**Cognizant Digital Nurture 4.0**

**Name: Siri Chandana Chittipolu**

**Email:** [**sirichittipolu11@gmail.com**](mailto:sirichittipolu11@gmail.com)

**Superset ID: 6386277**

**Mandatory Hands-On Exercises**

1. **ReactJS-HOL:**

**Solution:**

**Steps:**

* Created React app using npx create-react-app cricketapp and set up component folder inside src/components.
* Defined a players array of 11 objects with name and score inside App.js.

**//App.js**

import React from "react";

import ListofPlayers from "./components/ListofPlayers";

import Scorebelow70 from "./components/Scorebelow70";

import { OddPlayers } from "./components/OddPlayers";

import { EvenPlayers } from "./components/EvenPlayers";

import ListofIndianPlayers from "./components/ListofIndianPlayers";

function App() {

  const flag = true;

  const players = [

    { name: "Mr. Jack", score: 50 },

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

    { name: "Mr. John", score: 40 },

    { name: "Mr. Alan", score: 61 },

    { name: "Ms. Elizabeth", score: 61 },

    { name: "Ms. Sarah", score: 90 },

    { name: "Mr. Black", score: 60 },

    { name: "Ms. Vani", score: 74 },

    { name: "Mr. Rakesh", score: 85 },

    { name: "Mr. Ramu", score: 75 },

    { name: "Ms. Jalsa", score: 64 },

  ];

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

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

  const IndianTeam = [...T20Players, ...RanjiTrophyPlayers];

  return (

    <div>

      {flag === true ? (

        <div>

          <h1>List of Players</h1>

          <ListofPlayers players={players} />

          <hr />

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

          <Scorebelow70 players={players} />

        </div>

      ) : (

        <div>

          <div>

            <h1>Indian Team</h1>

            <h1>Odd Players</h1>

            <OddPlayers players={IndianTeam} />

            <hr />

            <h1>Even Players</h1>

            <EvenPlayers players={IndianTeam} />

          </div>

          <hr />

          <div>

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

            <ListofIndianPlayers IndianPlayers={IndianTeam} />

          </div>

        </div>

      )}

    </div>

  );

}

export default App;

* Used map() and filter() to display:
  + All players (ListofPlayers)

**//ListofPlayers**

function ListofPlayers({ players }) {

  return (

    <ul>

      {players.map((item, index) => (

        <li key={index}>

          {item.name} - {item.score}

        </li>

      ))}

    </ul>

  );

}

export default ListofPlayers;

* + Players with score less than 70 (Scorebelow70)

**//Scorebelow70.js**

function Scorebelow70({ players }) {

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

  return (

    <ul>

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

        <li key={index}>

          {item.name} - {item.score}

        </li>

      ))}

    </ul>

  );

}

export default Scorebelow70;

* Used array destructuring in:
  + OddPlayers.js to show 1st, 3rd, 5th players

**//OddPlayers.js**

export function OddPlayers({ players }) {

  const [first, , third, , fifth] = players;

  return (

    <ul>

      <li>First: {first}</li>

      <li>Third: {third}</li>

      <li>Fifth: {fifth}</li>

    </ul>

  );

}

* + EvenPlayers.js to show 2nd, 4th, 6th players

**//EvenPlayers.js**

export function EvenPlayers({ players }) {

  const [, second, , fourth, , sixth] = players;

  return (

    <ul>

      <li>Second: {second}</li>

      <li>Fourth: {fourth}</li>

      <li>Sixth: {sixth}</li>

    </ul>

  );

}

* Merged T20 and Ranji players using spread operator into a single IndianTeam array.
* Displayed Indian players list using ListofIndianPlayers.js.

**//ListofIndianPlayers.js**

function ListofIndianPlayers({ IndianPlayers }) {

  return (

    <ul>

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

        <li key={index}>{player}</li>

      ))}

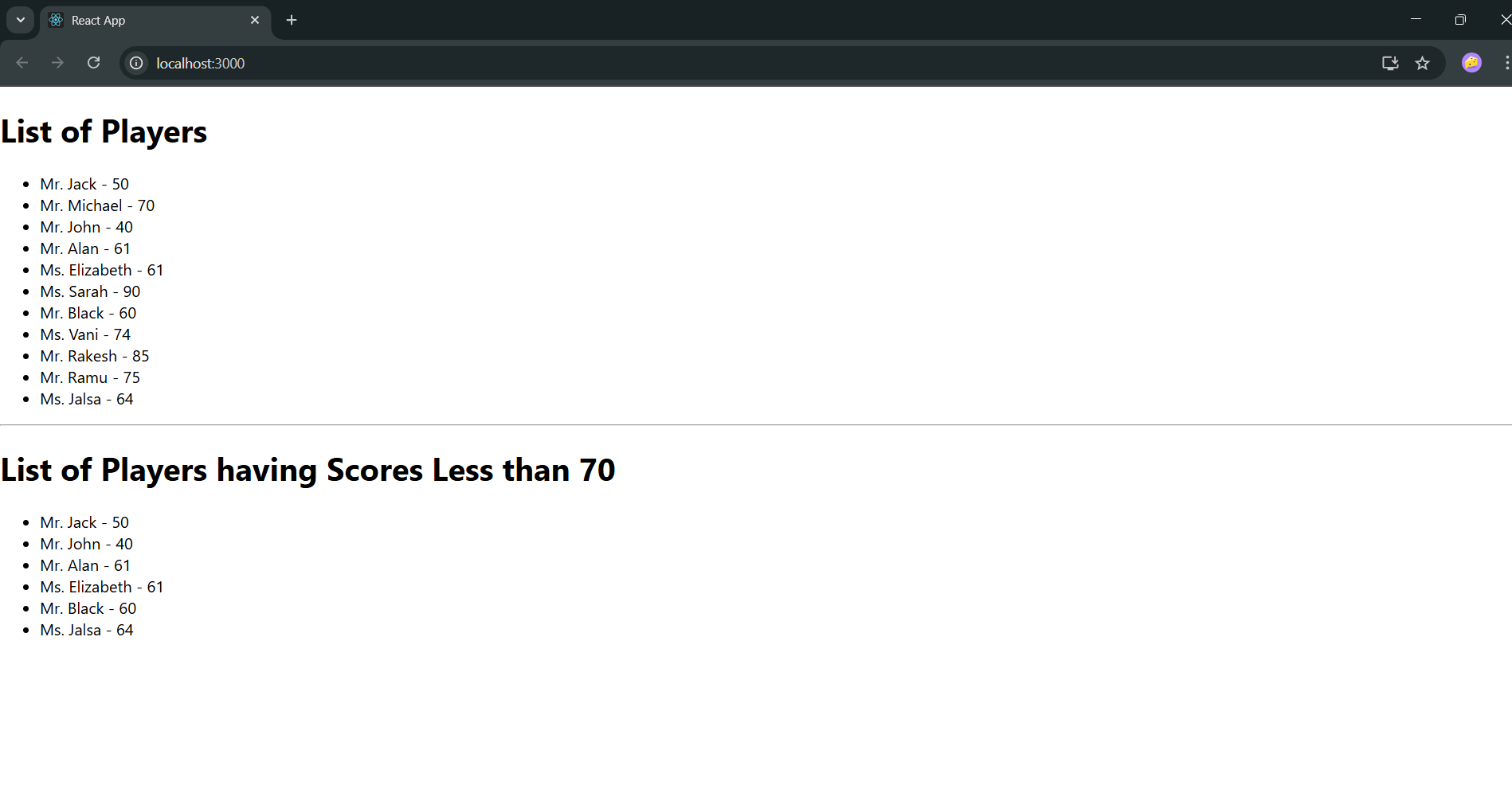
    </ul>

  );

