**JAVA FSE WEEK 7**

**Reactjs-HOL**

**Mandatory Hands-on Excercises**

**1 Hands-on : cricketapp**

**Objective:**Create a React application named **cricketapp** that demonstrates ES6 features using:

* map() method to list player details
* Arrow functions to filter players by score
* Destructuring to separate even and odd indexed players
* Merging arrays using the spread operator
* Conditional rendering using a flag

**Steps:**

1. npx create-react-app cricketapp
2. cd cricketapp
3. Create two components: ListofPlayers.js and IndianPlayers.js inside src
4. Modify App.js to conditionally render components using a flag variable
5. npm start

**Code :**

**ListofPlayers.js**

import React from 'react';

const players = [

{ name: 'Virat', score: 90 },

{ name: 'Rohit', score: 85 },

{ name: 'Dhoni', score: 50 },

{ name: 'Jadeja', score: 45 },

{ name: 'Hardik', score: 78 },

{ name: 'Shami', score: 34 },

{ name: 'Gill', score: 88 },

{ name: 'Surya', score: 69 },

{ name: 'Pant', score: 73 },

{ name: 'Bumrah', score: 38 },

{ name: 'Kuldeep', score: 40 },

];

function ListofPlayers() {

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

return (

<div>

<h2>Players with Score ≥ 70</h2>

<ul>

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

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

))}

</ul>

</div>

);

}

export default ListofPlayers;

**IndianPlayers.js**

import React from 'react';

function IndianPlayers() {

const T20players = ['Kohli', 'Rohit', 'Surya', 'Pant'];

const RanjiTrophy = ['Dhawan', 'Shaw', 'Pujara'];

const allPlayers = [...T20players, ...RanjiTrophy];

const evenTeam = allPlayers.filter((\_, index) => index % 2 === 0);

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

return (

<div>

<h2>Even Team Players</h2>

<ul>

{evenTeam.map((player, index) => <li key={index}>{player}</li>)}

</ul>

<h2>Odd Team Players</h2>

<ul>

{oddTeam.map((player, index) => <li key={index}>{player}</li>)}

</ul>

</div>

);

}

export default IndianPlayers;

**App.js**

import React from 'react';

import ListofPlayers from './ListofPlayers';

import IndianPlayers from './IndianPlayers';

