**9.Objectives**

* List the features of ES6
* Explain JavaScript let
* Identify the differences between var and let
* Explain JavaScript const
* Explain ES6 class fundamentals
* Explain ES6 class inheritance
* Define ES6 arrow functions
* Identify set(), map()

In this hands-on lab, you will learn how to:

* Use map() method of ES6
* Apply arrow functions of ES6
* Implement Destructuring features of ES6

**Prerequisites**

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

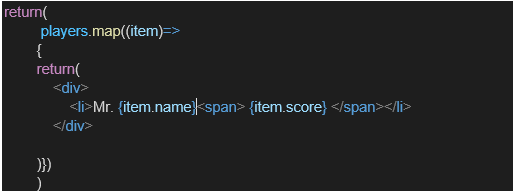
**Notes**

Estimated time to complete this lab: **60 minutes.**

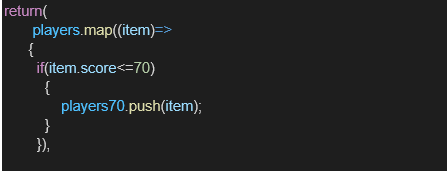
Create a React Application named “cricketapp” with the following components:

1. ListofPlayers

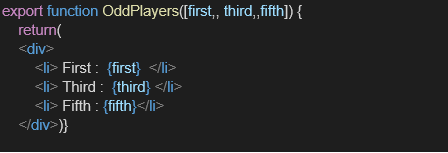
* Declare an array with 11 players and store details of their names and scores using the map feature of ES6



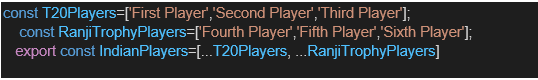
* Filter the players with scores below 70 using arrow functions of ES6.



1. IndianPlayers
   1. Display the Odd Team Player and Even Team players using the Destructuring features of ES6



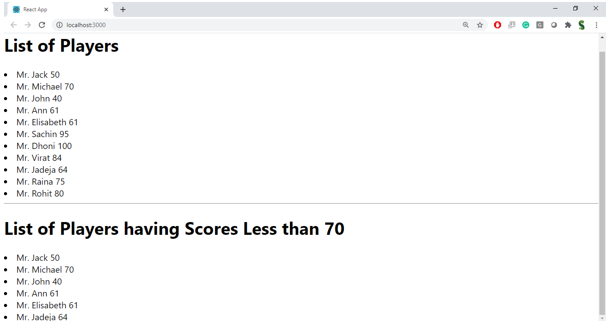
* 1. Declare two arrays T20players and RanjiTrophy players and merge the two arrays and display them using the Merge feature of ES6



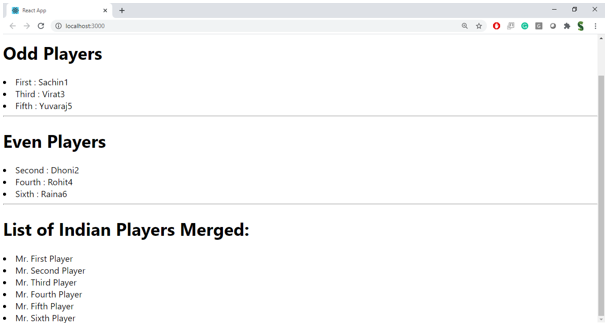
Display these two components in the same home page using a simple if else in the flag variable.

