</div>

</div>

);

}

export default BookDetails;

**2.src/BlogDetails.js**

import React from 'react';

function BlogDetails() {

return (

<div>

<h2>Blog Details</h2>

<div>

<h3>React Learning</h3>

<p>Stephen Biz</p>

<p>Welcome to learning React!</p>

</div>

<div>

<h3>Installation</h3>

<p>Schewzdenier</p>

<p>You can install React from npm.</p>

</div>

</div>

);

}

export default BlogDetails;

**3.src/CourseDetails.js**

import React from 'react';

function CourseDetails() {

return (

<div>

<h2>Course Details</h2>

<div>

<h3>Angular</h3>

<p>4/5/2021</p>

</div>

<div>

<h3>React</h3>

<p>6/3/2020</p>

</div>

</div>

);

}

export default CourseDetails;

4.App.js

import React, { useState } from 'react';

import './App.css';

import BookDetails from './BookDetails';

import BlogDetails from './BlogDetails';

import CourseDetails from './CourseDetails';

function App() {

// useState hook to manage which component to display.

// It can be 'courses', 'books', or 'blogs'.

const [activeView, setActiveView] = useState('courses'); // Default view

// --- Method 1: Using if/else statement ---

// This is a very clear and readable way to handle complex conditional logic.

const renderContentWithIfElse = () => {

if (activeView === 'courses') {

return <CourseDetails />;

} else if (activeView === 'books') {

return <BookDetails />;

} else if (activeView === 'blogs') {

return <BlogDetails />;

} else {

return <p>Please select a view.</p>;

}

};

return (

<div className="App">

<div className="controls">

<h1>Select a View</h1>

<button onClick={() => setActiveView('courses')}>Show Courses</button>

<button onClick={() => setActiveView('books')}>Show Books</button>

<button onClick={() => setActiveView('blogs')}>Show Blogs</button>

</div>

<div className="content-display">

{/\* --- Method 2: Using Logical && Operator --- \*/}

{/\* This is a concise way to render a component ONLY if a condition is true. \*/}

{/\* It's great for simple "show/hide" logic. \*/}

<div className="content-section">

<h2>Rendered with Logical &&</h2>

{activeView === 'courses' && <CourseDetails />}

{activeView === 'books' && <BookDetails />}

{activeView === 'blogs' && <BlogDetails />}

</div>

{/\* --- Method 3: Using the Ternary Operator --- \*/}

{/\* This is useful for choosing between TWO options. We can nest them, \*/}

{/\* but it can become hard to read. Here, we use it to pick one of the three. \*/}

