**Cognizant - DN 4.0 I Deep Skilling**

**WEEK-7**

**ReactJS**

****Lab 9 –** Create a new React Application with the name “cricketapp”**

**Hands On:**

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
* Filter the players with scores below 70 using arrow functions of ES6.

1. IndianPlayers
   1. Display the Odd Team Player and Even Team players using the Destructuring features of ES6.
   2. Declare two arrays T20players and RanjiTrophy players and merge the two arrays and display them using the Merge feature of ES6

Display these two components in the same home page using a simple if else in the flag variable.

**SOLUTION :**

**CODE -:**

1. **src/components/IndianPlayers.jsx**

// IndianPlayers.jsx

import React from "react";

const IndianPlayers = () => {

// Suppose we have a base list of Indian players

const allPlayers = [

"Rohit Sharma",

"Virat Kohli",

"KL Rahul",

"Shubman Gill",

"Hardik Pandya",

"Ravindra Jadeja",

"Jasprit Bumrah",

"Shreyas Iyer",

"Ishan Kishan",

"Suryakumar Yadav",

"Yuzvendra Chahal",

];

// Split into odd/even index teams (0-based)

const oddTeam = allPlayers.filter((\_, i) => i % 2 === 0); // indices 0,2,...

const evenTeam = allPlayers.filter((\_, i) => i % 2 === 1); // indices 1,3,...

// Destructure to pick first and rest for each

const [firstOdd, ...otherOdd] = oddTeam;

const [firstEven, ...otherEven] = evenTeam;

// Two separate arrays to merge (T20 and Ranji players)

const T20players = ["Suryakumar Yadav", "Jasprit Bumrah", "Rohit Sharma"];

const RanjiTrophyPlayers = ["Player A", "Player B", "Player C"];

// Merge using ES6 spread

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

return (

<div style={{ padding: "1rem", fontFamily: "sans-serif" }}>

<h2>Indian Players (Odd/Even Teams)</h2>

<div>

<h3>Odd Team</h3>

<p>

<strong>First Odd Player:</strong> {firstOdd}

</p>

{otherOdd.length > 0 && (

<div>

<p>

<strong>Other Odd Players:</strong>

</p>

<ul>

{otherOdd.map((p, i) => (

<li key={i}>{p}</li>

))}

</ul>

</div>

)}

</div>

<div>

<h3>Even Team</h3>

<p>

<strong>First Even Player:</strong> {firstEven}

</p>

{otherEven.length > 0 && (

<div>

<p>

<strong>Other Even Players:</strong>

</p>

<ul>

{otherEven.map((p, i) => (

<li key={i}>{p}</li>

))}

</ul>

</div>

)}

</div>

<div>

<h2>Merged Players (T20 + Ranji Trophy)</h2>

<ul>

{mergedPlayers.map((p, i) => (

<li key={i}>{p}</li>

))}

</ul>

</div>

</div>

);

};

export default IndianPlayers;

1. **Src/components/LitsofPlayers.jsx**

// ListOfPlayers.jsx

import React from "react";

// Simple ES6 class to model a Player (demonstrating class fundamentals)

class Player {

constructor(name, score) {

this.name = name;

this.score = score;

}

}

const players = [

new Player("Virat Kohli", 85),

new Player("Rohit Sharma", 72),

new Player("Jasprit Bumrah", 65),

new Player("KL Rahul", 91),

new Player("Shubman Gill", 55),

new Player("Hardik Pandya", 77),

new Player("Ravindra Jadeja", 69),

new Player("Shreyas Iyer", 60),

new Player("Yuzvendra Chahal", 48),

new Player("Ishan Kishan", 82),

new Player("Suryakumar Yadav", 94),

];

const ListOfPlayers = () => {

// Filter players with score below 70 using arrow function

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

return (

<div style={{ padding: "1rem", fontFamily: "sans-serif" }}>

<h2>All Players (11)</h2>

<ul>

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

<li key={idx}>

{player.name} - Score: {player.score}

</li>

))}

</ul>

<h3>Players with score below 70</h3>

{lowScorers.length ? (

<ul>

{lowScorers.map((p, idx) => (

<li key={idx}>

{p.name} - {p.score}

</li>

))}

</ul>

) : (

<p>None</p>

)}

</div>

);

};

export default ListOfPlayers;

1. **App.js**

// App.js

import React from "react";

import ListOfPlayers from "./components/ListOfPlayers";

import IndianPlayers from "./components/IndianPlayers";

function App() {

// Toggle flag to show different component(s)