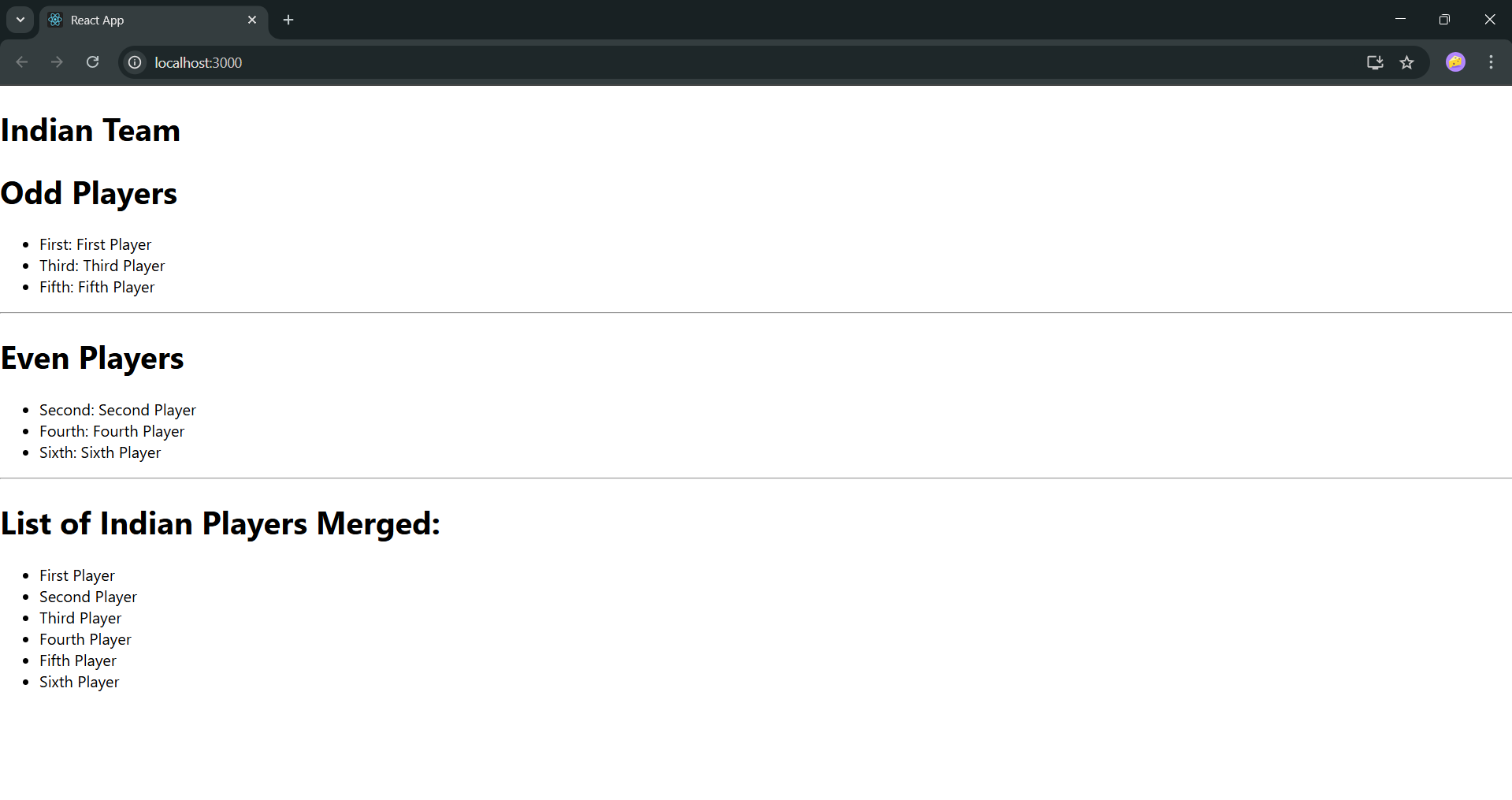
}

export default ListofIndianPlayers;

* Used a flag variable to conditionally render:
  + When true: player list and score < 70

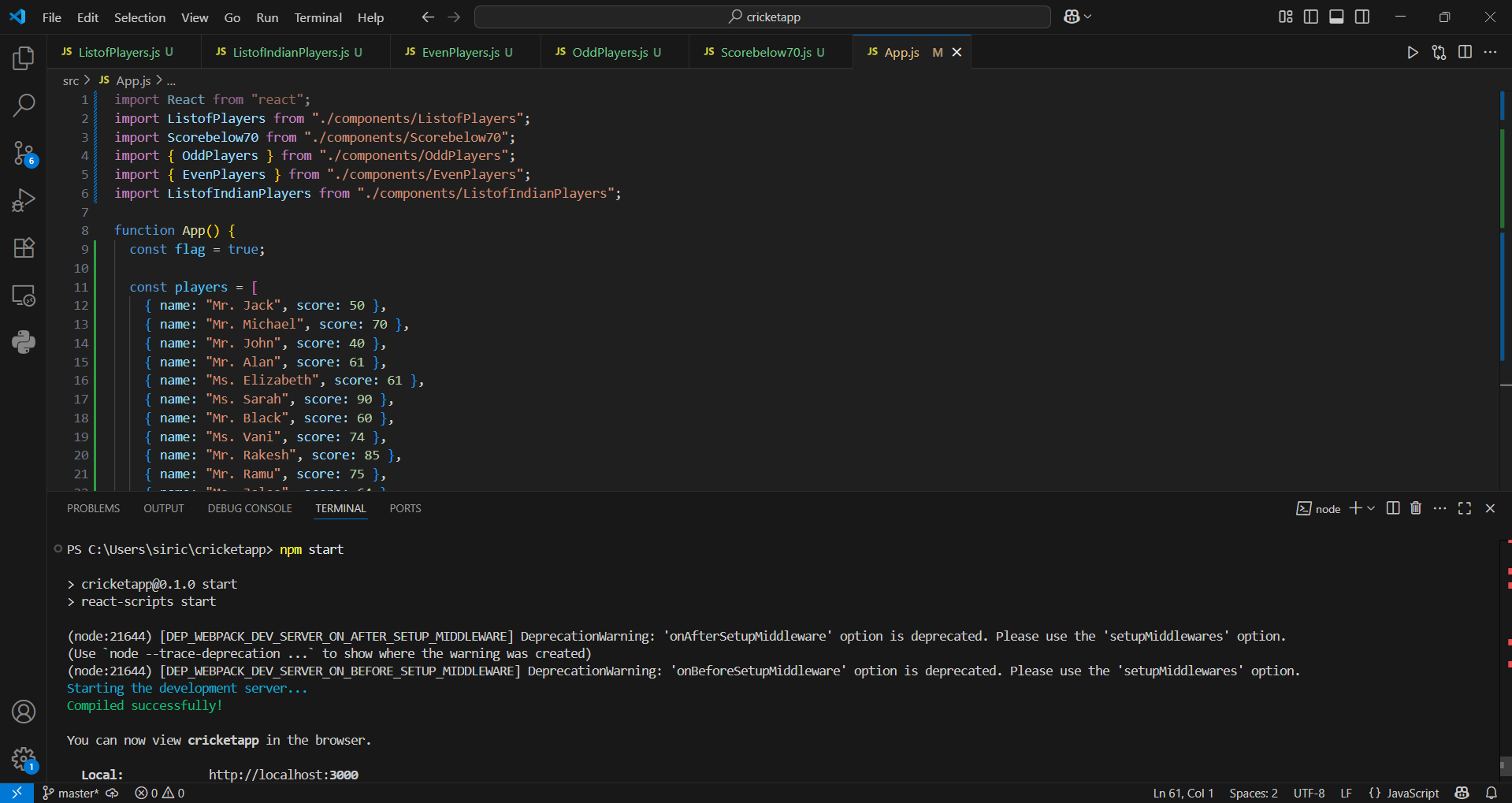
****

* + When false: odd/even + merged Indian team



* Passed props properly and handled destructuring cleanly inside the components.

**Output:**



**10. ReactJS-HOL:**

**Solution:**

**Steps:**

1. Created a React app using npx create-react-app officespacerentalapp.

1. Defined JSX elements for heading, image, and office details like Name, Rent, and Address.
2. Used a JavaScript object to store office details and displayed them using JSX.
3. Added inline CSS to change Rent color:

* Red if Rent ≤ 60000
* Green if Rent > 60000

1. Created an array of multiple office objects and used .map() to loop through and display all office cards.
2. Used valid image URLs for each office to show the space clearly.
3. Styled the output using simple JSX and className to align layout properly.

**//App.js**

import React from "react";

import './App.css';

function App() {

  const officeSpaces = [

    { Name: "DBS", Rent: 50000, Address: "Chennai", img: "https://addressadvisors.com/templates/yootheme/cache/c3b7e4.webp" },

    { Name: "WeWork", Rent: 70000, Address: "Mumbai", img: "https://cdn.sanity.io/images/uqxwe2qj/production/2e7c4475e431d5a82dc3bfbf06477abd87531d8d-943x49&auto=format&fit=fill&w=943" },

    { Name: "Regus", Rent: 60000, Address: "Delhi", img: "https://etimg.etb2bimg.com/photo/53061565.cms" },

  ];

  return (

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

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

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

        let rentColor = item.Rent <= 60000 ? "red" : "green";

        return (

          <div key={index} style={{ marginBottom: "30px" }}>

            <img

              src={item.img}

              alt="Office Space"

              width="25%"

              height="25%"

              style={{ borderRadius: "10px" }}

            />

            <h2>Name: {item.Name}</h2>

            <h3 style={{ color: rentColor }}>Rent: Rs. {item.Rent}</h3>

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

          </div>

        );