function App() {

const flag = true; // change to false to render IndianPlayers

return (

<div className="App">

<h1>🏏 Cricket App</h1>

{flag ? <ListofPlayers /> : <IndianPlayers />}

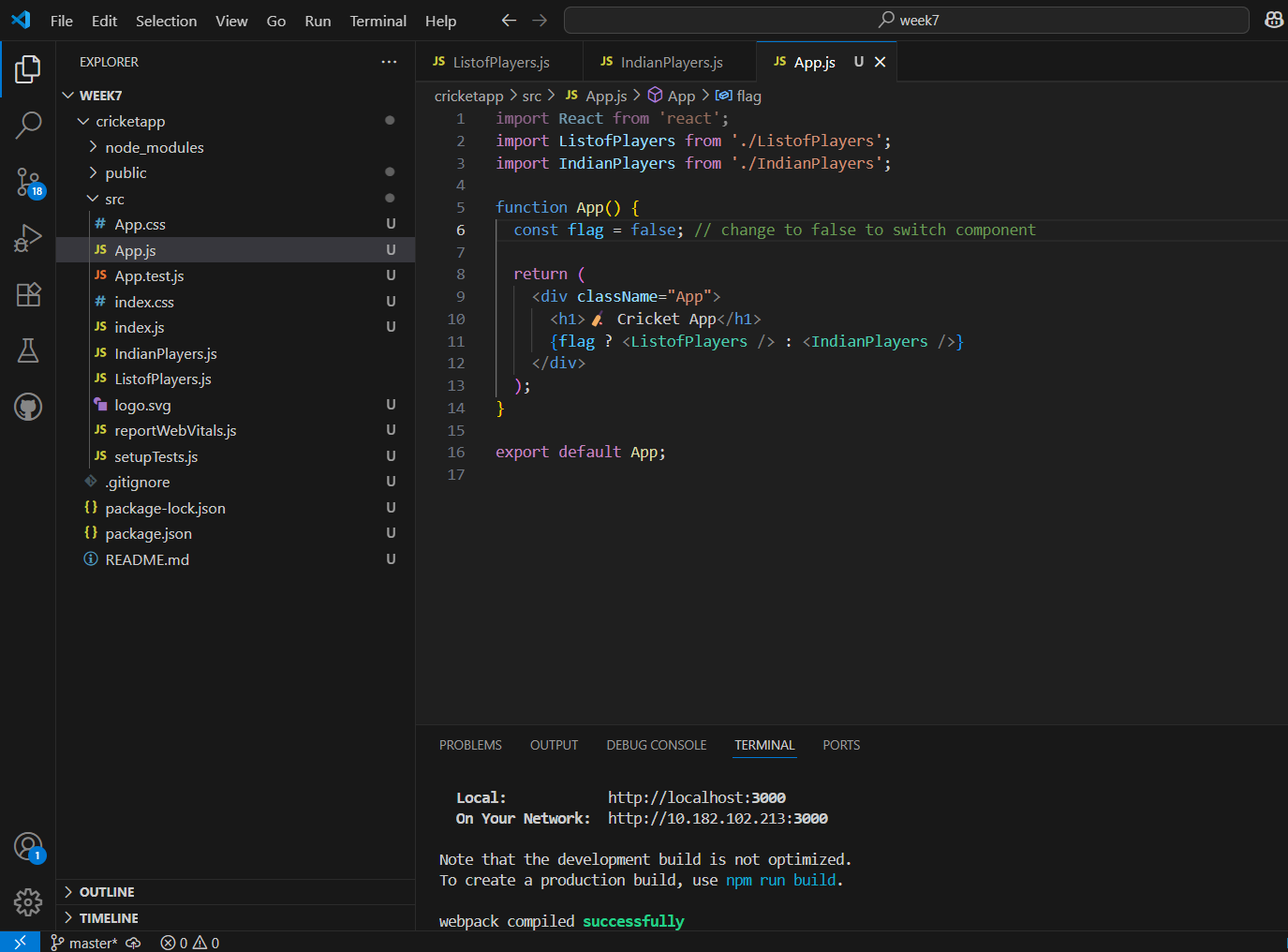
</div>

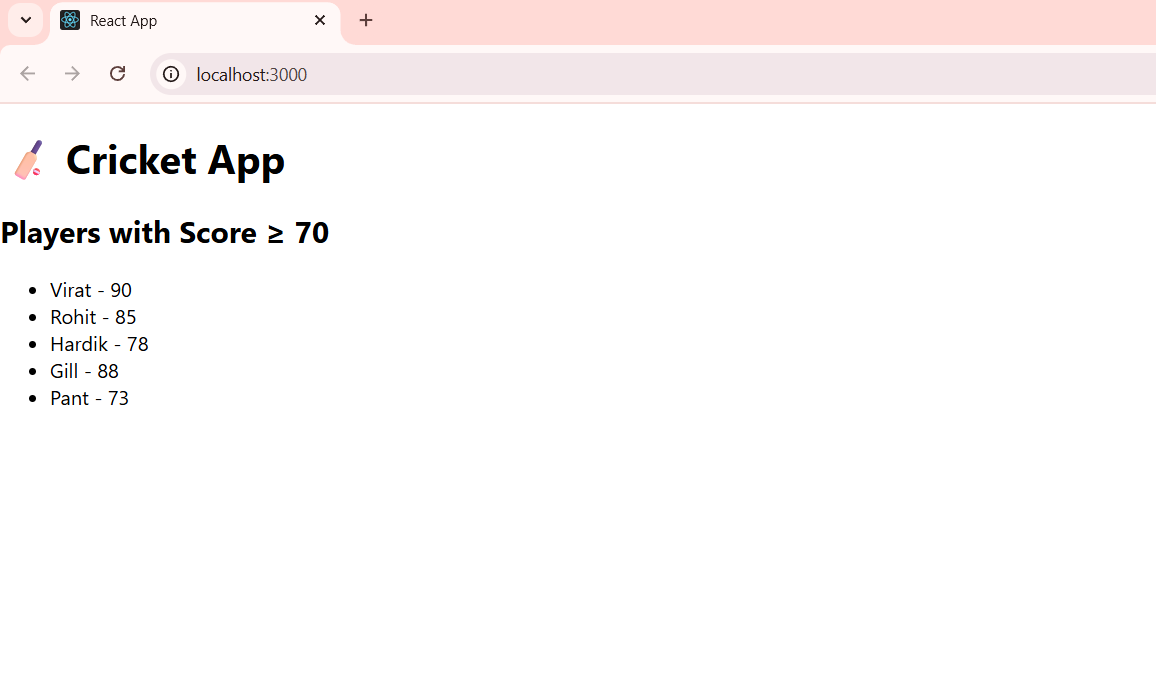
);