const flag = true; // change to false to switch

return (

<div>

<header style={{ background: "#282c34", padding: "1rem", color: "white" }}>

<h1 style={{ margin: 0 }}>cricketapp</h1>

</header>

<main>

<h2 style={{ marginLeft: "1rem" }}>ListOfPlayers</h2>

<ListOfPlayers />

<h2 style={{ marginLeft: "1rem" }}>IndianPlayers (odd/even + merged)</h2>

<IndianPlayers />

</main>

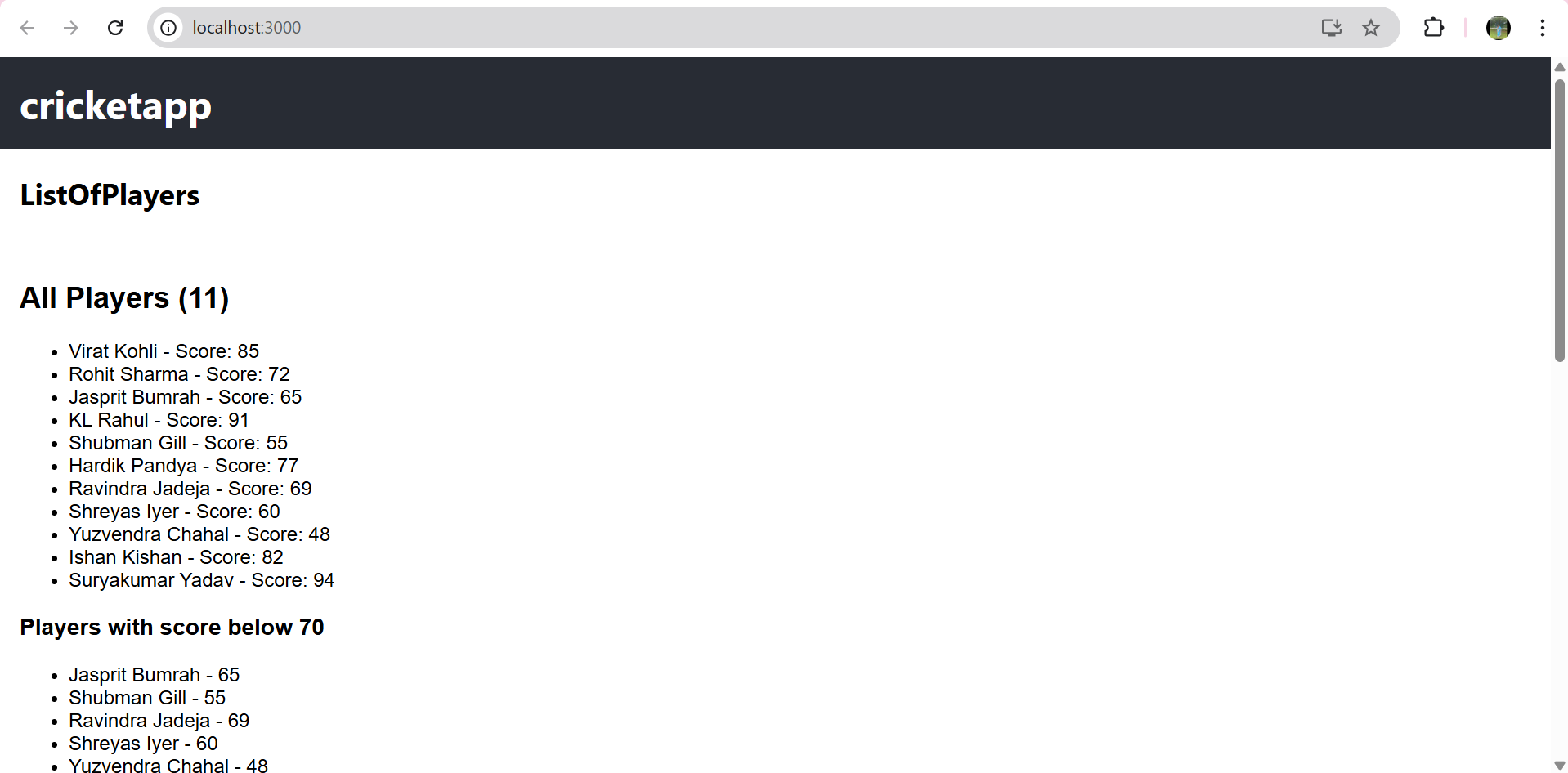
</div>

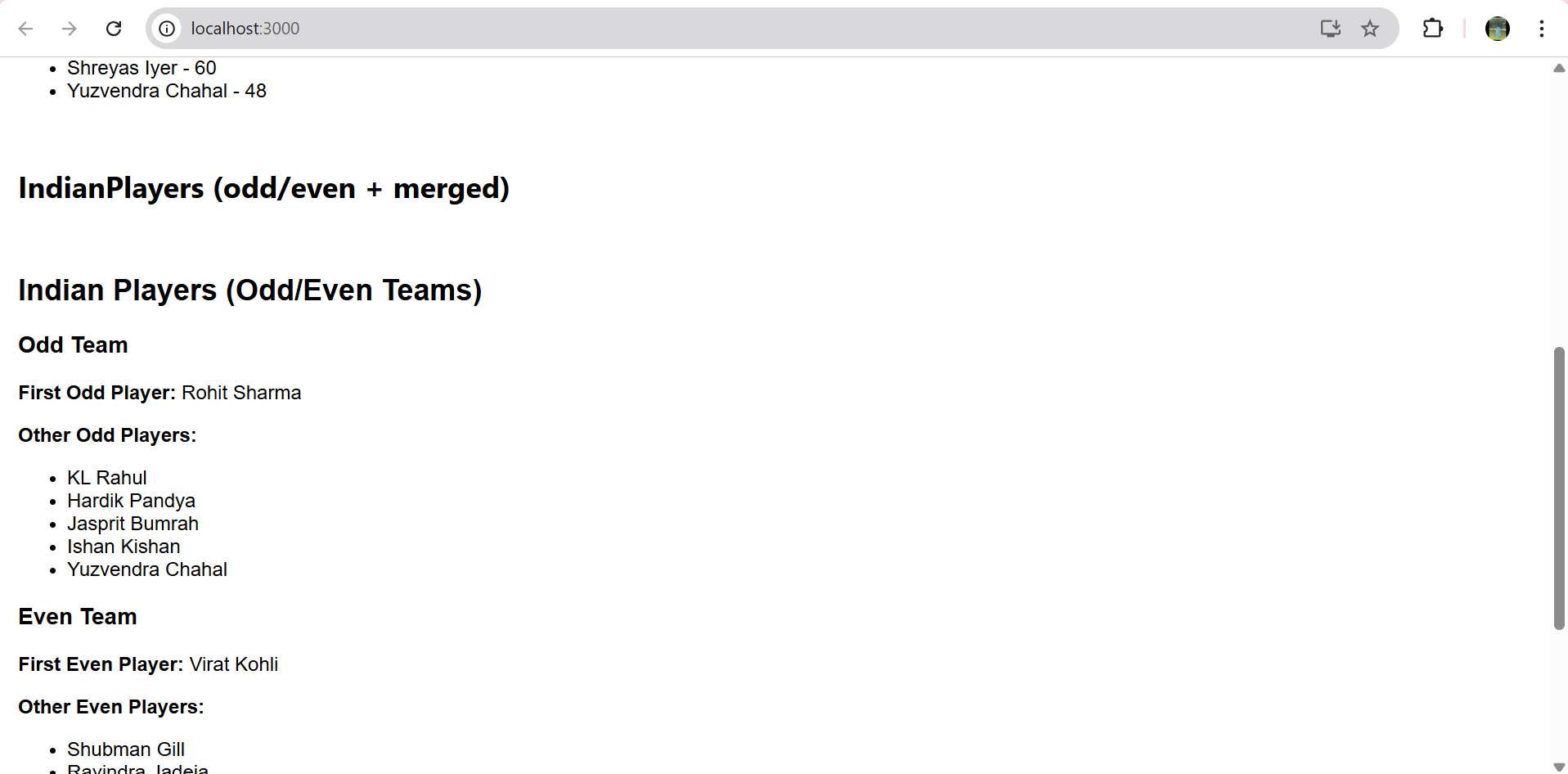
);