      })}

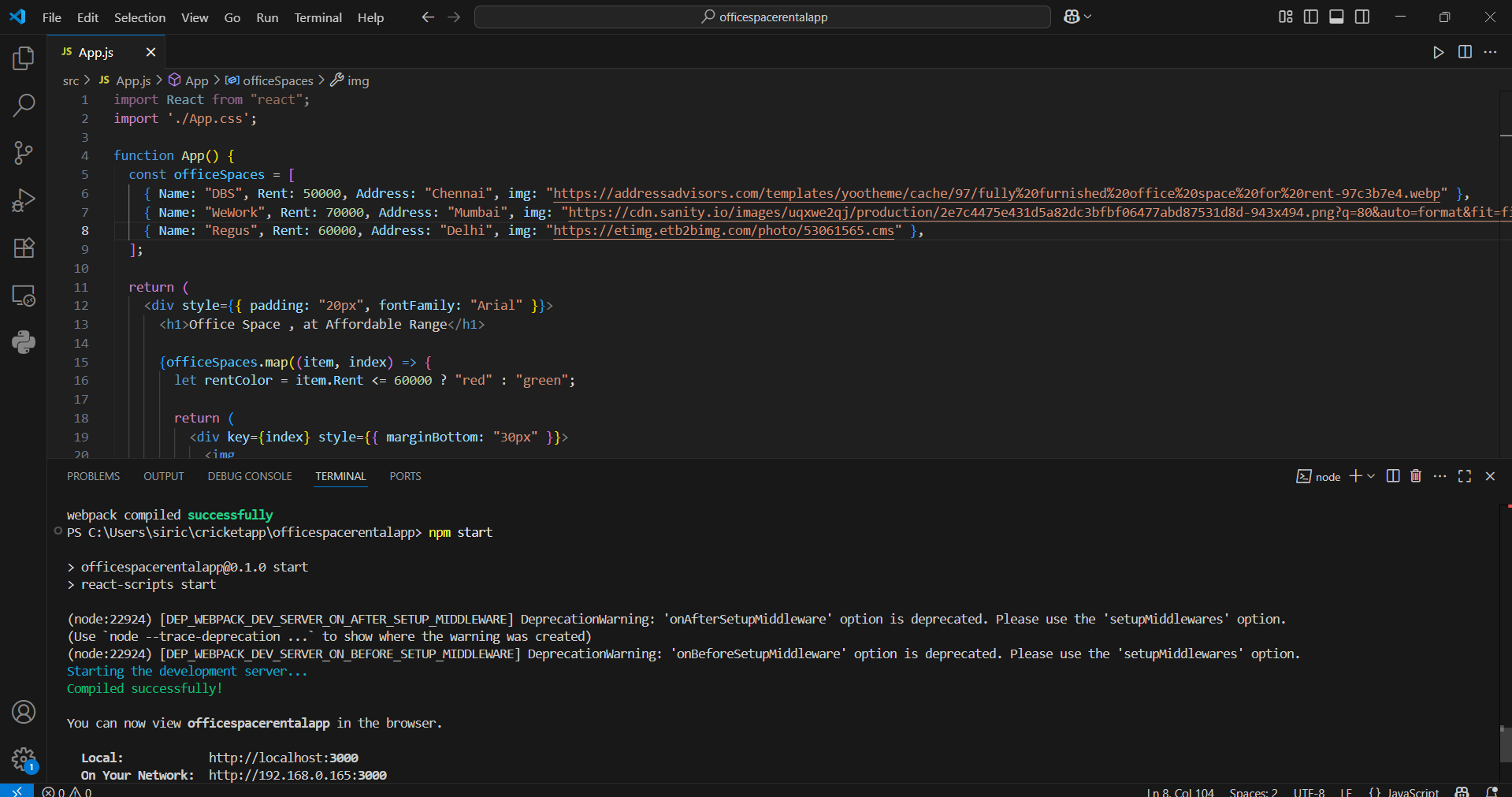
    </div>

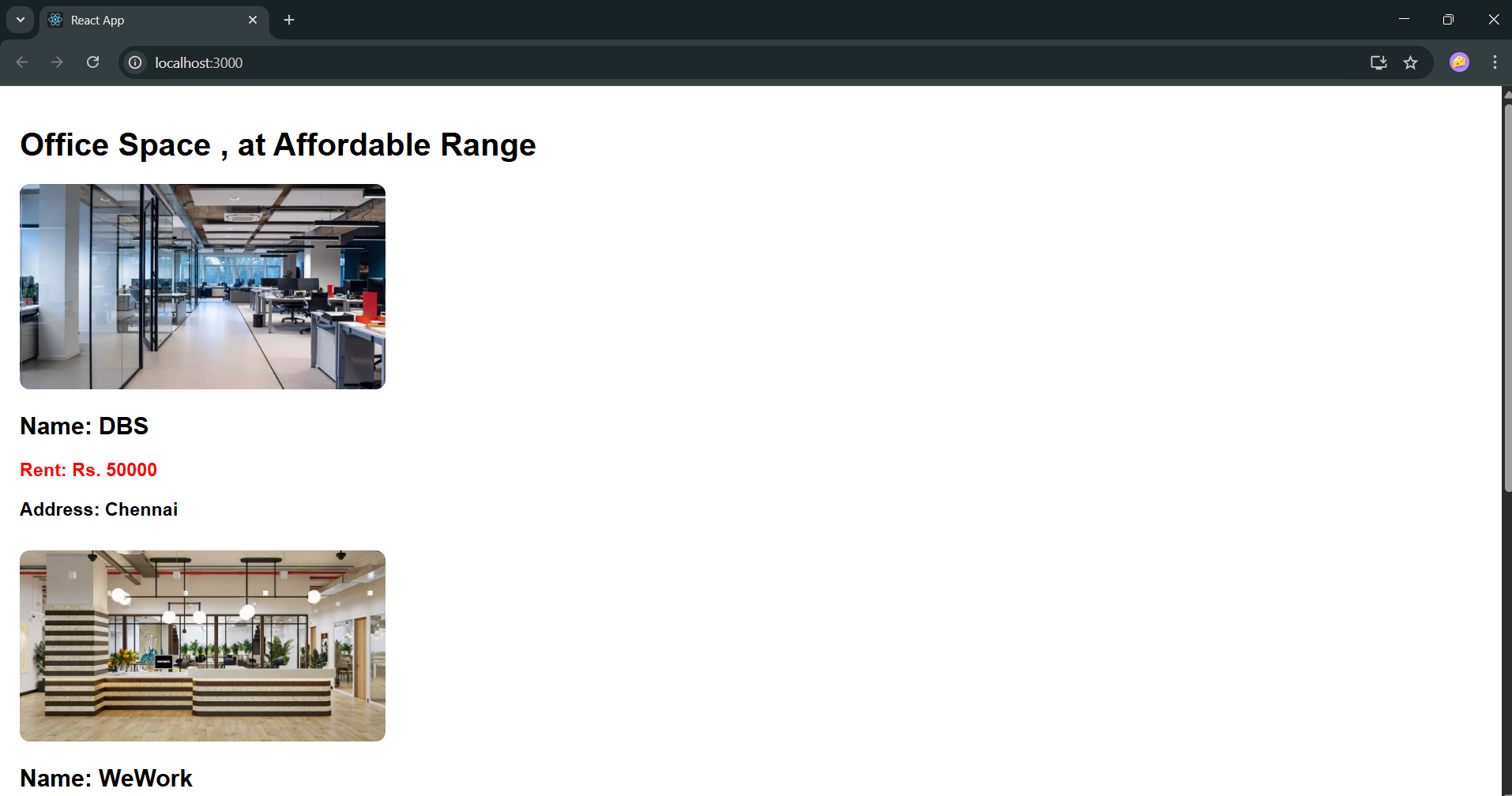
  );

}

export default App;

**Output:**

****

****

**11. ReactJS-HOL:**

**Solution:**

**Steps:**

1. Set up the project:

* Created a React app using npx create-react-app eventexamplesapp and opened it in VS Code.

1. Created CurrencyConverter component:

* Used a class-based component.
* Added amount, currency, and result to the state.

1. Handled input changes:

* Used a single handleChange method to update values based on input.

1. Handled form submit:

* On submit, used handleSubmit to:
  1. Prevent default action.
  2. Convert INR to Euro using a fixed rate (1 INR = 0.011 EUR).
  3. Show the result in an alert popup using alert().

1. Integrated into App.js:

* Imported and used <CurrencyConverter /> below the other buttons.

1. Tested the app:

* Ran npm start and checked that the conversion shows up in an alert popup when I click Submit.

