**WEEK - 7**

**REACT**

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

**File name: App.js**

import React from 'react';

import ListofPlayers from './ListofPlayers';

import IndianPlayers from './IndianPlayers';

const App = () => {

const flag = false; // Change to false to switch components

return (

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

<h1>Cricket App</h1>

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

</div>

);

};

export default App;

**File name: 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;

}

@keyframes App-logo-spin {

from {

transform: rotate(0deg);

}

to {

transform: rotate(360deg);

}

}

**File name: index.css**

body {

margin: 0;

font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',

'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',

sans-serif;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

}

code {

font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',

monospace;

}

**File name: index.js**

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

<React.StrictMode>

<App />

</React.StrictMode>

);

**File name: IndianPlayers.js**

import React from 'react';

const IndianPlayers = () => {

const T20players = ["Virat", "Rohit", "Hardik", "Gill", "Pant"];

const RanjiTrophyPlayers = ["Pujara", "Rahane", "Jadeja", "Shreyas", "Ashwin"];

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

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

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

return (

<div>

<h2>Odd Team Players</h2>

{oddTeam.map((p, i) => <p key={i}>{p}</p>)}

<h2>Even Team Players</h2>

{evenTeam.map((p, i) => <p key={i}>{p}</p>)}

<h3>Merged Team</h3>

{allPlayers.map((p, i) => <p key={i}>{p}</p>)}

</div>

);

};

export default IndianPlayers;

**File name: ListofPlayers.js**

import React from 'react';

const ListofPlayers = () => {

const players = [

{ name: "Virat", score: 85 },

{ name: "Rohit", score: 45 },

{ name: "Hardik", score: 72 },

{ name: "Dhoni", score: 68 },

{ name: "Jadeja", score: 90 },

{ name: "Gill", score: 55 },

{ name: "Pant", score: 35 },

{ name: "Surya", score: 88 },

{ name: "Shami", score: 60 },

{ name: "Bumrah", score: 95 },

{ name: "Kuldeep", score: 40 }

];