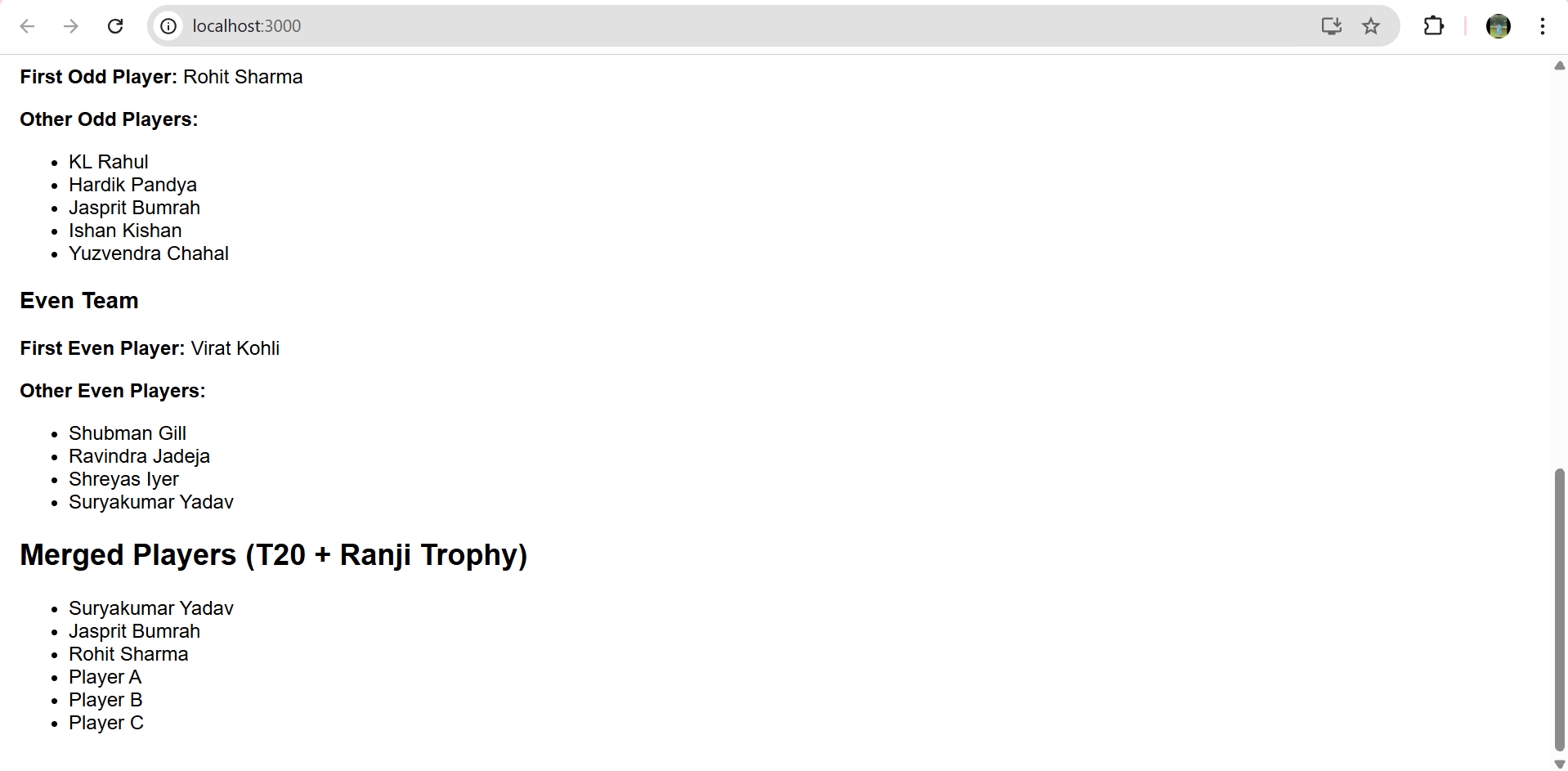
}

export default App;

**OUTPUT -:**







****Lab 10 –** Create a new React Application with the name “officerentalapp”.**

**Hands On:**

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

Create an element to display the heading of the page.

Attribute to display the image of the office space

Create an object of office to display the details like Name, Rent and Address.

Create a list of Object and loop through the office space item to display more data.

To apply Css, Display the color of the Rent in Red if it’s below 60000 and in Green if it’s above 60000.