**//App.js**

import React, { Component } from 'react';

import CurrencyConverter from './CurrencyConverter';

class App extends Component {

  constructor(props) {

    super(props);

    this.state = {

      count: 5,

    };

  }

  increment = () => {

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

    this.sayHello();

  }

  sayHello = () => {

    alert("Hello from React!");

  }

  decrement = () => {

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

  }

  sayWelcome = (message) => {

    alert(message);

  }

  handleClick = (event) => {

    alert("I was clicked");

    console.log("Synthetic event type:", event.type); // Using SyntheticEvent

  }

  render() {

    return (

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

        <p>{this.state.count}</p>

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

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

        <br /><br />

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

        <br /><br />

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

        <h2 style={{ color: 'green' }}>Currency Convertor!!!</h2>

        <CurrencyConverter />

      </div>

    );

}

export default App;

**//CurrencyConverter.js**

import React, { Component } from 'react';

class CurrencyConverter extends Component {

  constructor(props) {

    super(props);

    this.state = {

      amount: '',

      currency: '',

      result: null

    };

  }

  handleChange = (event) => {

    this.setState({ [event.target.name]: event.target.value });

  }

  handleSubmit = (event) => {

    event.preventDefault();

    const { amount } = this.state;

    const euroRate = 0.011; // 1 INR = 0.011 EUR

    if (!amount || isNaN(amount)) {

      alert("Please enter a valid numeric amount.");

      return;

    }

    const result = amount \* euroRate;

    alert(`Converted ${amount} INR to ${result.toFixed(2)} EUR`);

    this.setState({ result }); // Optional if you want to reuse result elsewhere

  }

  render() {

    return (

      <div>

        <form onSubmit={this.handleSubmit}>

          <label>

            Amount:

            <input

              type="text"

              name="amount"

              value={this.state.amount}

              onChange={this.handleChange}

            />

          </label>

          <br />

          <label>

            Currency:

            <input

              type="text"

              name="currency"

              value={this.state.currency}

              onChange={this.handleChange}

              placeholder="Enter 'Euro'"

            />

          </label>

          <br />

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

        </form>

      </div>

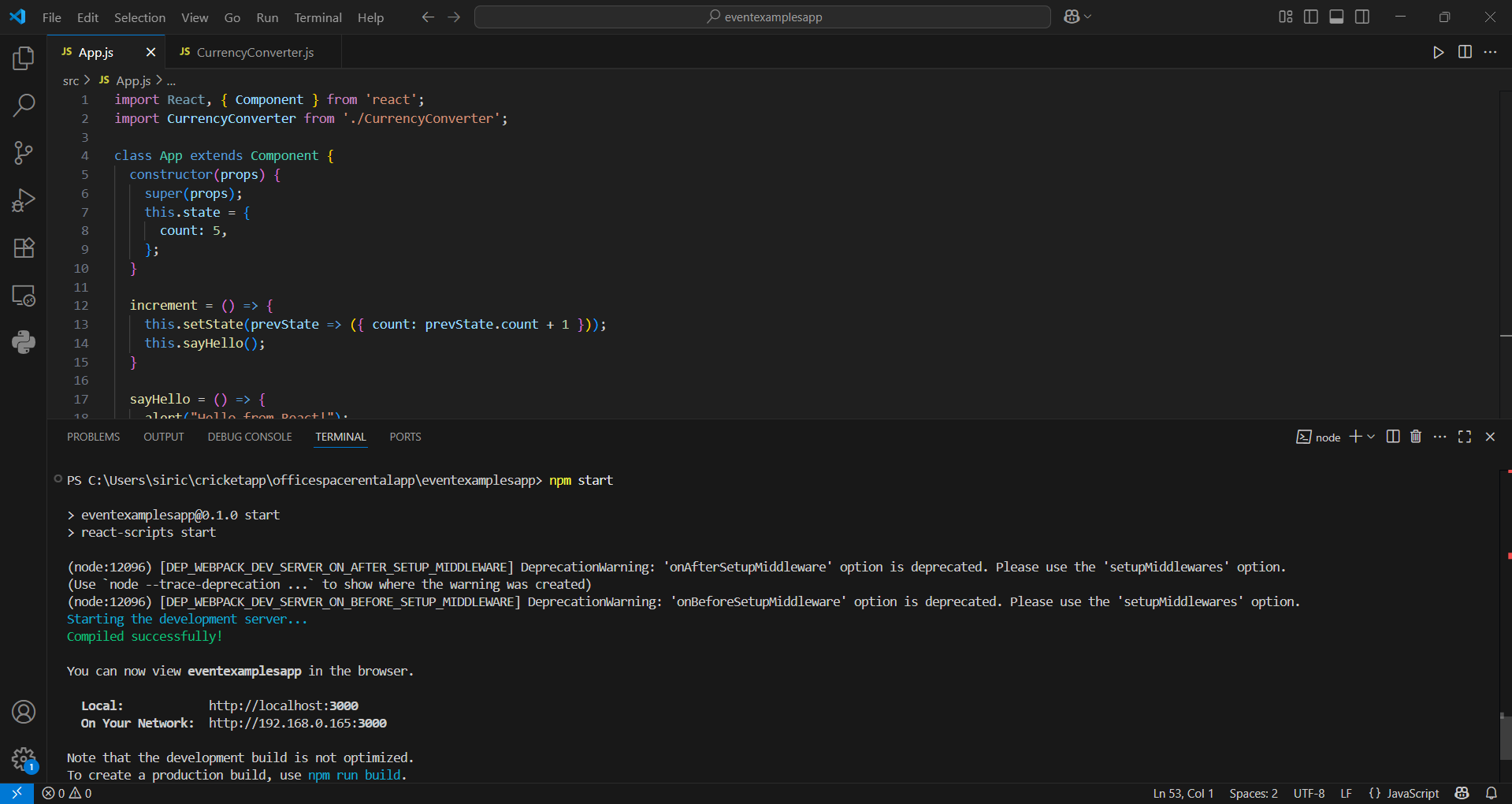
    );