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

return (

<div>

<h2> All Players</h2>

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

<p key={i}>{p.name} - Score: {p.score}</p>

))}

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

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

<p key={i}>{p.name}</p>

))}

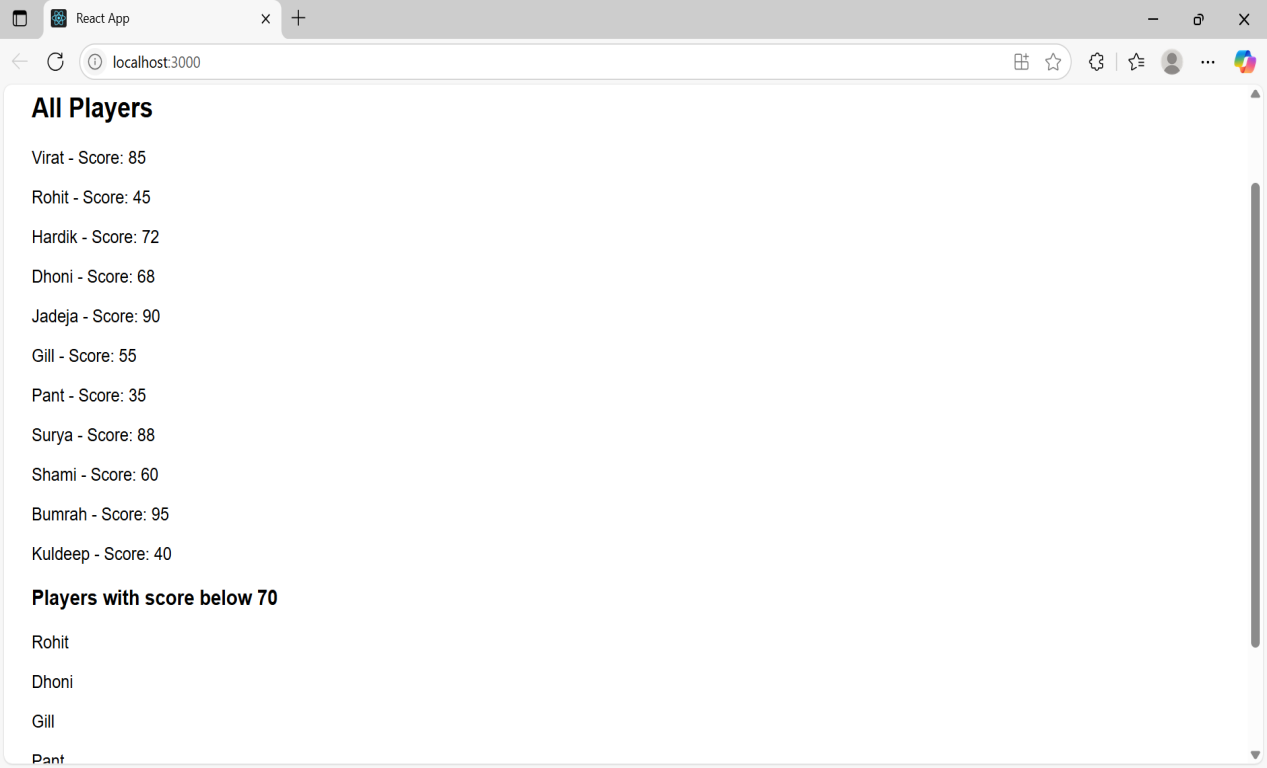
</div>

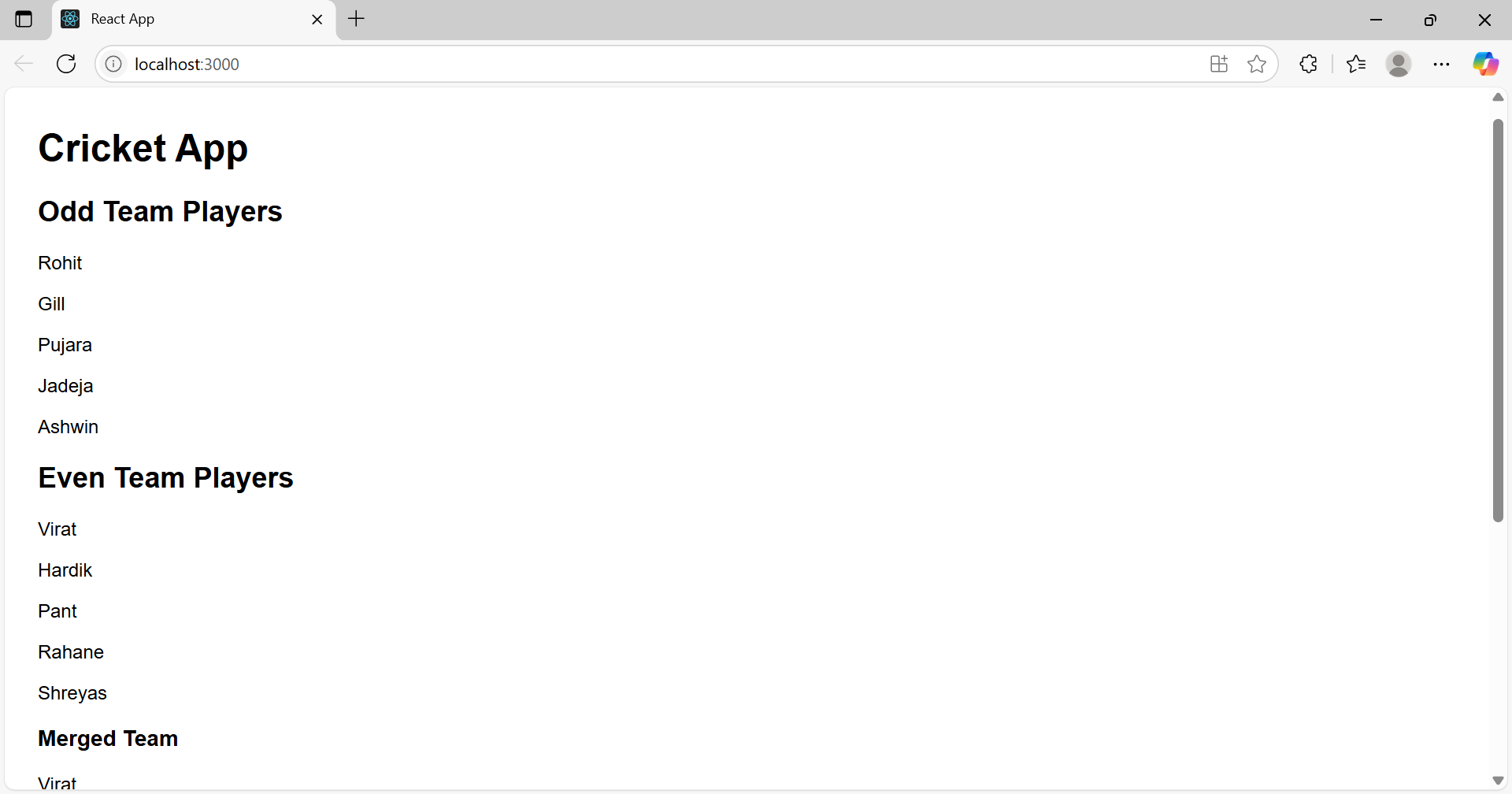
);

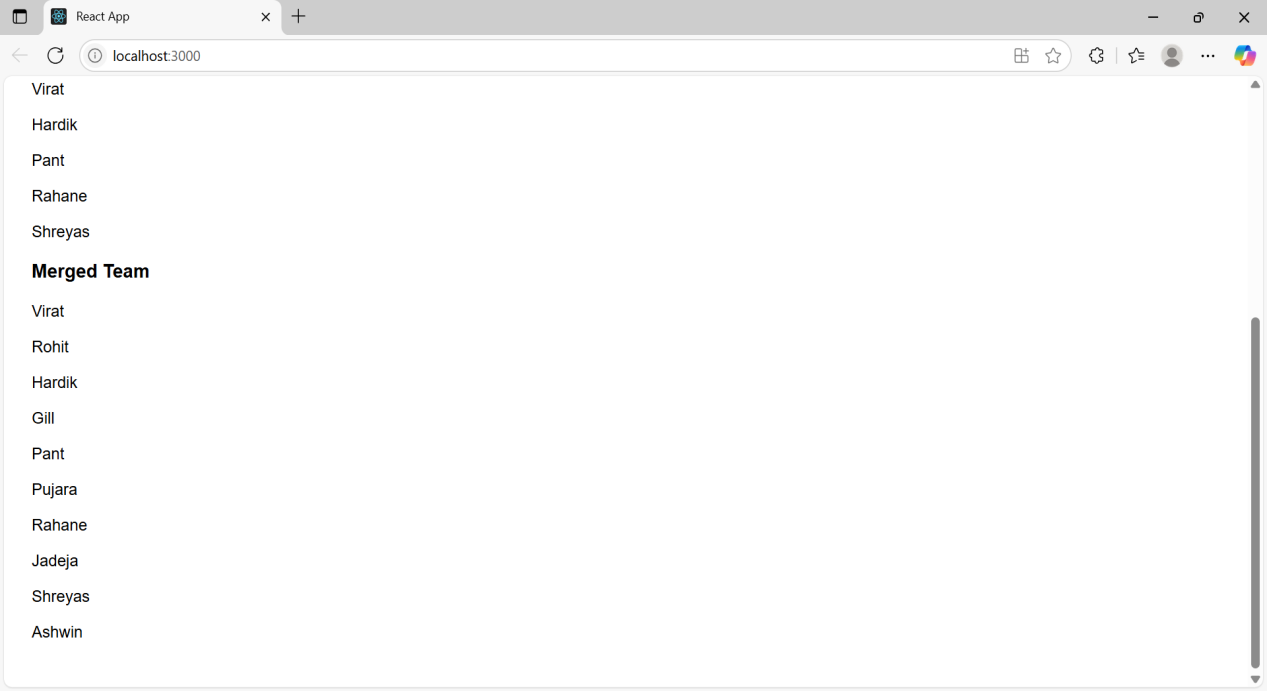
};

export default ListofPlayers;