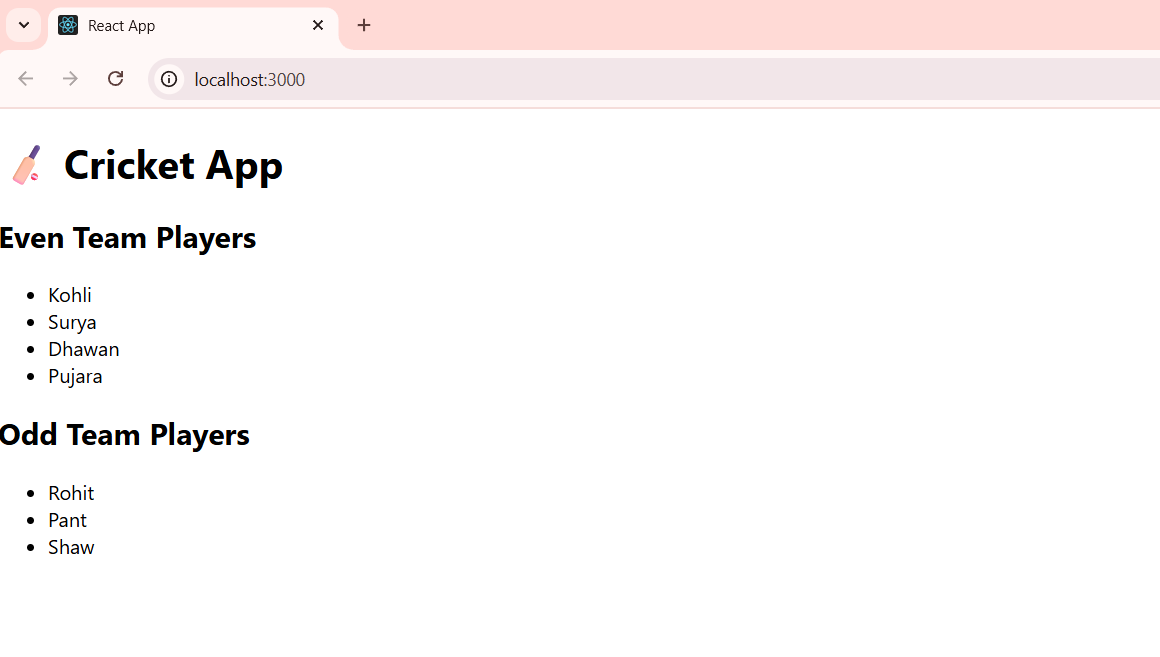
}

export default App;

**OUTPUT:**







**2 Hands-on : officespacerentalapp**

**Objective:**Create a React application named **officespacerentalapp** to:

* Use JSX syntax to create elements and attributes
* Render DOM elements to display office space data
* Display office rent in red if it’s below 60000 and in green if it’s above 60000

**Steps:**

1. npx create-react-app officespacerentalapp
2. cd officespacerentalapp
3. Modify App.js
4. npm start

**Code :**

**App.js**

import React from 'react';

function App() {

const office = {

name: 'Tech Park Workspace',

rent: 55000,

address: '1st Floor, Silicon Valley, Hyderabad'

};

const officeList = [

{ name: 'Startup Hub', rent: 45000, address: 'Gachibowli' },

{ name: 'Innovation Tower', rent: 75000, address: 'Hitech City' },

{ name: 'Prime Offices', rent: 62000, address: 'Madhapur' },

];

const getRentColor = (rent) => {

return {

color: rent < 60000 ? 'red' : 'green'

};

};

return (

<div>

<h1>Office Space Rental App</h1>

<h2>Single Office Details</h2>

<p><strong>Name:</strong> {office.name}</p>

<p><strong>Rent:</strong> <span style={getRentColor(office.rent)}>{office.rent}</span></p>

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

<h2>All Offices List</h2>

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

<div key={index} style={{ border: '1px solid gray', padding: '10px', marginBottom: '10px' }}>

<p><strong>Name:</strong> {item.name}</p>

<p><strong>Rent:</strong> <span style={getRentColor(item.rent)}>{item.rent}</span></p>