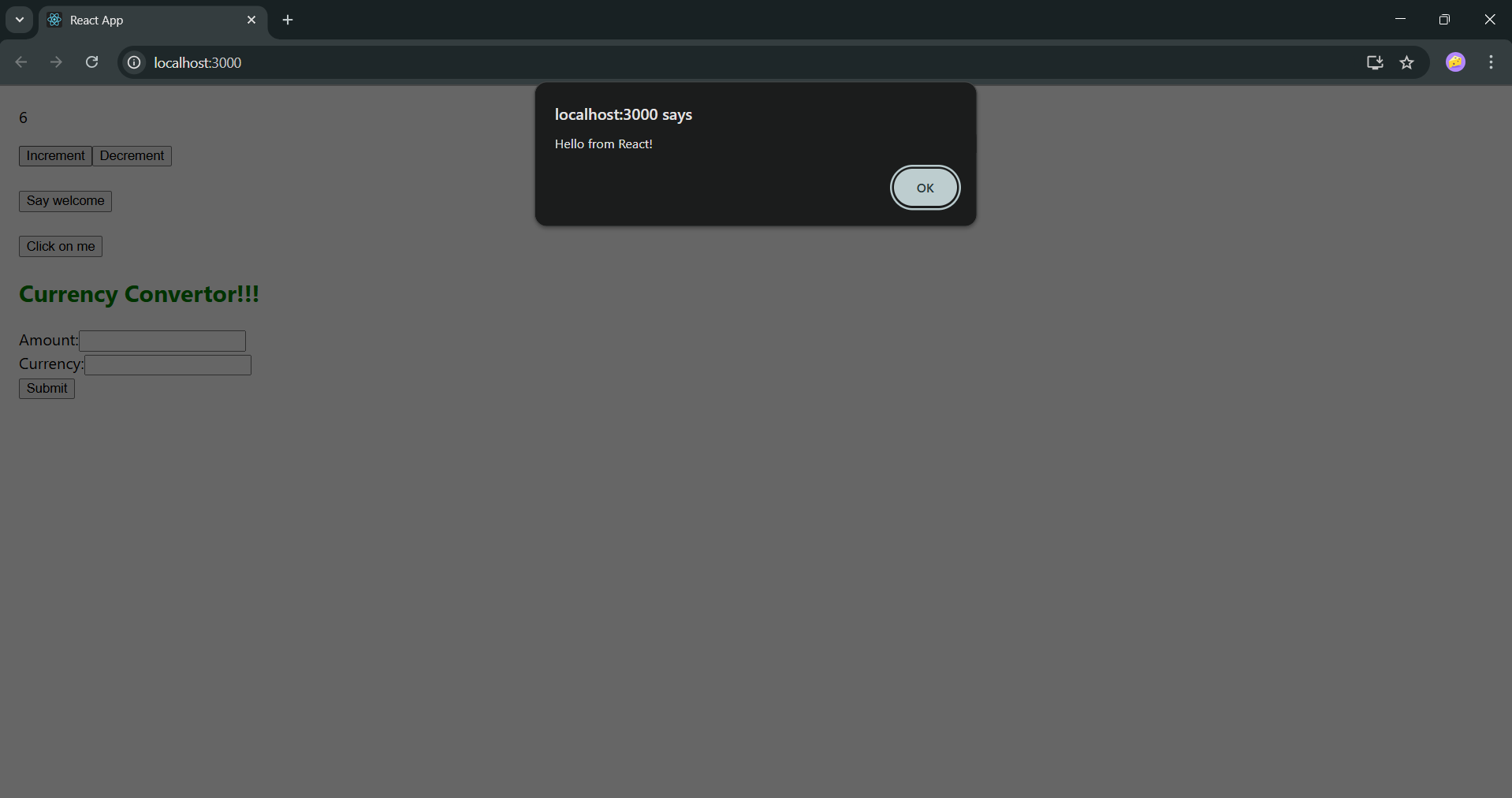
  }

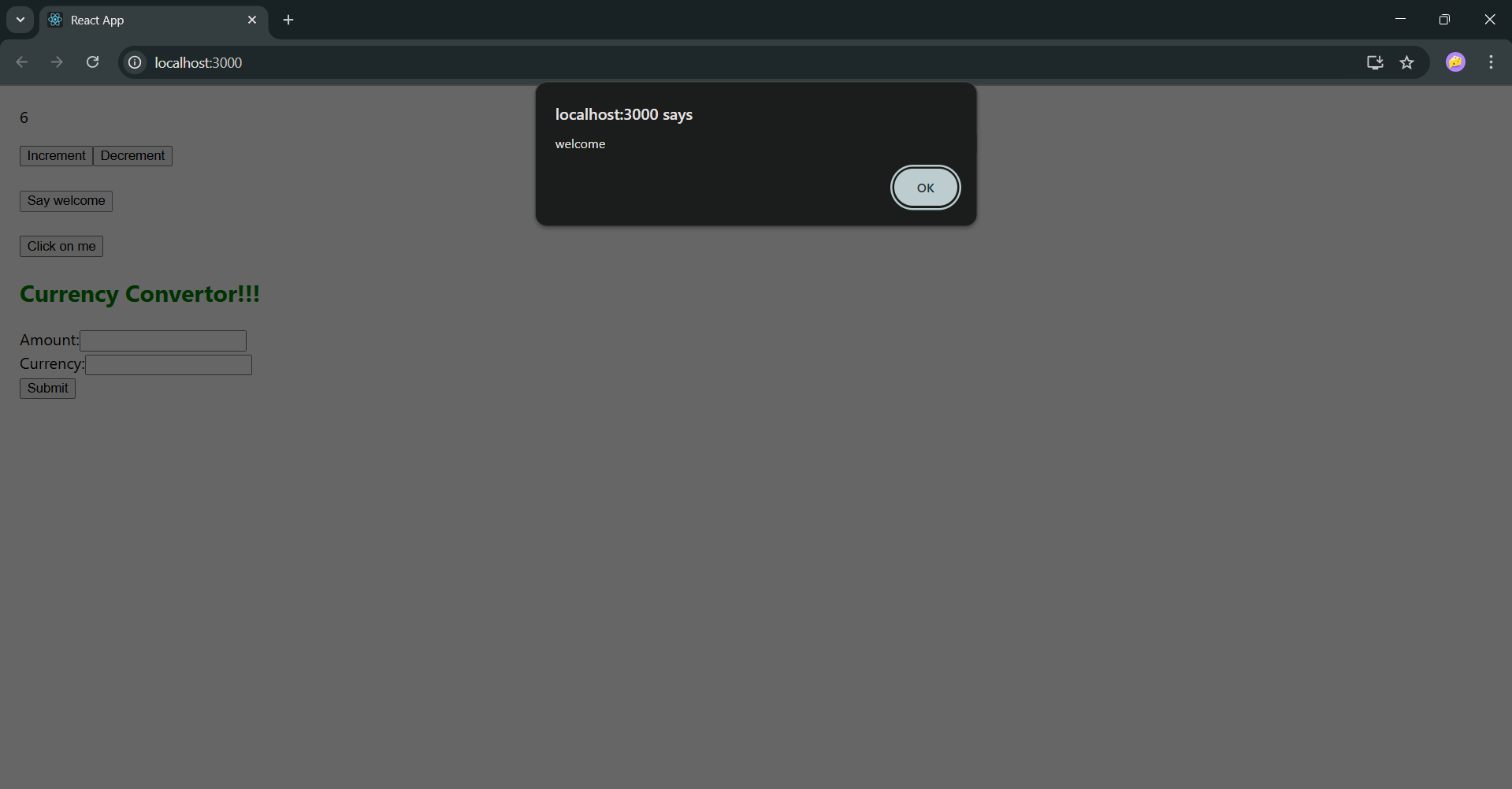
}

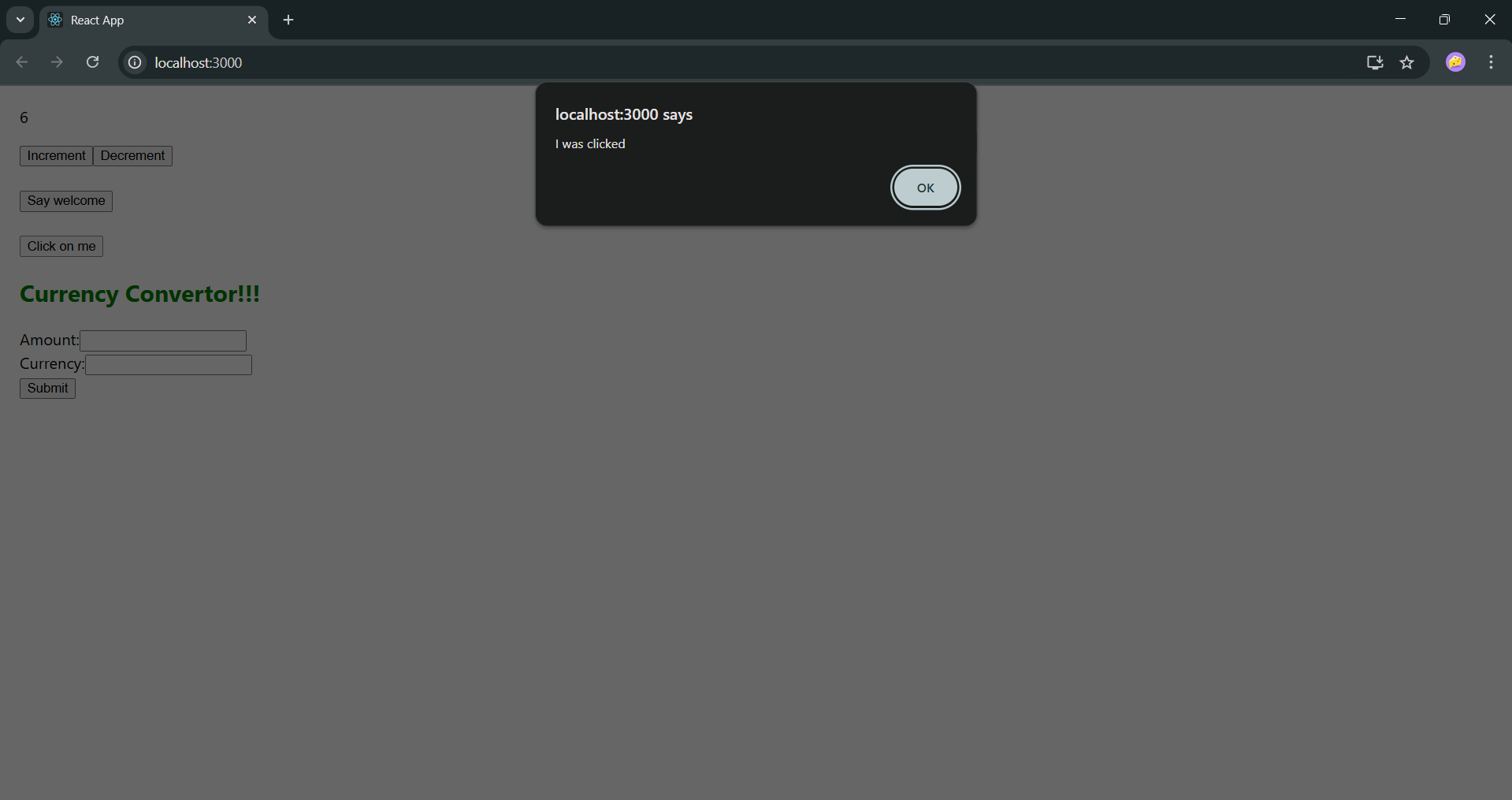
export default CurrencyConverter;