**OUTPUT:**

****

****

****

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

**File name: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;

}

@keyframes App-logo-spin {

from {

transform: rotate(0deg);

}

to {

transform: rotate(360deg);

}

}

**File name: App.js**

// src/App.js

import React from 'react';

const App = () => {

const heading = <h1>Office Space Rental Portal</h1>;

const imageURL = "https://images.pexels.com/photos/37347/office-sitting-room-executive-sitting.jpg";

const office = {

name: "TechHub Workspace",

rent: 55000,

address: "Plot No. 23, HiTech City, Hyderabad"

};

const officeList = [

{ name: "Startup Bay", rent: 45000, address: "Madhapur, Hyderabad" },

{ name: "InnovateX", rent: 62000, address: "Gachibowli, Hyderabad" },

{ name: "CodeNest", rent: 58000, address: "Kondapur, Hyderabad" },

{ name: "DevSquare", rent: 70000, address: "Banjara Hills, Hyderabad" }

];

const getRentStyle = (rent) => ({

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

fontWeight: 'bold'

});

return (

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

{heading}

<img

src={imageURL}

alt="Office Space"

style={{ width: "400px", height: "200px", objectFit: "cover", borderRadius: "8px", marginBottom: "20px" }}

/>

<h2>Featured Office</h2>

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

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

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

<h2>Available Office Spaces</h2>

<ul>

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

<li key={index}>

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

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

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

<hr />

</li>

))}

</ul>

</div>

);

};

export default App;

**File name:index.css**

body {

margin: 0;

font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',

'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',

sans-serif;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

}

code {

font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',

monospace;

}

**File name: index.js**

// src/index.js

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

// 🧱 Rendering JSX to the DOM

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<App />);

**File name:index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<meta name="theme-color" content="#000000" />

<meta

name="description"

content="Web site created using create-react-app"

/>

<link rel="apple-touch-icon" href="%PUBLIC\_URL%/logo192.png" />

<!--

manifest.json provides metadata used when your web app is installed on a

user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/

-->

<link rel="manifest" href="%PUBLIC\_URL%/manifest.json" />

<!--

Notice the use of %PUBLIC\_URL% in the tags above.

It will be replaced with the URL of the `public` folder during the build.

Only files inside the `public` folder can be referenced from the HTML.

Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC\_URL%/favicon.ico" will

work correctly both with client-side routing and a non-root public URL.

Learn how to configure a non-root public URL by running `npm run build`.

-->

<title>React App</title>

</head>

<body>

<noscript>You need to enable JavaScript to run this app.</noscript>

<div id="root"></div>

<!--

This HTML file is a template.

If you open it directly in the browser, you will see an empty page.

You can add webfonts, meta tags, or analytics to this file.

The build step will place the bundled scripts into the <body> tag.

To begin the development, run `npm start` or `yarn start`.

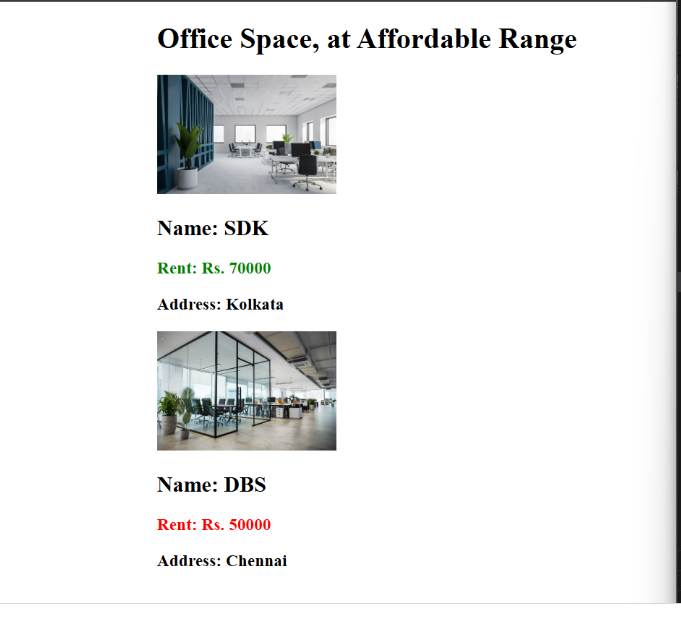
To create a production bundle, use `npm run build` or `yarn build`.

-->

</body>

</html>

**OUTPUT:**

****

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

**File name: App.js**

import React, { Component } from 'react';

import CurrencyConvertor from './CurrencyConvertor';

class App extends Component {

constructor(props) {

super(props);

this.state = {

count: 0

};

}

increment = () => {

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

alert("Hello member");

alert("This is a static message");

};

decrement = () => {

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

};

sayWelcome = () => {

alert("Welcome");

};

handleClick = () => {

alert("I was clicked");

};

render() {

return (

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

<h1>{this.state.count}</h1>

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

<button onClick={this.decrement} style={{ marginLeft: '10px' }}>Decrement</button>

<br /><br />

<button onClick={this.sayWelcome}>Say welcome</button>

<br /><br />

<button onClick={this.handleClick}>Click on me</button>

<CurrencyConvertor />

</div>

);

}

}

export default App;

**File name: 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;

}

@keyframes App-logo-spin {

from {

transform: rotate(0deg);

}

to {

transform: rotate(360deg);

}

}

**File name:index.js**

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<App />);

**File name:index.css**

body {

margin: 0;

font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',

'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',

sans-serif;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

}

code {

font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',

monospace;

}

**File name:index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<meta name="theme-color" content="#000000" />

<meta

name="description"

content="Web site created using create-react-app"

/>

<link rel="apple-touch-icon" href="%PUBLIC\_URL%/logo192.png" />

<!--

manifest.json provides metadata used when your web app is installed on a

user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/

-->

<link rel="manifest" href="%PUBLIC\_URL%/manifest.json" />

<!--

Notice the use of %PUBLIC\_URL% in the tags above.

It will be replaced with the URL of the `public` folder during the build.