**SOLUTION :**

**CODE -:**

1. **OfficeList.jsx**

// OfficeList.jsx

import React from "react";

// Single office item component

const OfficeItem = ({ office }) => {

// Inline style for rent: red if < 60000, green if >= 60000

const rentStyle = {

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

fontWeight: "bold",

};

// Container styling (inline as per lab)

const containerStyle = {

border: "1px solid #ccc",

borderRadius: "8px",

padding: "1rem",

marginBottom: "1rem",

display: "flex",

gap: "1rem",

alignItems: "flex-start",

fontFamily: "Arial, sans-serif",

};

const imageStyle = {

width: "150px",

height: "100px",

objectFit: "cover",

borderRadius: "4px",

};

return (

<div style={containerStyle}>

<div>

<img

src={office.image}

alt={`Office: ${office.name}`}

style={imageStyle}

/>

</div>

<div>

<h3 style={{ margin: "0 0 0.5rem 0" }}>{office.name}</h3>

<p style={{ margin: "0.25rem 0" }}>

<strong>Address:</strong> {office.address}

</p>

<p style={{ margin: "0.25rem 0" }}>

<strong>Rent:</strong>{" "}

<span style={rentStyle}>₹{office.rent.toLocaleString()}</span>

</p>

</div>

</div>

);

};

const OfficeList = () => {

// List of office objects

const offices = [

{

name: "Sunrise Business Tower",

rent: 55000,

address: "123 MG Road, Bangalore",

image:

"https://via.placeholder.com/300x200.png?text=Office+1",

},

{

name: "Skyline Plaza",

rent: 75000,

address: "45 Park Street, Kolkata",

image:

"https://via.placeholder.com/300x200.png?text=Office+2",

},

{

name: "Greenleaf Workspace",

rent: 48000,

address: "9 Marine Drive, Mumbai",

image:

"https://via.placeholder.com/300x200.png?text=Office+3",

},

];

return (

<div style={{ padding: "1rem", fontFamily: "Arial, sans-serif" }}>

<h2>Office Space Rental Listings</h2>

{offices.map((office, idx) => (

<OfficeItem key={idx} office={office} />

))}

</div>

);

};

export default OfficeList;

1. **App.js**

// App.js

import React from "react";

import OfficeList from "./components/OfficeList"; // or "./Components/OfficeList" if your folder is capitalized

function App() {

return (

<div>

{/\* Heading of the page \*/}

<header style={{ background: "#222", padding: "1rem", color: "white" }}>

<h1 style={{ margin: 0 }}>officespacerentalapp</h1>

</header>

<main>

{/\* JSX rendering of office list \*/}

<OfficeList />

</main>

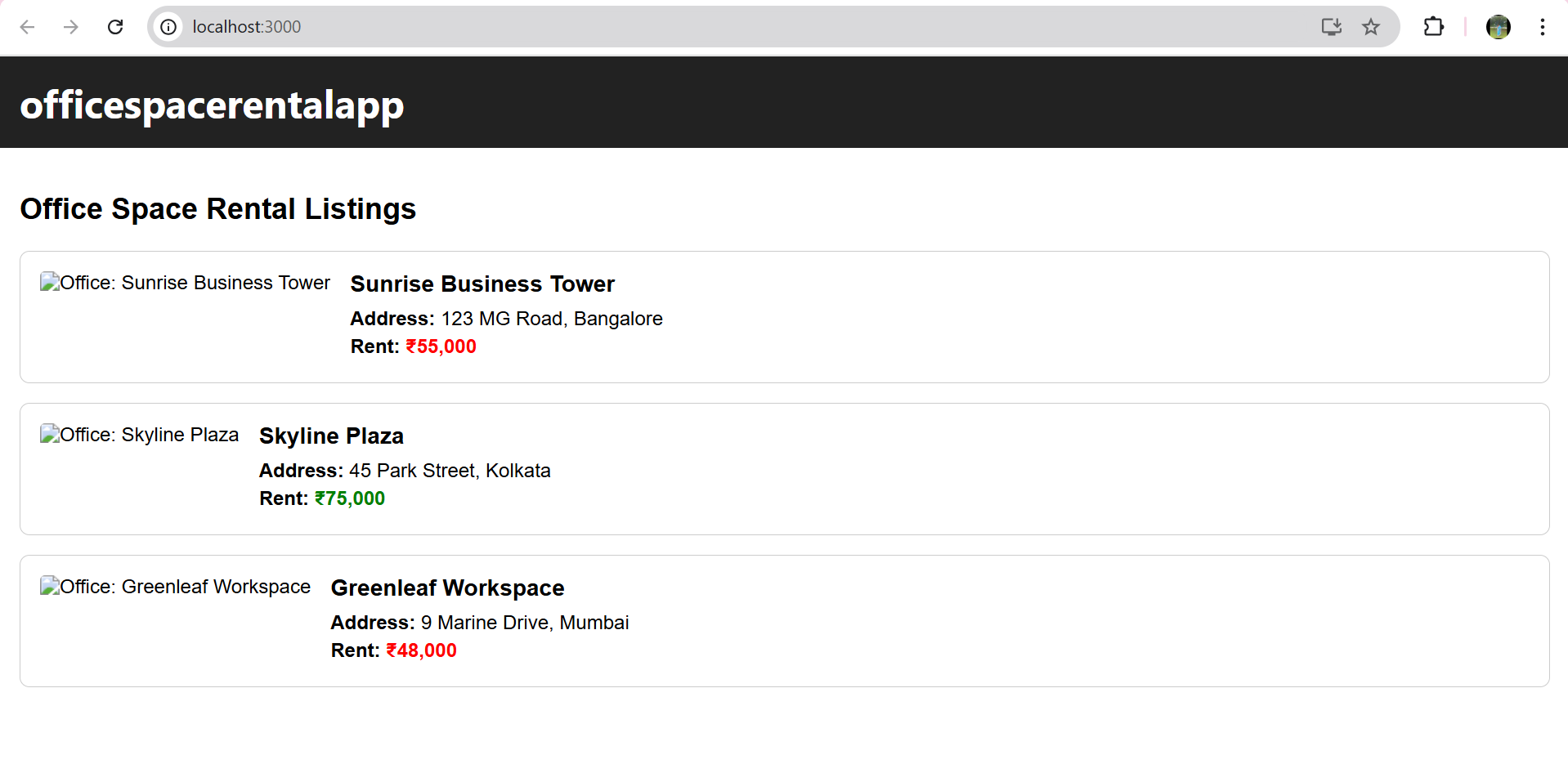
</div>

);