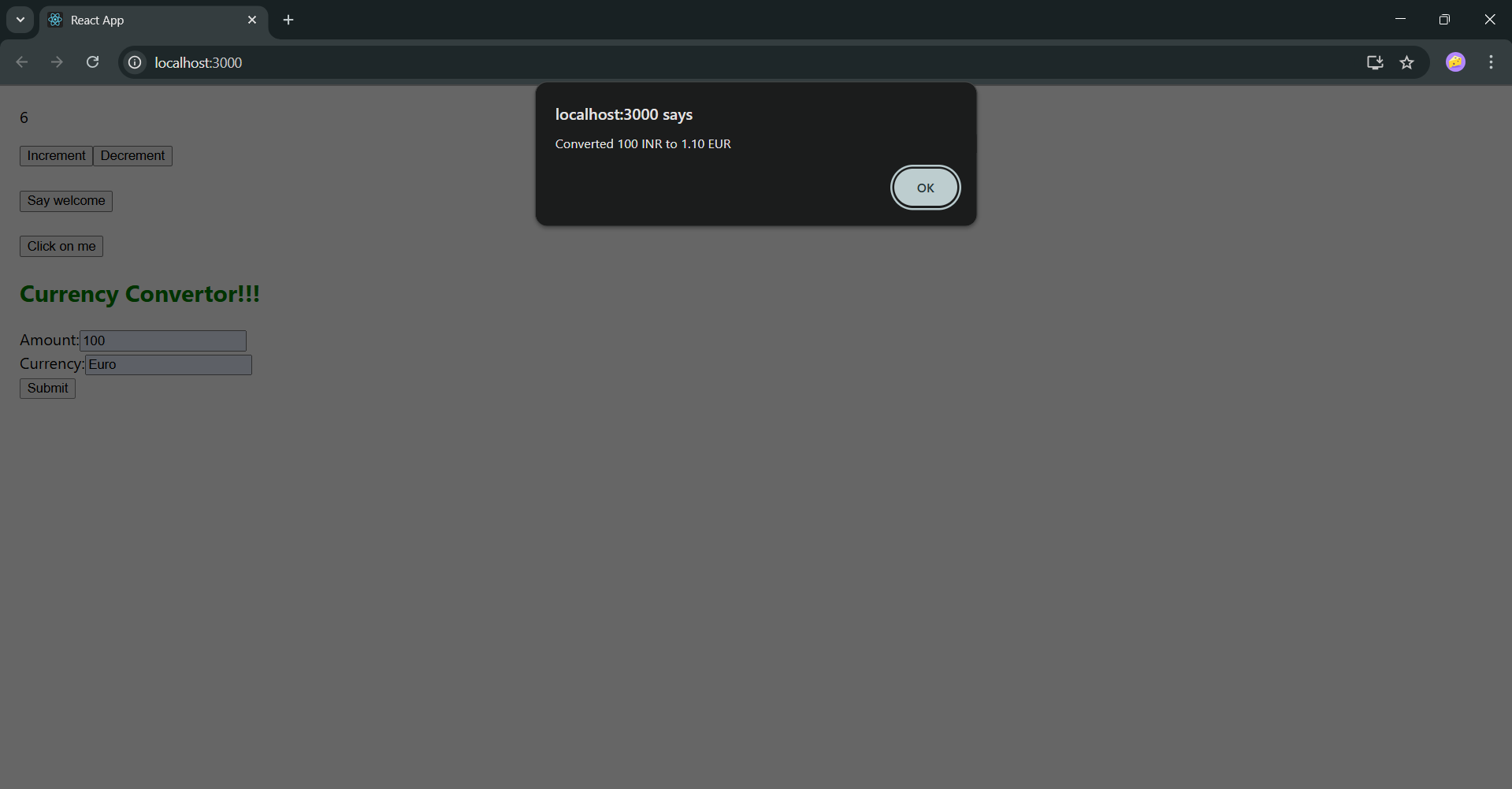
**Output:**

****

****

****

****

****

**12. ReactJS-HOL:**

1. Created a React app using npx create-react-app ticketbookingapp and opened it in VS Code.
2. Made LoginButton and LogoutButton components that handle click events using props.
3. Created a Greeting component that shows different messages based on login status.
4. Used class-based App component to manage the isLoggedIn state.
5. Wrote login/logout handlers to toggle the login state.
6. Used conditional rendering to switch between Login and Logout buttons.
7. Displayed Greeting + Button in the UI based on current state.
8. Ran and tested the app using npm start.

**//App.js**

import React, { Component } 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) {

  const isLoggedIn = props.isLoggedIn;

  if (isLoggedIn) {

    return <h2>Welcome back</h2>;

  }

  return <h2>Please sign up.</h2>;

}

class App extends Component {

  constructor(props) {

    super(props);

    this.state = { isLoggedIn: false };

  }

  handleLoginClick = () => {

    this.setState({ isLoggedIn: true });

  }

  handleLogoutClick = () => {

    this.setState({ isLoggedIn: false });

  }

  render() {

    const isLoggedIn = this.state.isLoggedIn;

    let button;

    if (isLoggedIn) {

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

    } else {

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

    }

    return (

      <div style={{ textAlign: 'center', marginTop: '100px' }}>

        <Greeting isLoggedIn={isLoggedIn} />

        {button}

      </div>

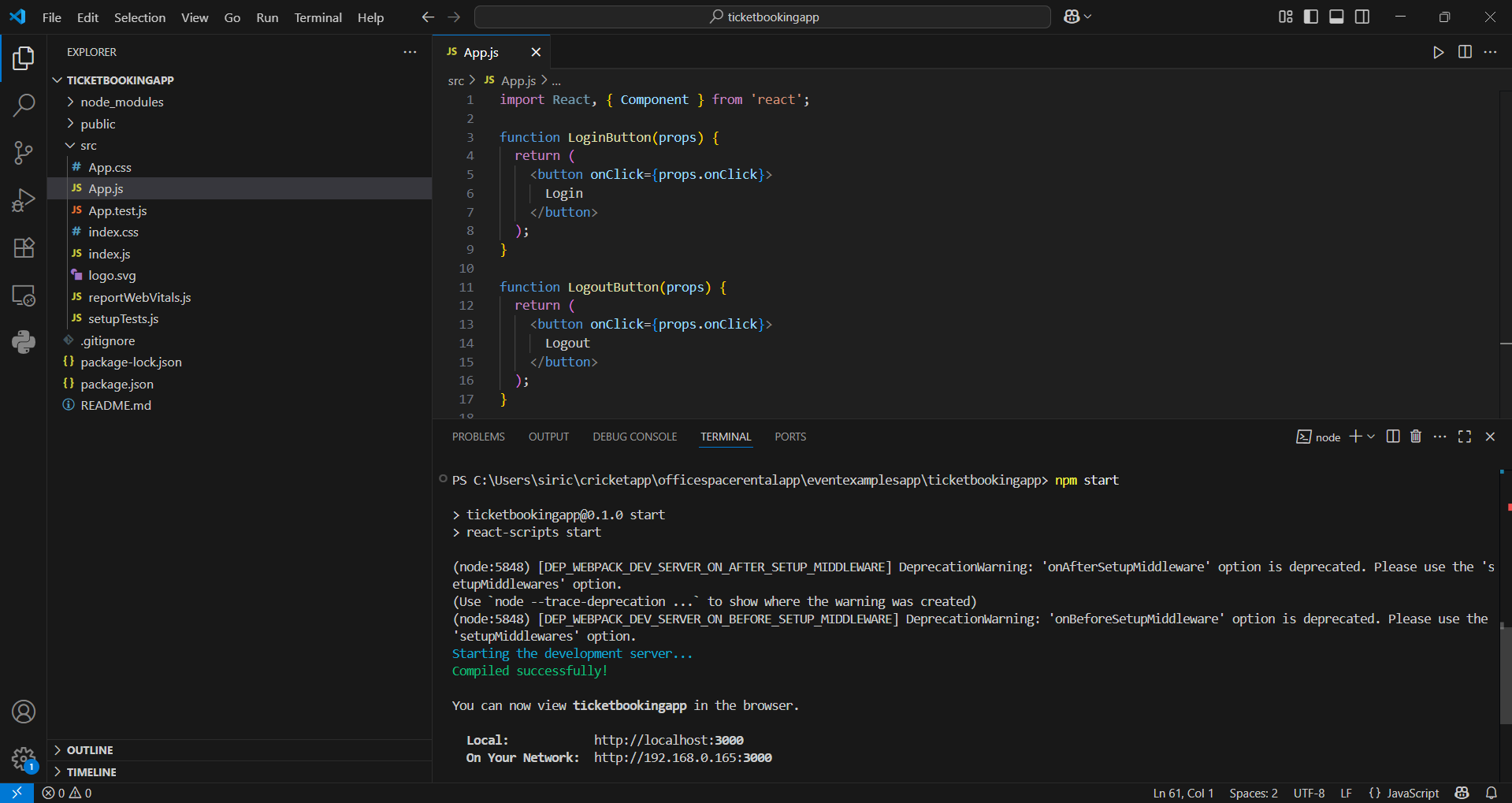
    );