Only files inside the `public` folder can be referenced from the HTML.

Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC\_URL%/favicon.ico" will

work correctly both with client-side routing and a non-root public URL.

Learn how to configure a non-root public URL by running `npm run build`.

-->

<title>React App</title>

</head>

<body>

<noscript>You need to enable JavaScript to run this app.</noscript>

<div id="root"></div>

<!-- This HTML file is a template.

If you open it directly in the browser, you will see an empty page.

You can add webfonts, meta tags, or analytics to this file.

The build step will place the bundled scripts into the <body> tag.

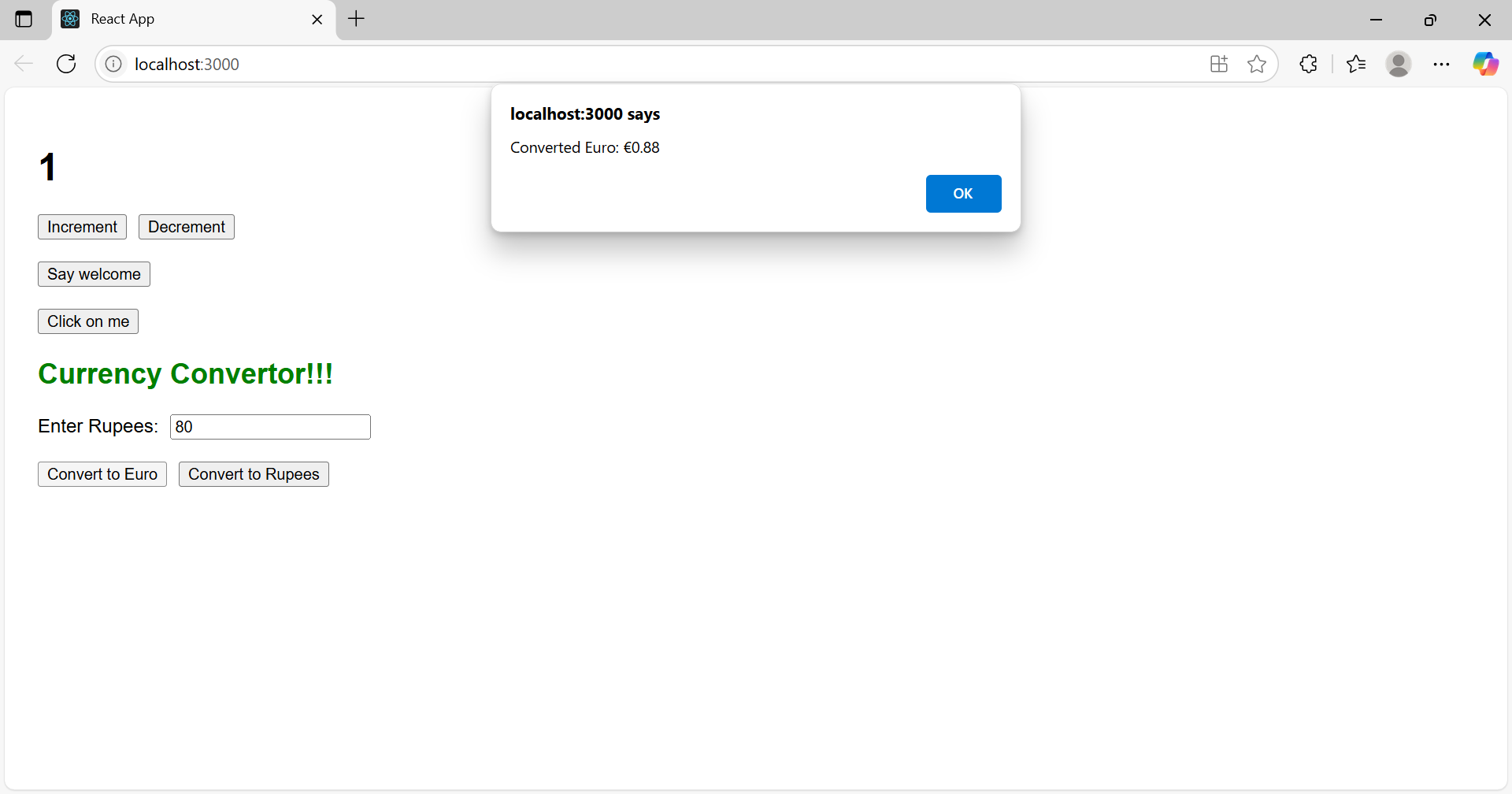
To begin the development, run `npm start` or `yarn start`.