<p><strong>Address:</strong> {item.address}</p>

</div>

))}

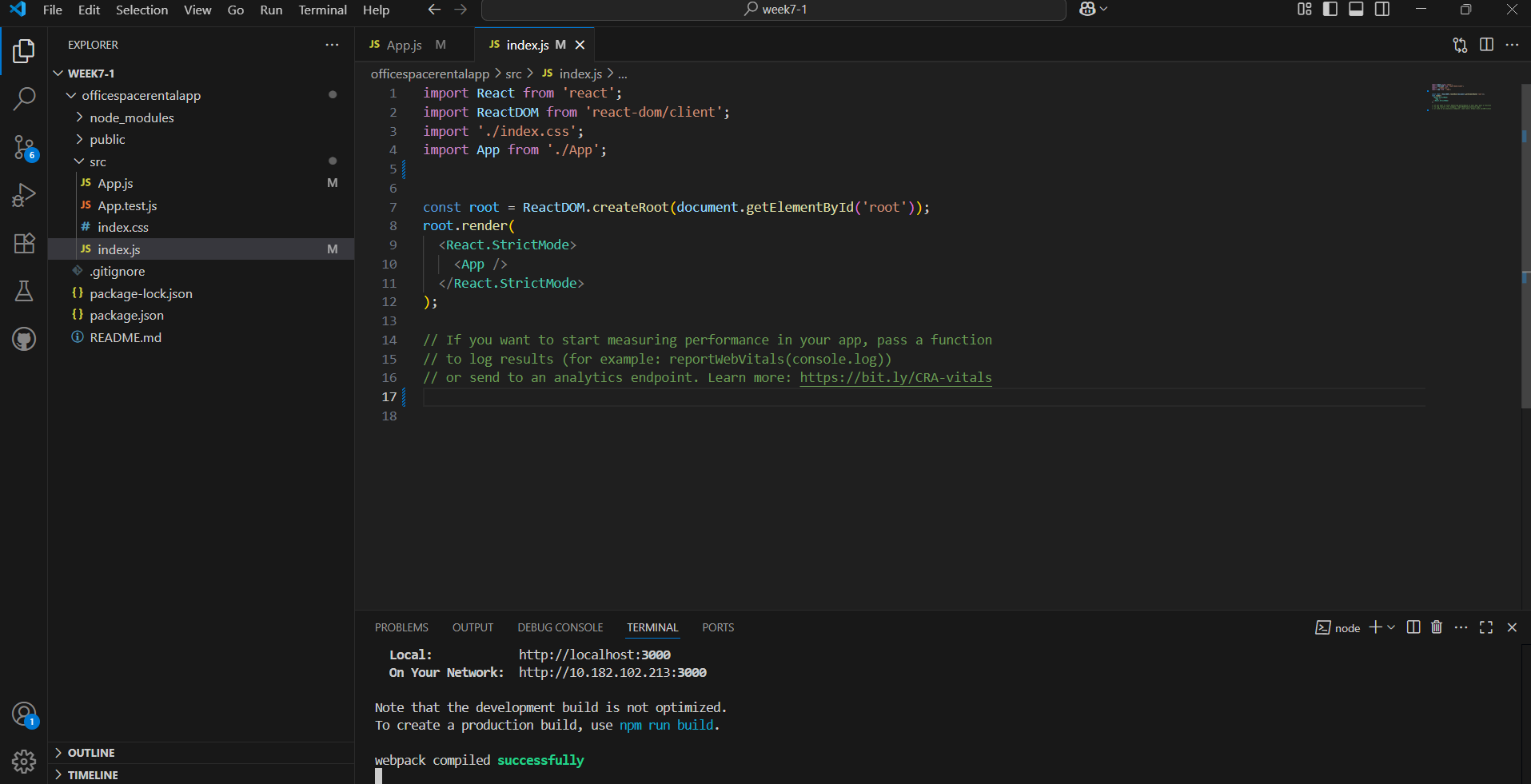
</div>

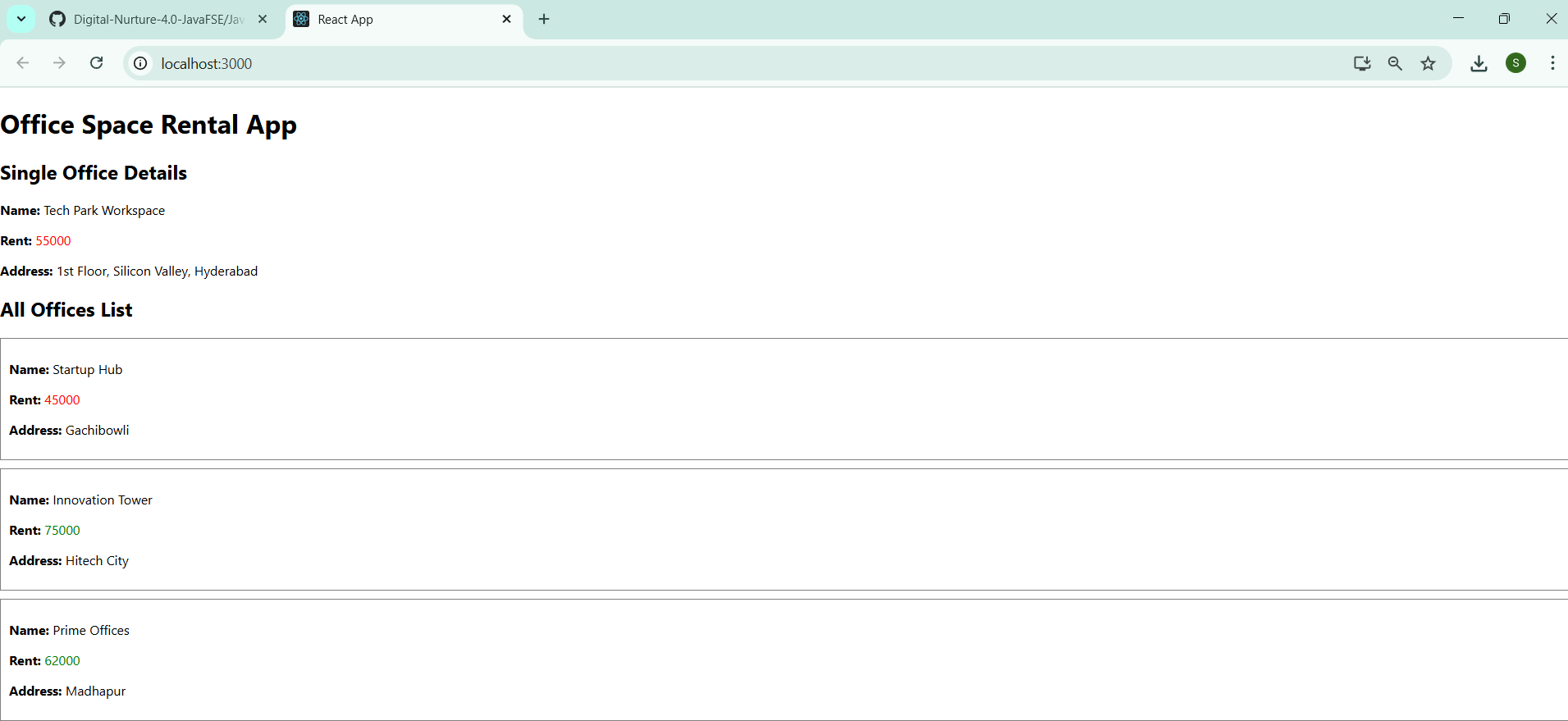
);

}

export default App;

**OUTPUT:**





**3 Hands-on : eventexamplesapp**

**Objective:**Create a React application named eventexamplesapp to demonstrate event handling using:

* Button click events
* Multiple function calls
* Passing arguments to event handlers
* Synthetic events
* Currency conversion using input field and state