}

export default App;

**OUTPUT -:**



****Lab 11 –** Create a new React Application with the name “eventexamplesapp”to handle various events.**

**Hands On:**

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.
2. Create a button “Say Welcome” which invokes the function which takes “welcome” as an argument.
3. Create a button which invokes synthetic event “OnPress” which display “I was clicked”

Create a “CurrencyConvertor” component which will convert the Indian Rupees to Euro when the Convert button is clicked.

Handle the Click event of the button to invoke the handleSubmit event and handle the conversion of the euro to rupees.

**SOLUTION :**

**CODE -:**

1. **src/components/Conuter.jsx**

// Counter.jsx

import React from "react";

class Counter extends React.Component {

constructor(props) {

super(props);

this.state = {

count: 0,

message: "",

};

// bind if not using arrow methods

this.increment = this.increment.bind(this);

this.decrement = this.decrement.bind(this);

this.sayHello = this.sayHello.bind(this);

this.handleIncrementClick = this.handleIncrementClick.bind(this);

}

sayHello() {

// one of the multiple methods invoked

this.setState({ message: "Hello! This is a static message." });

}

increment() {

this.setState((prev) => ({ count: prev.count + 1 }));

}

decrement() {

this.setState((prev) => ({ count: prev.count - 1 }));

}

handleIncrementClick(e) {

// synthetic event available here (e)

// invoke multiple methods: increment + sayHello

this.increment();

this.sayHello();

// optional: you can inspect synthetic event

console.log("Synthetic event type:", e.type);

}

render() {

return (

<div style={{ border: "1px solid #aaa", padding: "1rem", marginBottom: "1rem" }}>

<h2>Counter</h2>

<p>Current count: {this.state.count}</p>

<div style={{ display: "flex", gap: "0.5rem" }}>

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

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

</div>

<p>

<strong>Message:</strong> {this.state.message}

</p>

</div>

);

}

}

export default Counter;

1. **src/components/Welcomebutton.jsx**

// WelcomeButton.jsx

import React, { useState } from "react";

const WelcomeButton2 = () => {

const [text, setText] = useState("");

const sayWelcome = (word) => {

setText(`You clicked: ${word}`);

};

return (

<div style={{ border: "1px solid #aaa", padding: "1rem", marginBottom: "1rem" }}>

<h2>Say Welcome</h2>

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

{text && <p>{text}</p>}

</div>

);

};

export default WelcomeButton;

1. **src/components/ClickReporter.jsx**

// ClickReporter.jsx

import React, { useState } from "react";

const ClickReporter = () => {

const [clicked, setClicked] = useState(false);

const handlePress = (e) => {

// synthetic event is e

setClicked(true);

console.log("Event object:", e);

};

return (

<div style={{ border: "1px solid #aaa", padding: "1rem", marginBottom: "1rem" }}>

<h2>Synthetic Event Demo</h2>

<button onClick={handlePress}>Press Me</button>

{clicked && <p>I was clicked</p>}

</div>

);

};

export default ClickReporter;

1. **src/components/CuurencyConverter.jsx**

// CurrencyConverter.jsx

import React, { useState } from "react";

const CurrencyConverter = () => {

// Example rate: 1 Euro = 90 INR

const INR\_PER\_EURO = 90;

const [inr, setInr] = useState("");

const [euro, setEuro] = useState("");

const [result, setResult] = useState("");

const handleSubmit = (e) => {

e.preventDefault(); // prevent page reload (synthetic event)

// If INR is filled, convert to Euro

if (inr !== "") {

const converted = parseFloat(inr) / INR\_PER\_EURO;

setResult(`${inr} INR = ${converted.toFixed(2)} EUR`);

} else if (euro !== "") {

const converted = parseFloat(euro) \* INR\_PER\_EURO;

setResult(`${euro} EUR = ${converted.toFixed(2)} INR`);

} else {

setResult("Please enter INR or Euro.");

}

};

return (

<div style={{ border: "1px solid #aaa", padding: "1rem", marginBottom: "1rem" }}>

<h2>Currency Converter (INR ⇄ EUR)</h2>

<form onSubmit={handleSubmit}>

<div style={{ marginBottom: "0.5rem" }}>

<label>

INR:{" "}

<input

type="number"

value={inr}

onChange={(e) => {

setInr(e.target.value);

setEuro("");

setResult("");

}}

placeholder="Amount in INR"

/>

</label>

</div>

<div style={{ marginBottom: "0.5rem" }}>

<label>

Euro:{" "}

<input

type="number"

value={euro}

onChange={(e) => {

setEuro(e.target.value);

setInr("");

setResult("");

}}

placeholder="Amount in EUR"

/>

</label>

</div>

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

</form>

{result && (

<p style={{ marginTop: "0.5rem" }}>

<strong>{result}</strong>

</p>

)}

<p style={{ fontSize: "0.8rem", marginTop: "0.25rem" }}>

Rate used: 1 EUR = {INR\_PER\_EURO} INR

</p>

</div>

);

};

export default CurrencyConverter;