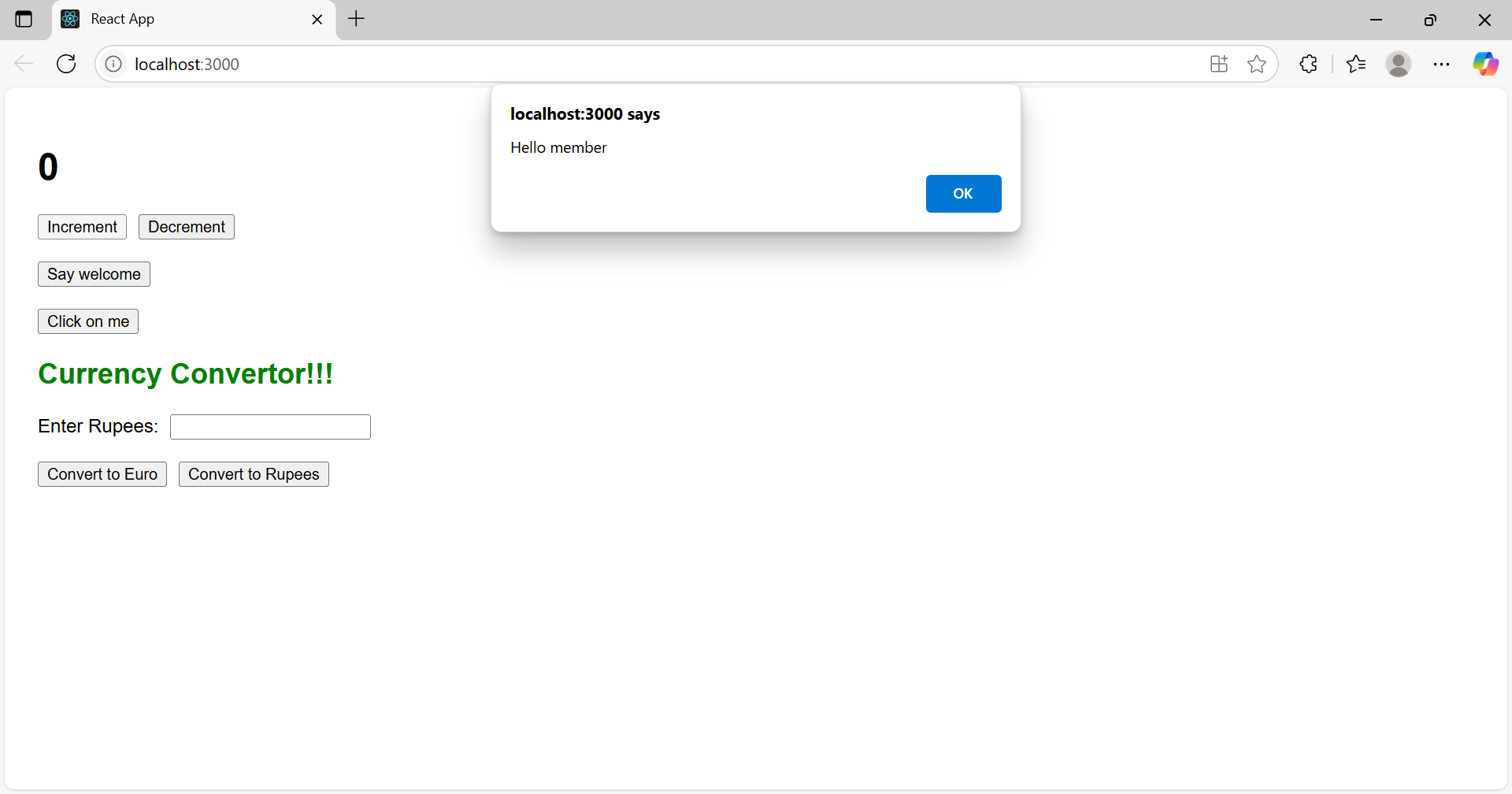
To create a production bundle, use `npm run build` or `yarn build`. -->

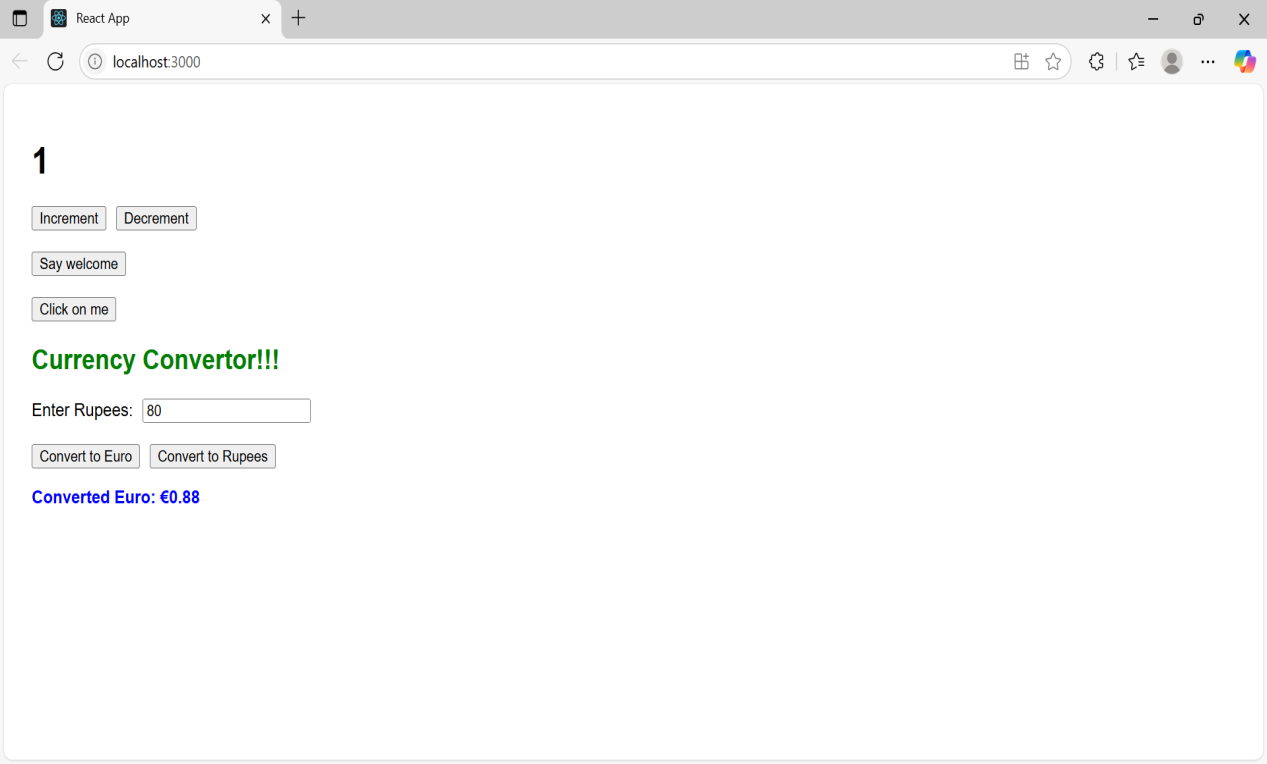
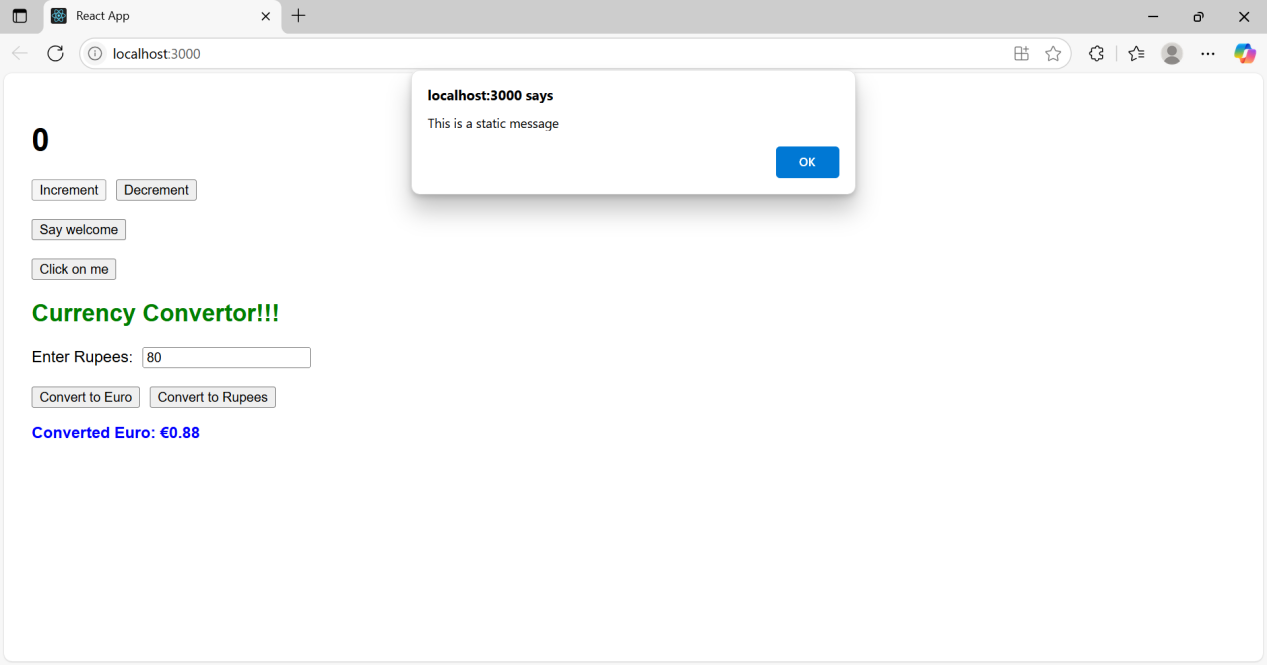
</body>