**Steps:**

1. npx create-react-app eventexamplesapp
2. cd eventexamplesapp
3. Modify App.js
4. npm start

**Code :**

**App.js**

import React, { useState } from 'react';

function App() {

const [count, setCount] = useState(0);

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

const [eur, setEur] = useState('');

const handleIncrement = () => {

setCount(count + 1);

sayHello();

};

const sayHello = () => {

alert("Hello! Count incremented");

};

const handleDecrement = () => {

setCount(count - 1);

};

const sayWelcome = (msg) => {

alert(`Message: ${msg}`);

};

const onPress = () => {

alert("I was clicked");

};

const handleConvert = () => {

if (inr) {

const converted = (parseFloat(inr) / 90).toFixed(2);

setEur(converted);

}

};

return (

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

<h1>React Event Handling Examples</h1>

<h2>Counter</h2>

<p>Count: {count}</p>

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

<button onClick={handleDecrement} style={{ marginLeft: '10px' }}>Decrement</button>

<h2>Say Welcome</h2>

<button onClick={() => sayWelcome("Welcome to React!")}>Say Welcome</button>

<h2>Synthetic Event</h2>

<button onClick={onPress}>Click Me</button>

<h2>Currency Converter (INR to Euro)</h2>

<input

type="text"

value={inr}

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

placeholder="Enter INR amount"