<div className="content-section">

<h2>Rendered with Ternary Operator</h2>

{activeView === 'courses' ? (

<CourseDetails />

) : activeView === 'books' ? (

<BookDetails />

) : (

<BlogDetails />

)}

</div>

{/\* We can also call the function that uses the if/else logic \*/}

<div className="content-section">

<h2>Rendered with if/else Function</h2>

{renderContentWithIfElse()}

</div>

</div>

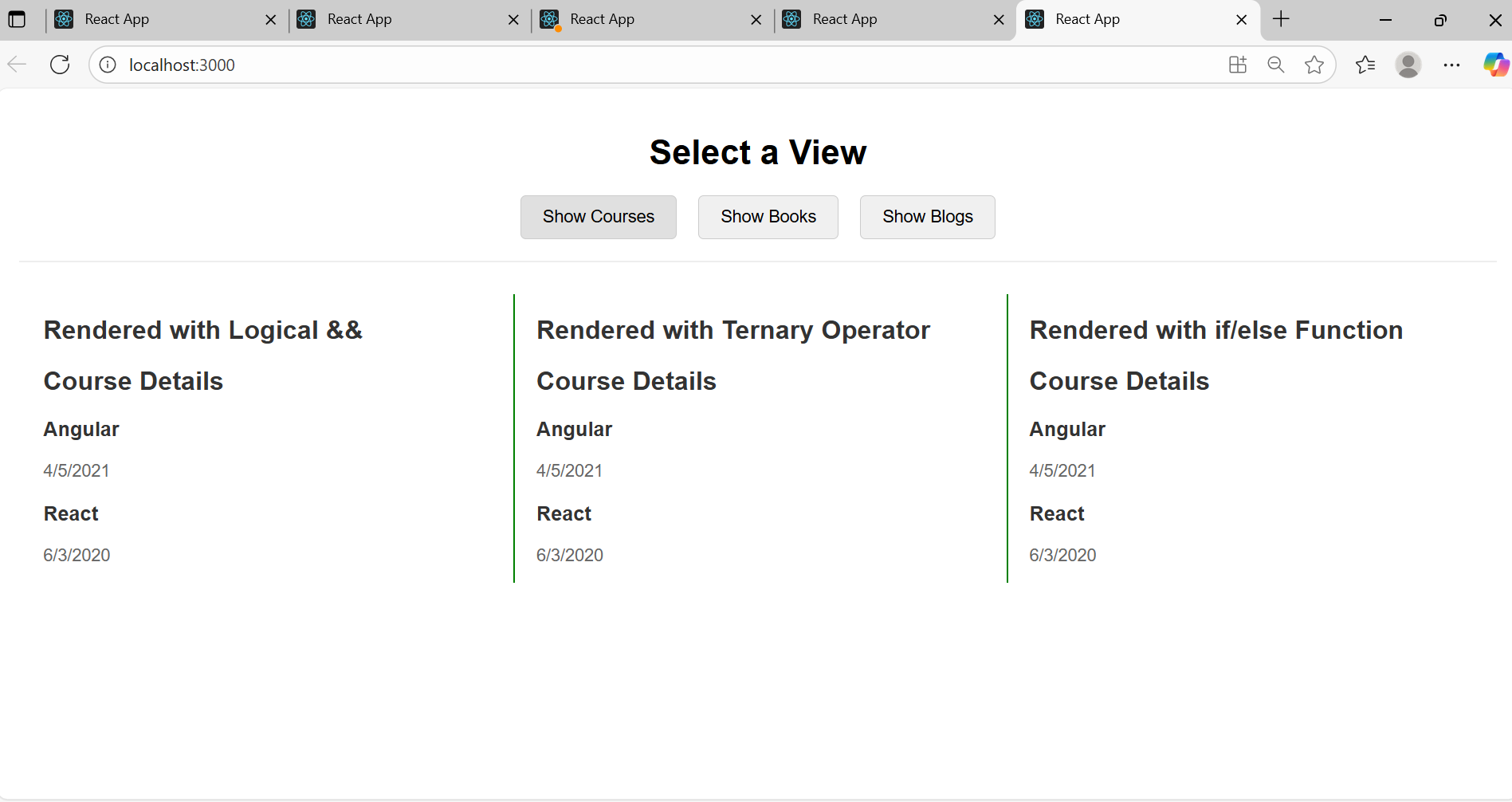
</div>

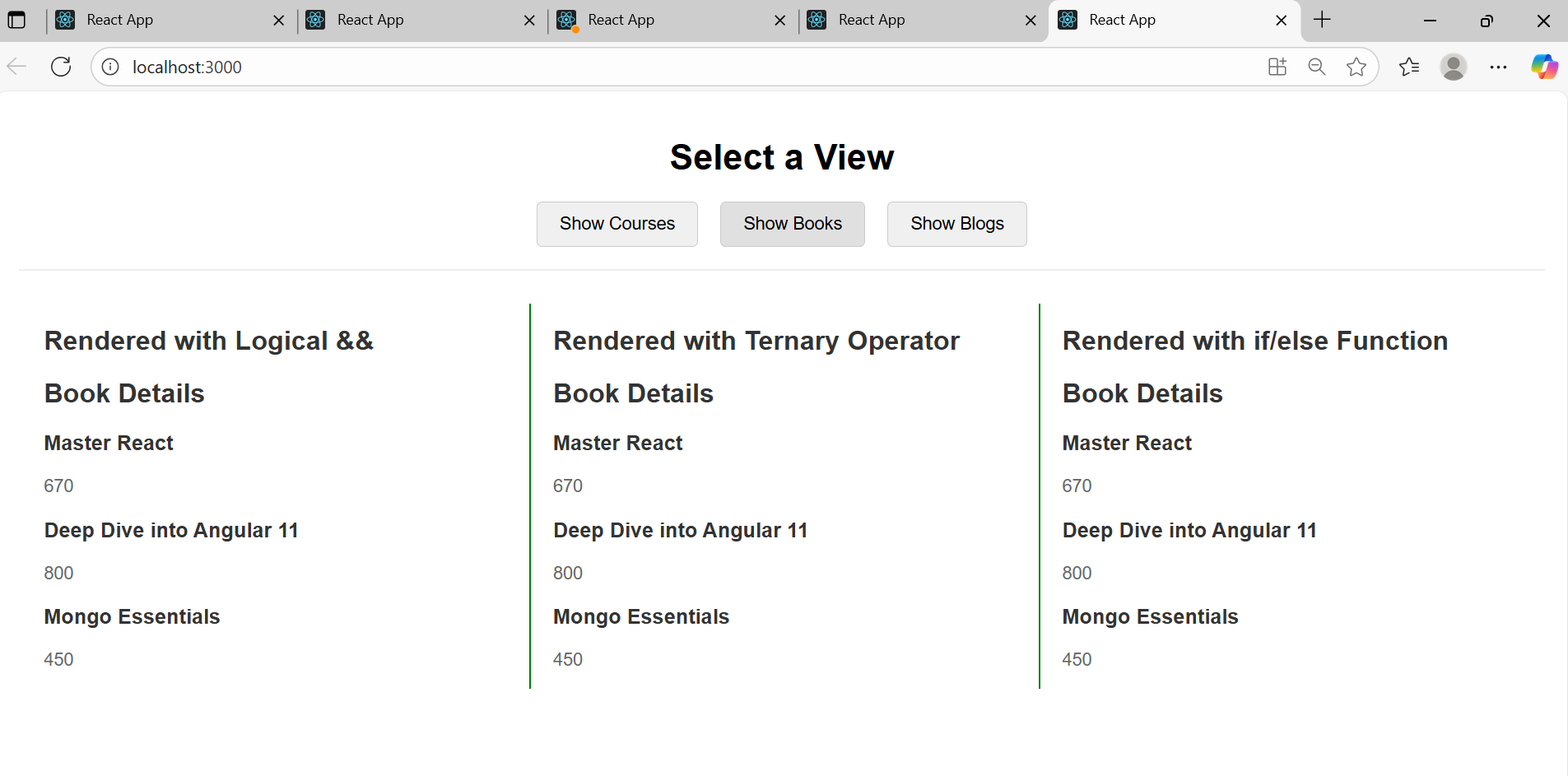
);