</html>

**OUTPUT:**

****

****

****

**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.**

**File name:App.js**

import React, { Component } from 'react';

class App extends Component {

constructor(props) {

super(props);

this.state = {

isLoggedIn: false

};

}

handleLogin = () => {

this.setState({ isLoggedIn: true });

};

handleLogout = () => {

this.setState({ isLoggedIn: false });

};

render() {

let message;

let button;

let signupPrompt;

if (this.state.isLoggedIn) {

message = <h2>Welcome User</h2>;

button = <button onClick={this.handleLogout}>Logout</button>;

signupPrompt = null;

} else {

message = <h2>Welcome Guest</h2>;

signupPrompt = <p style={{ color: 'red' }}>Please sign up to book your tickets.</p>;

button = <button onClick={this.handleLogin}>Login</button>;

}

return (

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

<h1>Ticket Booking App</h1>

{message}

{signupPrompt}

{button}

<br /><br />

{this.state.isLoggedIn ? (

<p>You can now book tickets</p>

) : (

<div>

<h3>Flight Details</h3>

<ul>

<li>Flight: AI202</li>

<li>From: Hyderabad</li>

<li>To: Delhi</li>

<li>Departure: 10:00 AM</li>

</ul>

</div>

)}

</div>

);

}

}

export default App;

**File name: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;

}

@keyframes App-logo-spin {

from {

transform: rotate(0deg);

}

to {

transform: rotate(360deg);

}

}

**File name:index.js**

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<App />);

**File name:index.css**

body {

margin: 0;

font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',

'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',

sans-serif;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

}

code {

font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',

monospace;

}

**File name:index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<meta name="theme-color" content="#000000" />

<meta

name="description"

content="Web site created using create-react-app"

/>

<link rel="apple-touch-icon" href="%PUBLIC\_URL%/logo192.png" />

<!--

manifest.json provides metadata used when your web app is installed on a

user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/

-->

<link rel="manifest" href="%PUBLIC\_URL%/manifest.json" />

<!--

Notice the use of %PUBLIC\_URL% in the tags above.

It will be replaced with the URL of the `public` folder during the build.

Only files inside the `public` folder can be referenced from the HTML.

Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC\_URL%/favicon.ico" will

work correctly both with client-side routing and a non-root public URL.

Learn how to configure a non-root public URL by running `npm run build`.

-->

<title>React App</title>

</head>

<body>

<noscript>You need to enable JavaScript to run this app.</noscript>

<div id="root"></div>

<!--

This HTML file is a template.

If you open it directly in the browser, you will see an empty page.

You can add webfonts, meta tags, or analytics to this file.

The build step will place the bundled scripts into the <body> tag.

To begin the development, run `npm start` or `yarn start`.