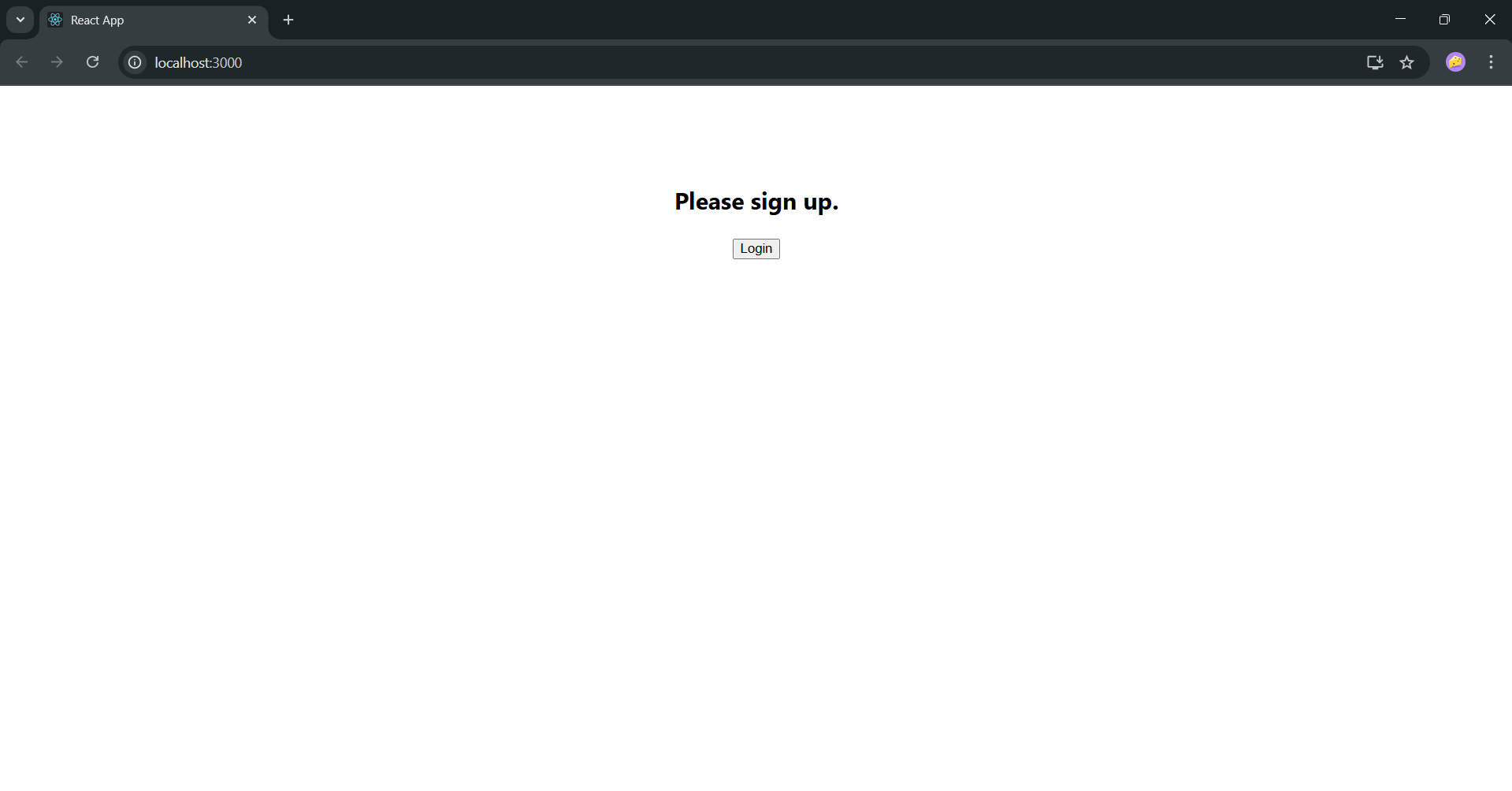
  }

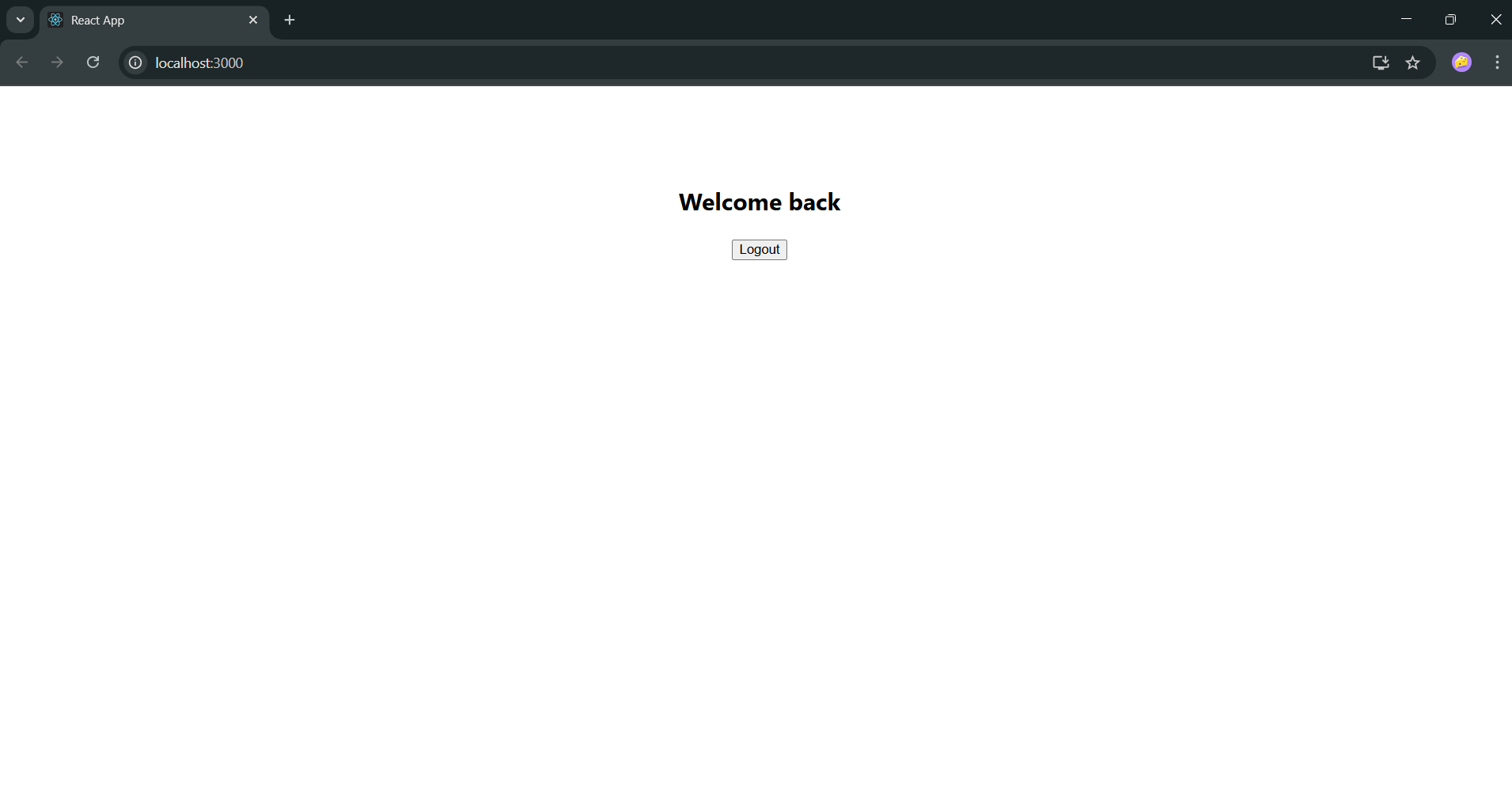
}

export default App;

**Output:**

****

****

****

**13. ReactJS-HOL:**

**Solution:**

**Steps:**

1. Created a React App using

* npx create-react-app bloggerapp

1. Used only App.js for everything to keep it simple.
2. Added data arrays for books, courses, and blogs inside App.js.
3. Rendered lists using .map() and used key for each item.
4. Used conditional rendering (&& and ternary operator) to display sections based on flags.
5. Styled layout with Flexbox in App.css to show 3 columns side by side.
6. Ran the app with npm start and verified the output.