/>

<button onClick={handleConvert} style={{ marginLeft: '10px' }}>Convert</button>

<p>Converted Amount: €{eur}</p>

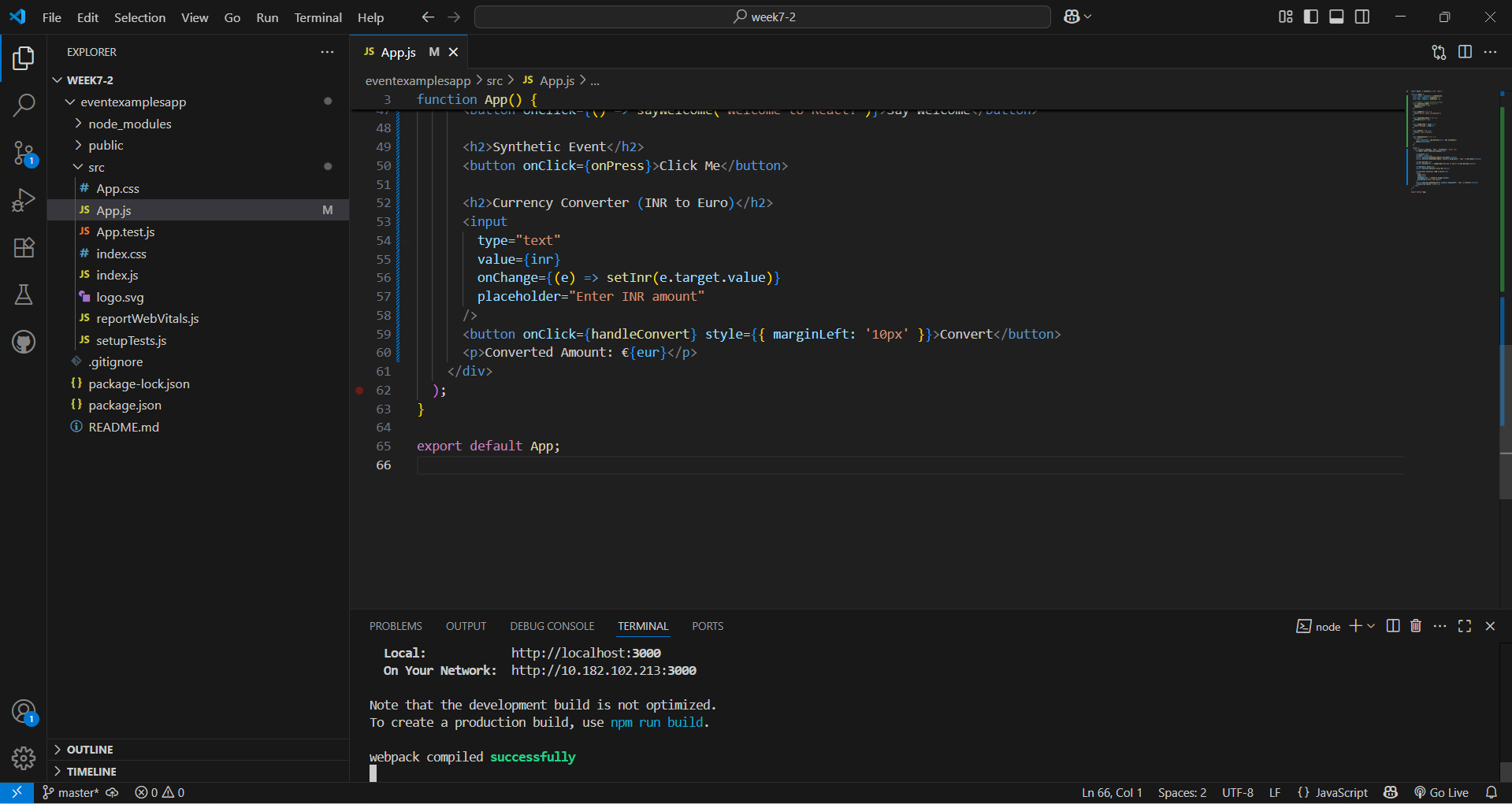
</div>

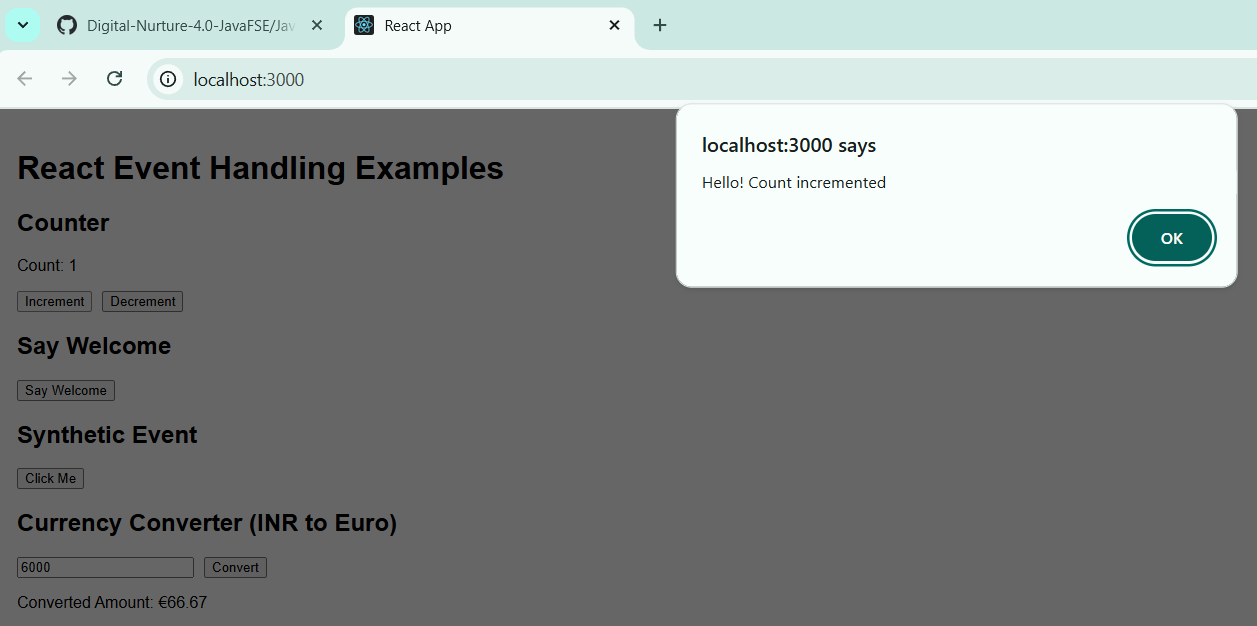
);