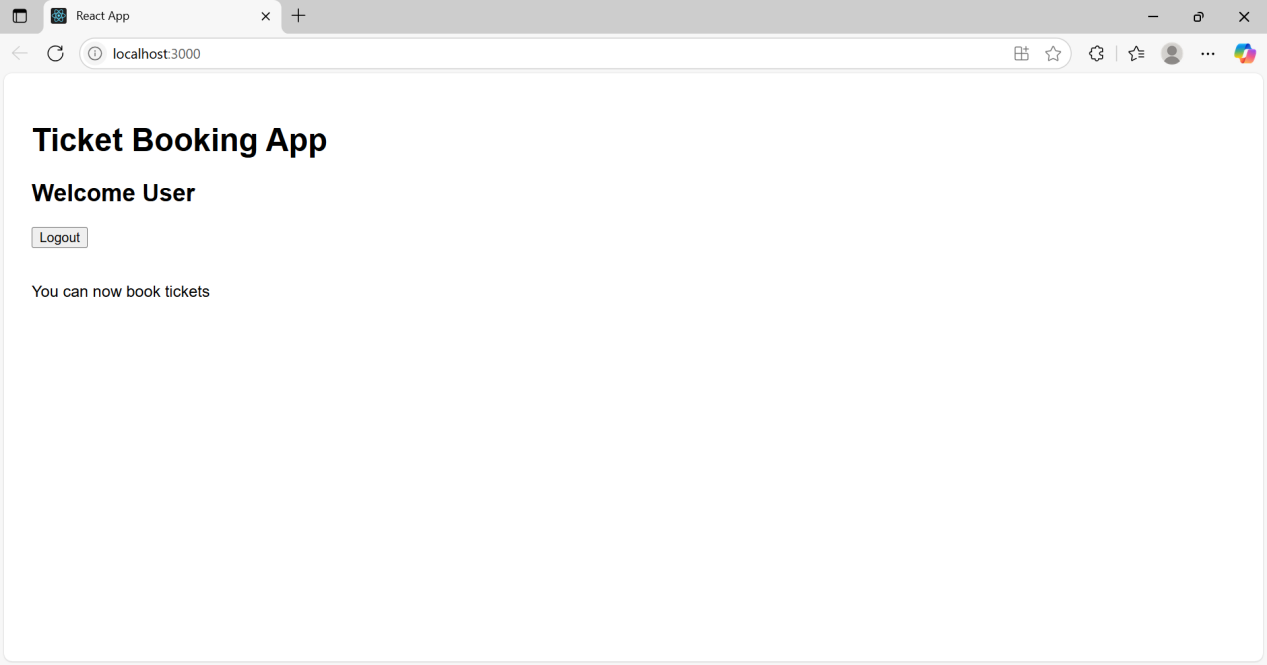
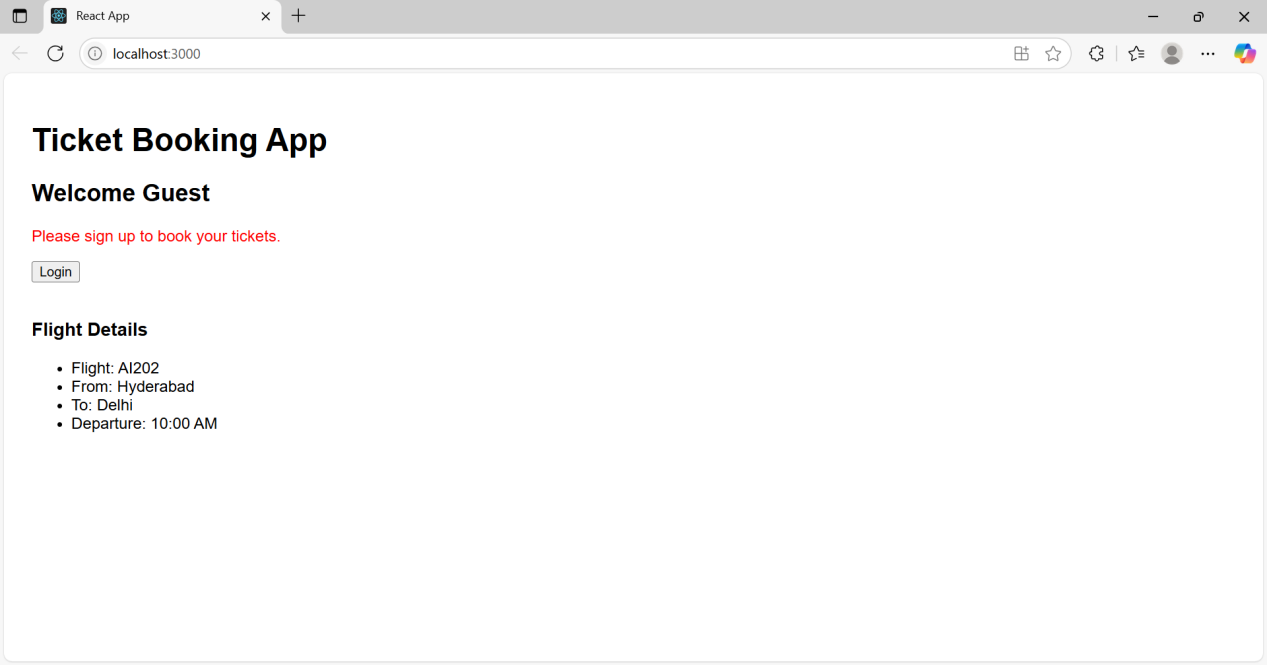
To create a production bundle, use `npm run build` or `yarn build`.

-->

</body>

</html>

**OUTPUT:**

****

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

**File name:App.js**

import React from 'react';

import BookDetails from './BookDetails';

import BlogDetails from './BlogDetails';

import CourseDetails from './CourseDetails';

const App = () => {

return (

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

<h1>Welcome to Blogger App</h1>

<div style={{ display: 'flex', justifyContent: 'space-between' }}>

<CourseDetails />

<BookDetails />

<BlogDetails />

</div>

</div>

);

};

export default App;

**File name: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;

}

@keyframes App-logo-spin {

from {

transform: rotate(0deg);

}

to {

transform: rotate(360deg);

}

}

**File name:index.js**

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<App />);

**File name:index.css**

body {

margin: 0;

font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',

'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',

sans-serif;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

}

code {

font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',

monospace;

}

**File name:index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<meta name="theme-color" content="#000000" />

<meta

name="description"

content="Web site created using create-react-app"

/>

<link rel="apple-touch-icon" href="%PUBLIC\_URL%/logo192.png" />

<!--

manifest.json provides metadata used when your web app is installed on a

user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/

-->

<link rel="manifest" href="%PUBLIC\_URL%/manifest.json" />

<!--

Notice the use of %PUBLIC\_URL% in the tags above.

It will be replaced with the URL of the `public` folder during the build.

Only files inside the `public` folder can be referenced from the HTML.

Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC\_URL%/favicon.ico" will

work correctly both with client-side routing and a non-root public URL.

Learn how to configure a non-root public URL by running `npm run build`.