1. **App.js**

// App.js

import React from "react";

import Counter from "./components/Counter.jsx";

import WelcomeButton2 from "./components/WelcomeButton2.jsx";

import ClickReporter from "./components/ClickReporter.jsx";

import CurrencyConverter from "./components/CurrencyConverter.jsx";

function App() {

return (

<div style={{ fontFamily: "Arial, sans-serif" }}>

<header style={{ background: "#222", padding: "1rem", color: "white" }}>

<h1 style={{ margin: 0 }}>eventexamplesapp</h1>

</header>

<main style={{ padding: "1rem" }}>

<Counter />

<WelcomeButton2 />

<ClickReporter />

<CurrencyConverter />

</main>

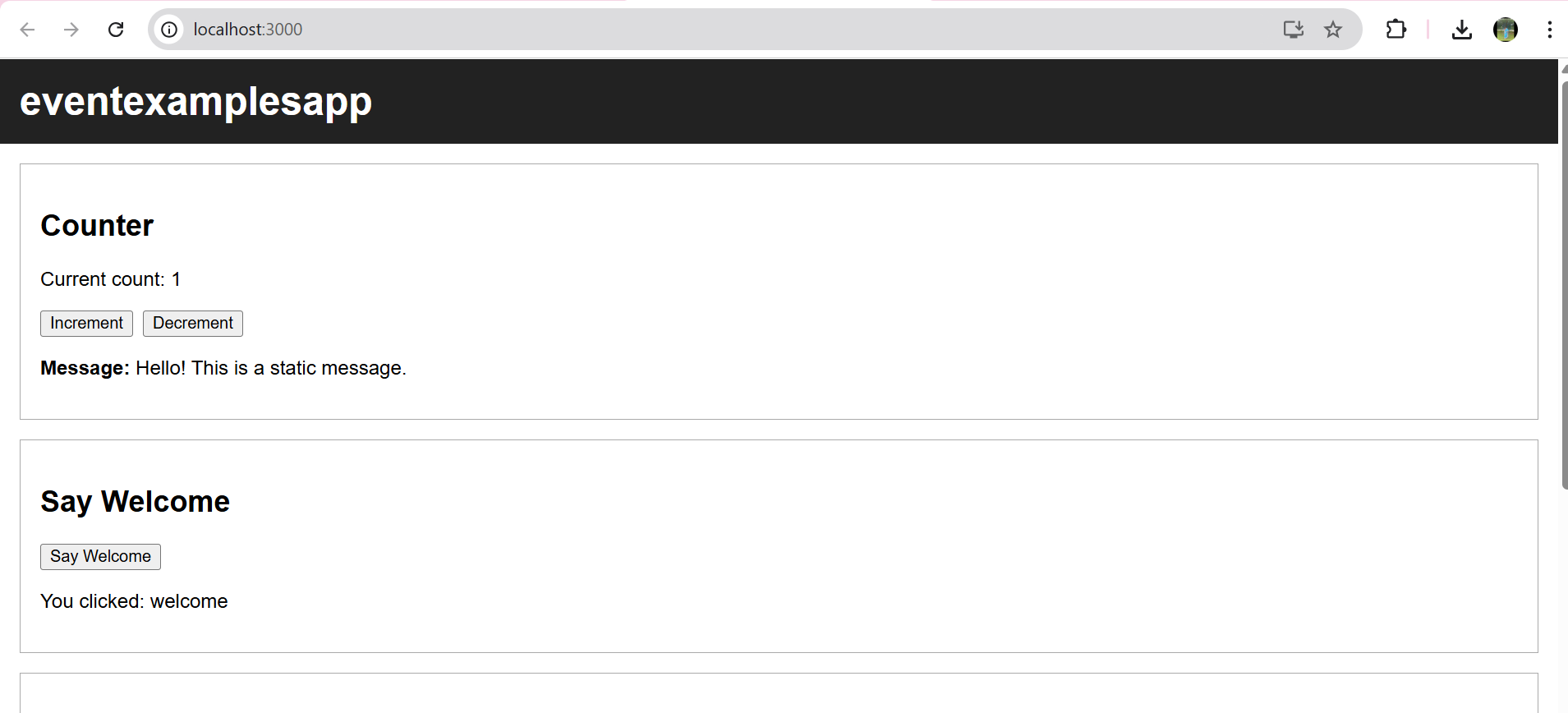
</div>

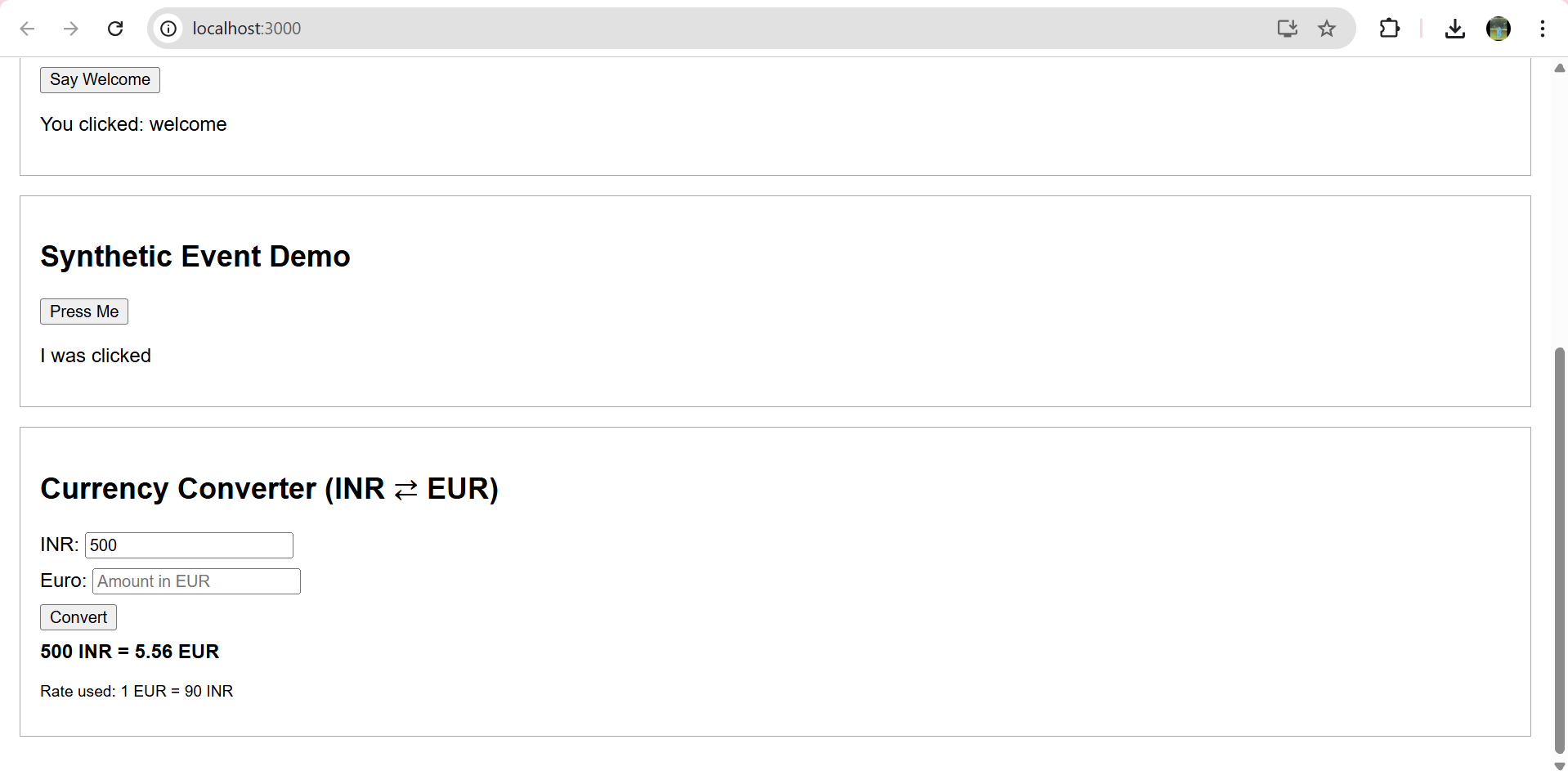
);

}

export default App;

**OUTPUT -:**





****Lab 12 –** Create a new React Application with the name “ticketbookingapp”.**

**Hands On:**

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.

**SOLUTION :**

**CODE -:**

1. **src/GuestPage.js**

import React from 'react';

function GuestPage({ onLogin }) {

return (

<div>

<h2>Welcome Guest</h2>

<p>View available flights below.</p>

<ul>

<li>✈️ Delhi to Mumbai - 9:00 AM</li>

<li>✈️ Bangalore to Kolkata - 1:30 PM</li>

</ul>

<button onClick={onLogin}>Login to Book</button>

</div>

);

}

export default GuestPage;

1. **Src/UserPage.js**

import React from 'react';

function UserPage({ onLogout }) {

return (

<div>

<h2>Welcome User</h2>

<p>You are now logged in. You can book tickets below.</p>

<ul>

<li>✈️ Book: Delhi to Mumbai - ₹3500</li>

<li>✈️ Book: Bangalore to Kolkata - ₹4200</li>

</ul>

<button onClick={onLogout}>Logout</button>

</div>

);

}

export default UserPage;

1. **App.js**

import React, { useState } from 'react';

import GuestPage from './GuestPage';

import UserPage from './UserPage';

function App() {

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

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

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

return (

<div className="App">

<h1>Flight Ticket Booking App</h1>

{

isLoggedIn ? (

<>

<UserPage onLogout={handleLogout} />

</>

) : (

<>

<GuestPage onLogin={handleLogin} />

</>

)