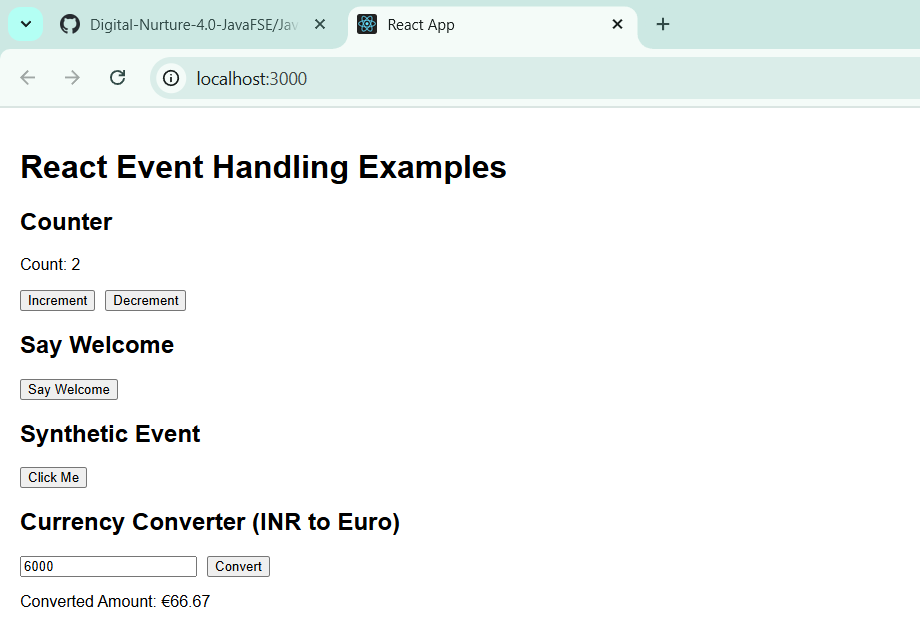
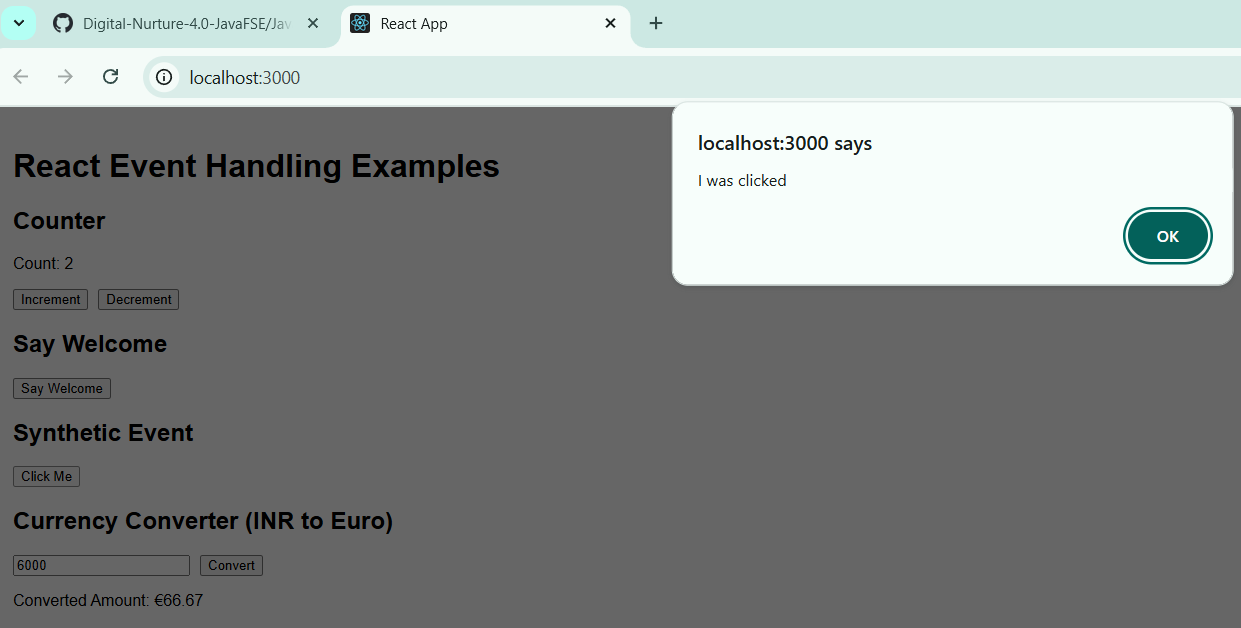
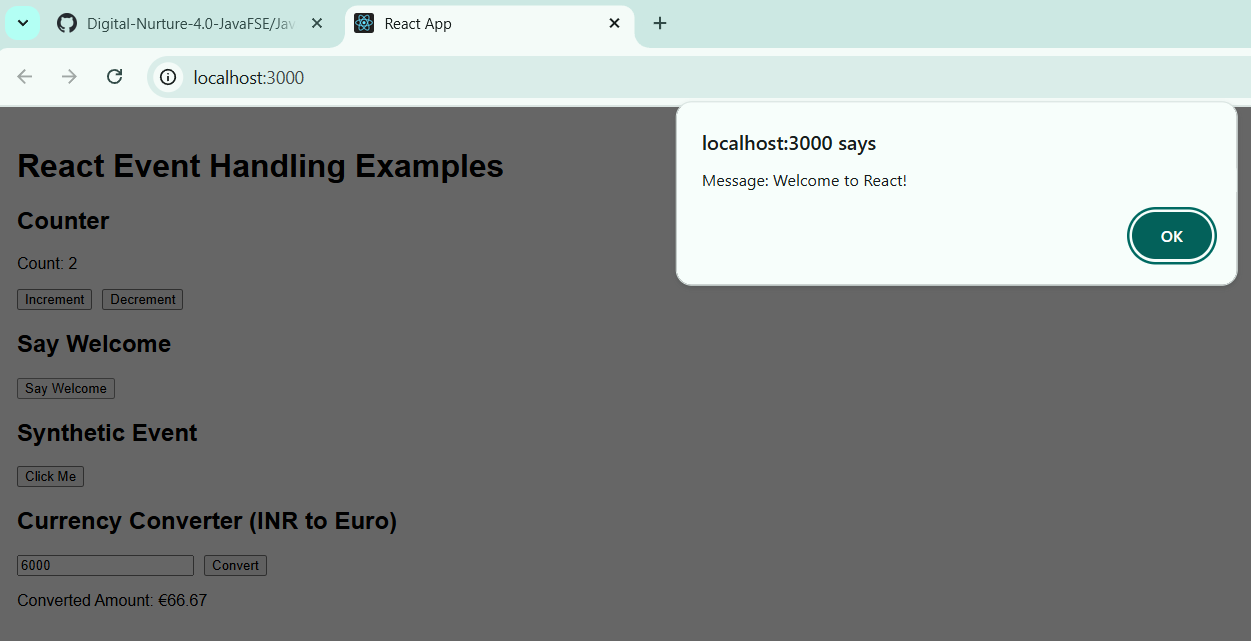
}

export default App;

**OUTPUT:**







**4 Hands-on : ticketbookingapp**

**Objective:**  
Create a React application named **ticketbookingapp** that demonstrates conditional rendering:

* Display a Guest page if the user is not logged in
* Display a User page with booking access if logged in
* Toggle between Login and Logout views using a button

**Steps:**

1. npx create-react-app ticketbookingapp
2. cd ticketbookingapp
3. Create components: Guest.js, User.js, and LoginControl.js
4. Modify App.js
5. npm start

**Code :**

**Guest.js**

import React from 'react';

function Guest() {

return (

<div>

<h2>Welcome Guest</h2>

<p>You can browse available flights.</p>

</div>

);

}

export default Guest;