-->

<title>React App</title>

</head>

<body>

<noscript>You need to enable JavaScript to run this app.</noscript>

<div id="root"></div>

<!--

This HTML file is a template.

If you open it directly in the browser, you will see an empty page.

You can add webfonts, meta tags, or analytics to this file.

The build step will place the bundled scripts into the <body> tag.

To begin the development, run `npm start` or `yarn start`.

To create a production bundle, use `npm run build` or `yarn build`.

-->

</body>

</html>

**File name:CourseDetails.js**

import React from 'react';

const CourseDetails = () => {

const courses = [

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

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

];

return (

<div style={{ width: '30%' }}>

<h2>Course Details</h2>

<ul>

{courses.map(course => (

<li key={course.id}>

<strong>{course.name}</strong><br />

{course.date}

</li>

))}

</ul>

</div>

);

};

export default CourseDetails;

**File name:BlogDetails.js**

import React from 'react';

const BlogDetails = () => {

const blogs = [

{

id: 1,

title: 'React Learning',

author: 'Stephen Biz',

description: 'Welcome to learning React!'

},

{

id: 2,

title: 'Installation',

author: 'Schwezdnier',

description: 'You can install React from npm.'

}

];

return (

<div style={{

width: '30%',

borderLeft: '3px solid green',

paddingLeft: '15px'

}}>

<h2>Blog Details</h2>

<ul>

{blogs.map(blog => (

<div key={blog.id}>

<h3>{blog.title}</h3>

<p><strong>{blog.author}</strong></p>

<p>{blog.description}</p>

</div>

))}

</ul>

</div>

);

};

export default BlogDetails;

**File name:BookDetails.js**

import React from 'react';

const BookDetails = () => {

const books = [

{ id: 1, bname: 'Master React', price: 670 },

{ id: 2, bname: 'Deep Dive into Angular 11', price: 800 },

{ id: 3, bname: 'Mongo Essentials', price: 450 }

];

return (

<div style={{

width: '30%',

borderLeft: '3px solid green',

paddingLeft: '15px'

}}>

<h2>Book Details</h2>

<ul>

{books.map(book => (

<div key={book.id}>

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

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

</div>

))}

</ul>

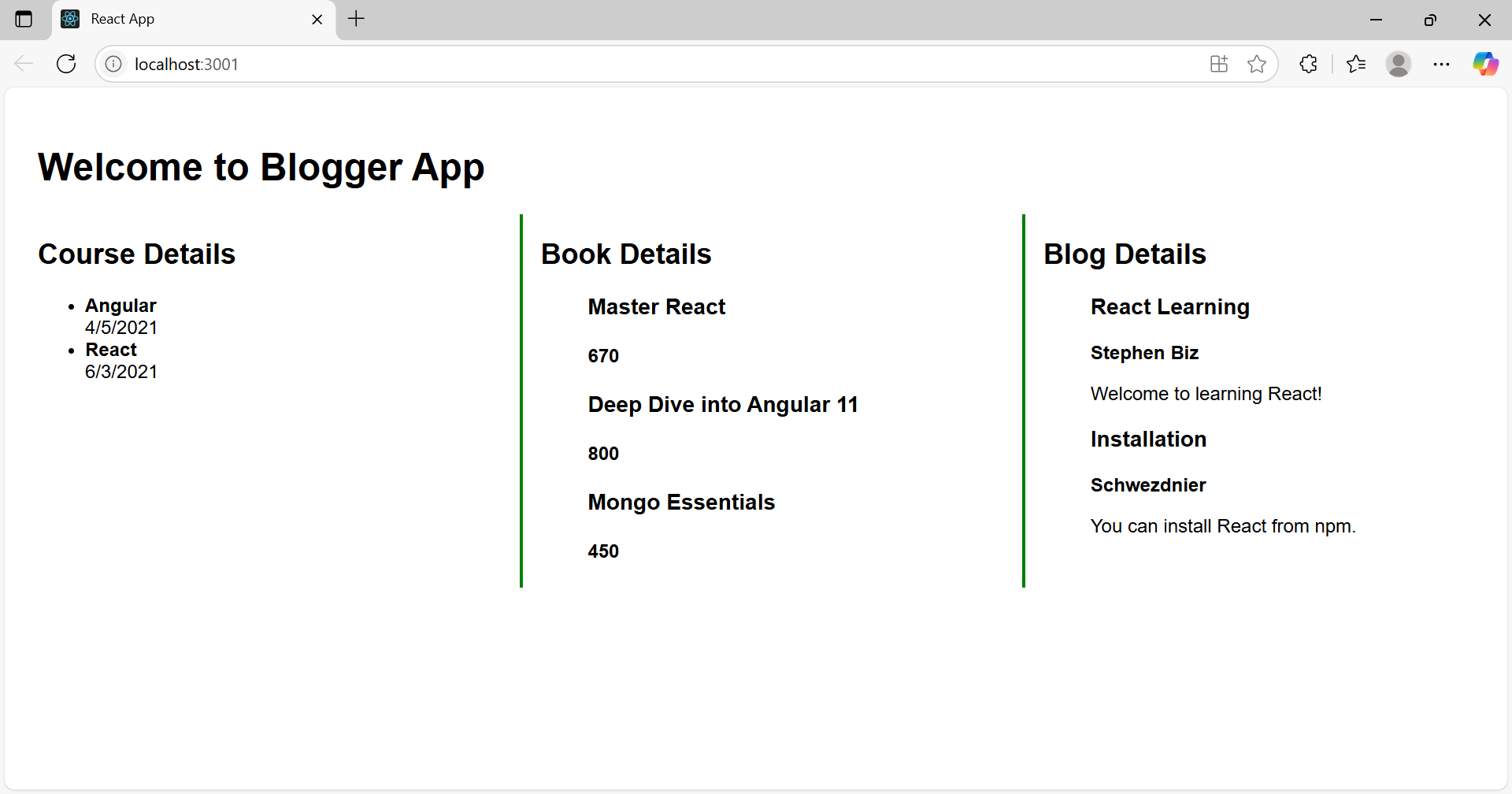
</div>

);

};

export default BookDetails;

**OUTPUT:**

****