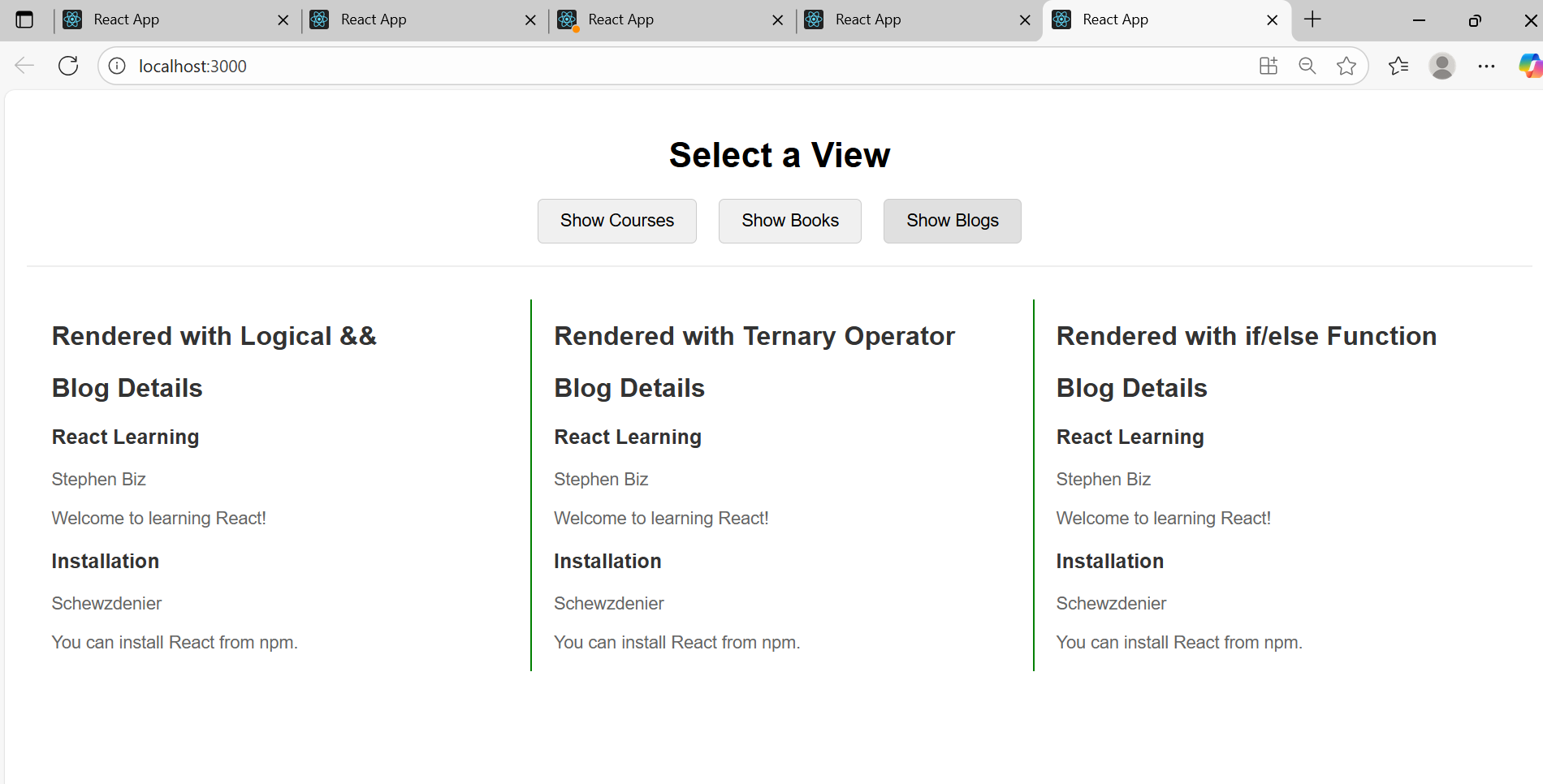
}

export default App;

**Output:**

****

****

****