**User.js**

import React from 'react';

function User() {

return (

<div>

<h2>Welcome User</h2>

<p>You can now book your flight tickets.</p>

</div>

);

}

export default User;

**LoginControl.js**

import React, { useState } from 'react';

import Guest from './Guest';

import User from './User';

function LoginControl() {

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

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

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

return (

<div>

{isLoggedIn ? <User /> : <Guest />}

<button onClick={isLoggedIn ? handleLogout : handleLogin}>

{isLoggedIn ? 'Logout' : 'Login'}

</button>

</div>

);

}

export default LoginControl;

App.js

jsx

CopyEdit

import React from 'react';

import LoginControl from './LoginControl';

function App() {

return (

<div className="App">

<h1>Ticket Booking App</h1>

<LoginControl />

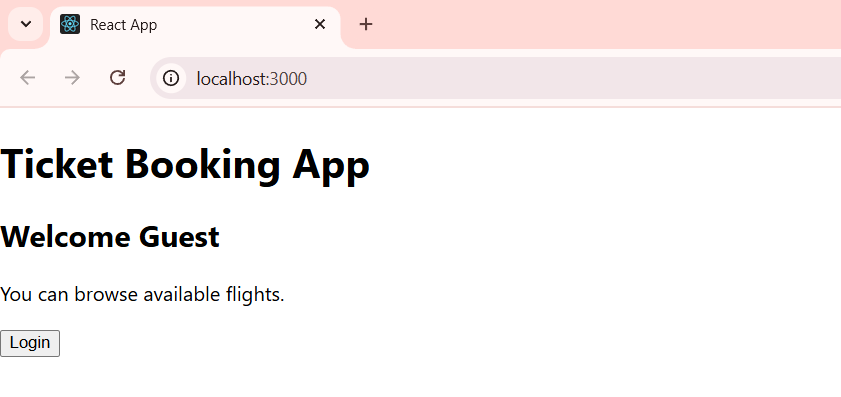
</div>

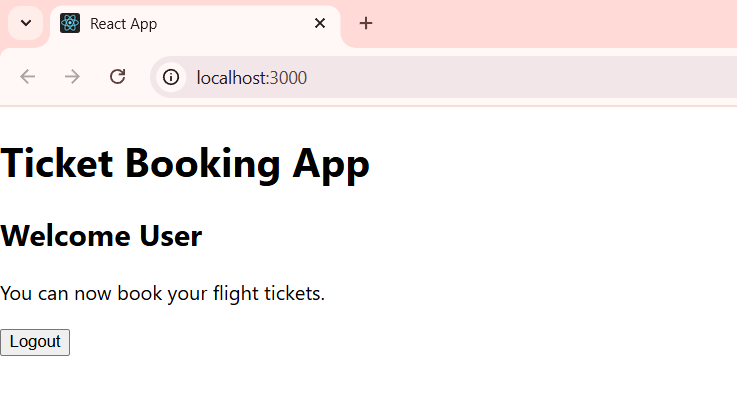
);

}

export default App;

**OUTPUT:**





**5 Hands-on : bloggerapp**

**Objective:**  
Create a React application named **bloggerapp** that demonstrates:

* Conditional rendering using if-else, ternary, && operator, and element variables
* List rendering with keys
* Extracting components and using map() function

**Steps:**

1. npx create-react-app bloggerapp
2. cd bloggerapp
3. Create components: BookDetails.js, BlogDetails.js, and CourseDetails.js
4. Modify App.js to implement multiple conditional rendering techniques
5. npm start

**Code :**

**BookDetails.js**

import React from 'react';

function BookDetails() {

const books = [

{ id: 1, title: "The Alchemist" },

{ id: 2, title: "1984" },

{ id: 3, title: "Atomic Habits" },

];

return (

<div>

<h2>Book Details</h2>

<ul>

{books.map(book => (

<li key={book.id}>{book.title}</li>

))}

</ul>

</div>

);

}

export default BookDetails;

**BlogDetails.js**

import React from 'react';

function BlogDetails() {

const blogs = [

"React Basics",

"Props and State",

"Conditional Rendering",

];

return (

<div>

<h2>Blog Posts</h2>

<ul>

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

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

))}

</ul>

</div>

);

}

export default BlogDetails;

**CourseDetails.js**

import React from 'react';

function CourseDetails() {

const courses = ["Frontend", "Backend", "DevOps"];

const showCourses = true;

return (

<div>

<h2>Course Details</h2>

{showCourses && (

<ul>

{courses.map((course, index) => (

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

))}

</ul>

)}

</div>

);

}

export default CourseDetails;

**App.js**

import React from 'react';

import BookDetails from './BookDetails';

import BlogDetails from './BlogDetails';

import CourseDetails from './CourseDetails';

function App() {

const view = "books"; // change to "blogs" or "courses"

let content;

if (view === "books") {

content = <BookDetails />;

} else if (view === "blogs") {

content = <BlogDetails />;

} else if (view === "courses") {

content = <CourseDetails />;

}

return (

<div className="App">

<h1>Blogger App</h1>

{content}

<hr />

<h3>Rendered Again Using Ternary:</h3>

{view === "books" ? <BookDetails /> : view === "blogs" ? <BlogDetails /> : <CourseDetails />}

</div>

);

}

export default App;

**OUTPUT:**