}

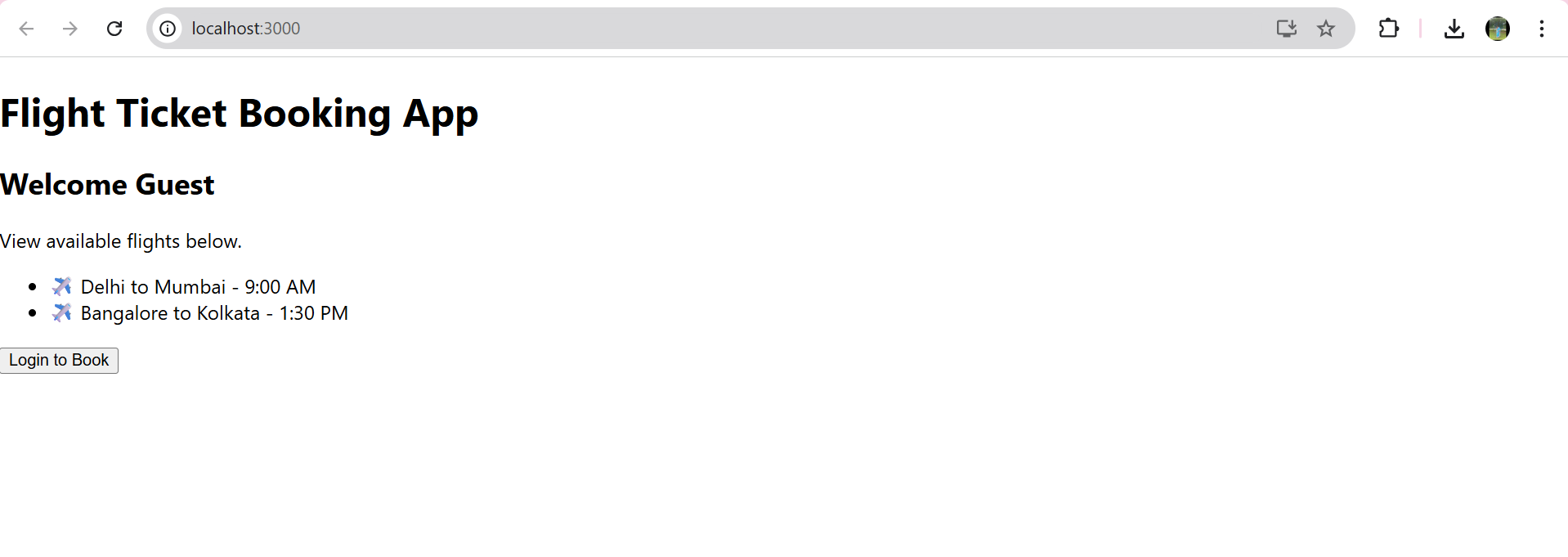
</div>

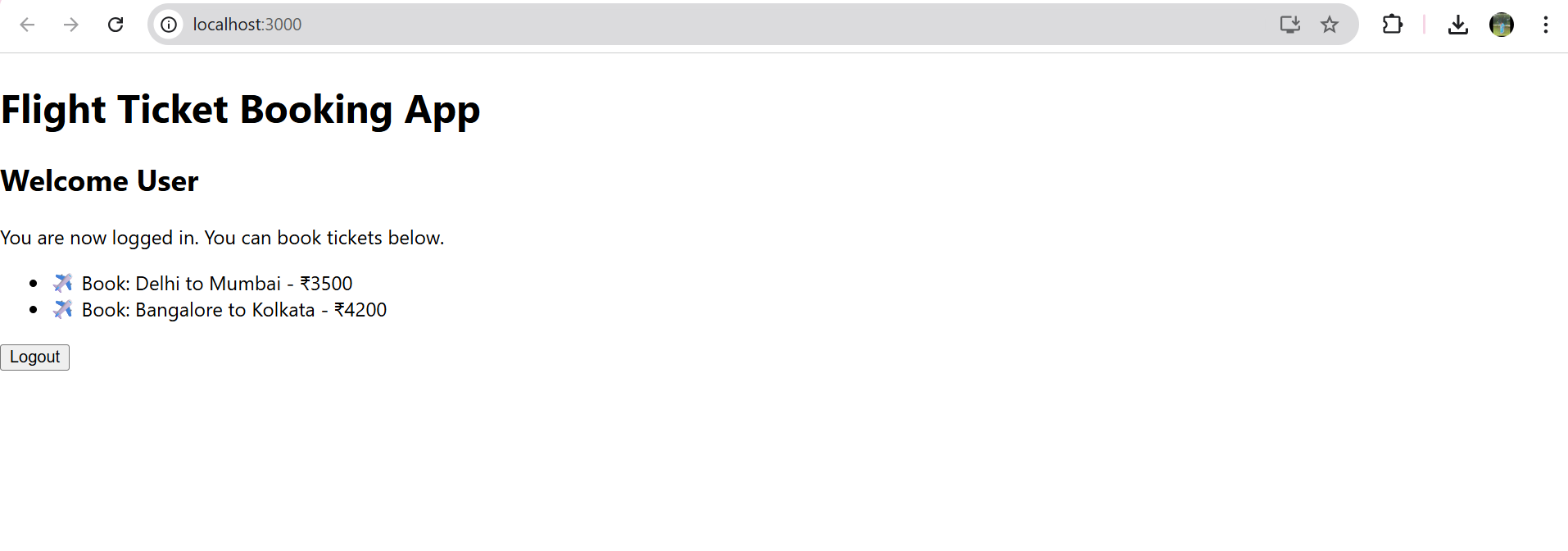
);

}

export default App;

**OUTPUT -:**





****Lab 13 –** Create a new React Application with the name “bloggerapp”.**

**Hands On:**

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.

**SOLUTION :**

**CODE -:**

1. **src/components/BookDetails.js**

function BookDetails() {

return (

<div>

<h2>📘 Book Details</h2>

<p>Book: "Atomic Habits" by James Clear</p>

</div>

);

}

export default BookDetails;

1. **src/components/BlogDetails.js**

function BlogDetails() {

return (

<div>

<h2>📝 Blog Details</h2>

<p>Blog: "React Tricks for Beginners"</p>

</div>

);

}

export default BlogDetails;

1. **Src/components/CourseDetails.js**

function CourseDetails() {

return (

<div>

<h2>🎓 Course Details</h2>

<p>Course: "Full Stack Web Development"</p>

</div>

);

}

export default CourseDetails;

1. **App.js**

import React, { useState } from 'react';

import BookDetails from './components/BookDetails';

import BlogDetails from './components/BlogDetails';

import CourseDetails from './components/CourseDetails';

function App() {

const [view, setView] = useState(''); // book, blog, course

// 🔁 Element variable (Switch-case)

let renderedComponent;

switch (view) {

case 'book':

renderedComponent = <BookDetails />;

break;

case 'blog':

renderedComponent = <BlogDetails />;

break;

case 'course':

renderedComponent = <CourseDetails />;

break;

default:

renderedComponent = <p>Please select a section to view.</p>;

}

return (

<div className="App">

<h1>BloggerApp Dashboard</h1>

{/\* Buttons to toggle views \*/}

<button onClick={() => setView('book')}>Show Book</button>

<button onClick={() => setView('blog')}>Show Blog</button>

<button onClick={() => setView('course')}>Show Course</button>

<button onClick={() => setView('')}>Clear</button>

<hr />

{/\* 🧠 Conditional rendering methods \*/}

<h3>1. Using element variable (switch):</h3>

{renderedComponent}

<h3>2. Ternary rendering:</h3>

{view === 'book' ? <BookDetails /> : <p>Not viewing books</p>}

<h3>3. Logical AND rendering:</h3>

{view === 'blog' && <BlogDetails />}

</div>

);

}

export default App;

**OUTPUT -:**