**//App.js**

import React from 'react';

import './App.css';

function App() {

  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, cname: 'Angular', date: '4/5/2021' },

    { id: 2, cname: 'React', date: '6/3/2020' },

  ];

  const blogs = [

    { id: 1, title: 'React Learning', author: 'Stephen Biz', description: 'Welcome to learning React!' },

    { id: 2, title: 'Installation', author: 'Schewzdneier', description: 'You can install React from npm.' },

  ];

  const showBooks = true;

  const showCourses = true;

  const showBlogs = true;

  const BookDetails = (

    <ul>

      {books.map((book) => (

        <div key={book.id}>

          <h3>{book.bname}</h3>

          <h4>{book.price}</h4>

        </div>

      ))}

    </ul>

  );

  const CourseDetails = (

    <ul>

      {courses.map((course) => (

        <div key={course.id}>

          <h3>{course.cname}</h3>

          <h4>{course.date}</h4>

        </div>

      ))}

    </ul>

  );

  const BlogDetails = (

    <ul>

      {blogs.map((blog) => (

        <div key={blog.id}>

          <h3>{blog.title}</h3>

          <h5>{blog.author}</h5>

          <p>{blog.description}</p>

        </div>

      ))}

    </ul>

  );

  return (

    <div className="container">

      {showCourses && (

        <div className="section">

          <h1>Course Details</h1>

          {CourseDetails}

        </div>

      )}

      {showBooks && (

        <div className="section">

          <h1>Book Details</h1>

          {BookDetails}

        </div>

      )}

      {showBlogs ? (

        <div className="section">

          <h1>Blog Details</h1>

          {BlogDetails}

        </div>

      ) : null}

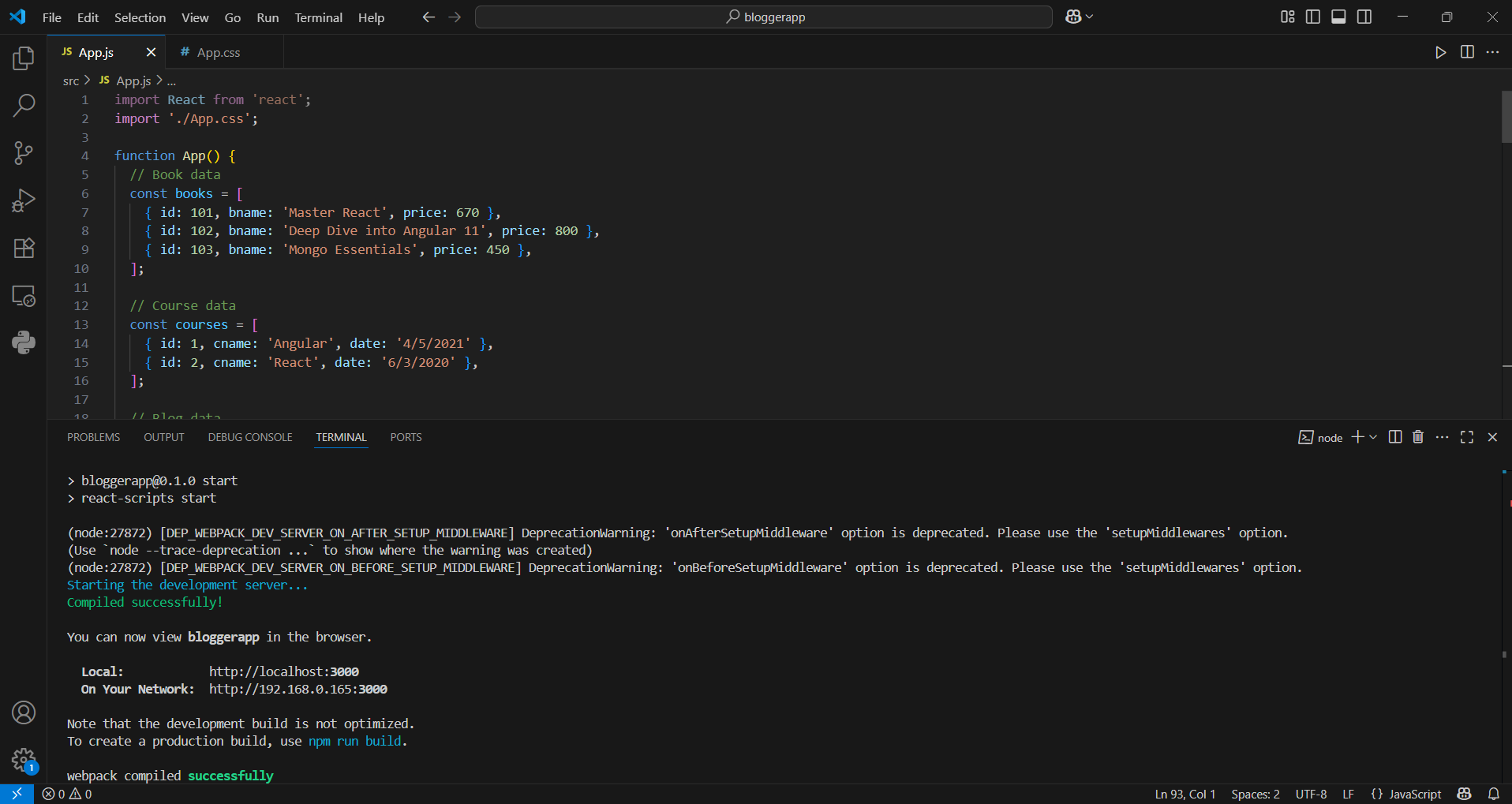
    </div>

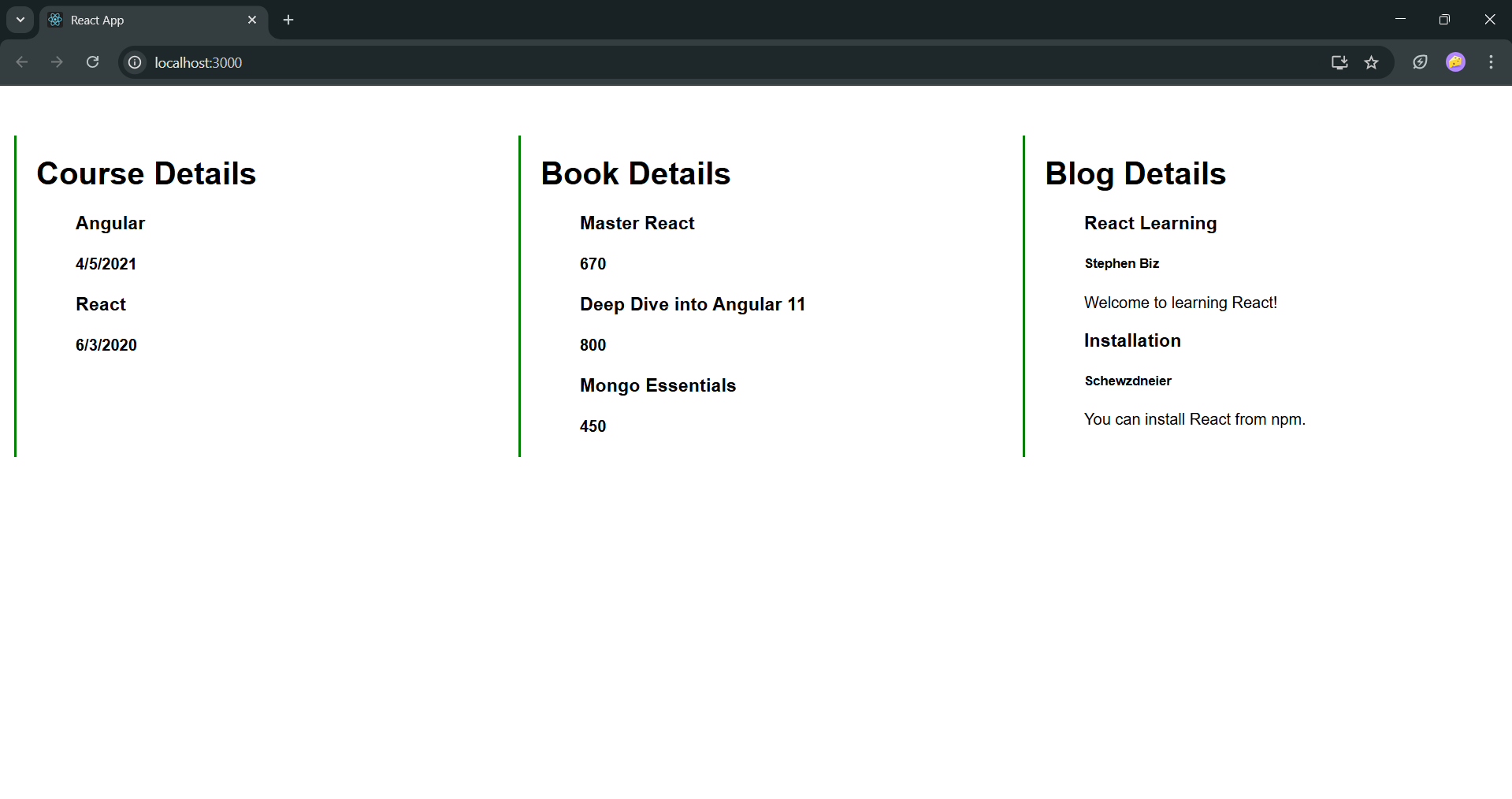
  );

}

export default App;

**Output:**

****

****