**Output:**

When Flag=true



When Flag=false



**Hint:**

**1.Create the React App:**

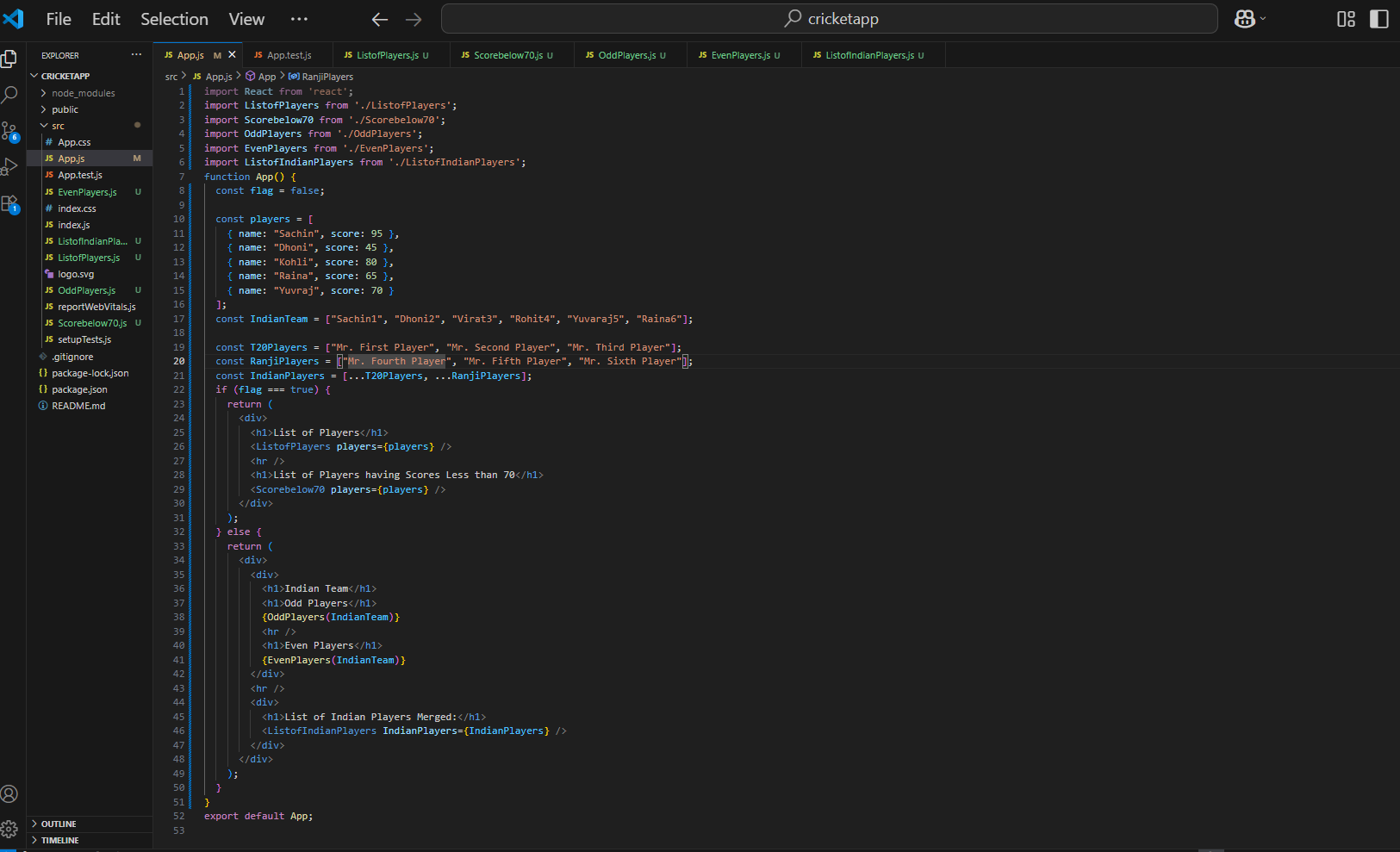
npx create-react-app cricketapp

cd cricketapp

npm start

Code

**2.App.js**

****

This App.js file is the main component that conditionally renders player-related components based on a flag variable.  
It demonstrates ES6 features like destructuring, arrow functions, map(), filter(), and spread syntax in a React application.

import React from 'react';

import ListofPlayers from './ListofPlayers';

import Scorebelow70 from './Scorebelow70';

import OddPlayers from './OddPlayers';

import EvenPlayers from './EvenPlayers';

import ListofIndianPlayers from './ListofIndianPlayers';

function App() {

  const flag = false;

  const players = [

    { name: "Sachin", score: 95 },

    { name: "Dhoni", score: 45 },

    { name: "Kohli", score: 80 },

    { name: "Raina", score: 65 },

    { name: "Yuvraj", score: 70 }

  ];

  const IndianTeam = ["Sachin1", "Dhoni2", "Virat3", "Rohit4", "Yuvaraj5", "Raina6"];

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

  const RanjiPlayers = ["Mr. Fourth Player", "Mr. Fifth Player", "Mr. Sixth Player"];

  const IndianPlayers = [...T20Players, ...RanjiPlayers];

  if (flag === true) {

    return (

      <div>

        <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>

        <div>

          <h1>Indian Team</h1>

          <h1>Odd Players</h1>

          {OddPlayers(IndianTeam)}

          <hr />

          <h1>Even Players</h1>

          {EvenPlayers(IndianTeam)}

        </div>

        <hr />

        <div>

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

          <ListofIndianPlayers IndianPlayers={IndianPlayers} />

        </div>

      </div>

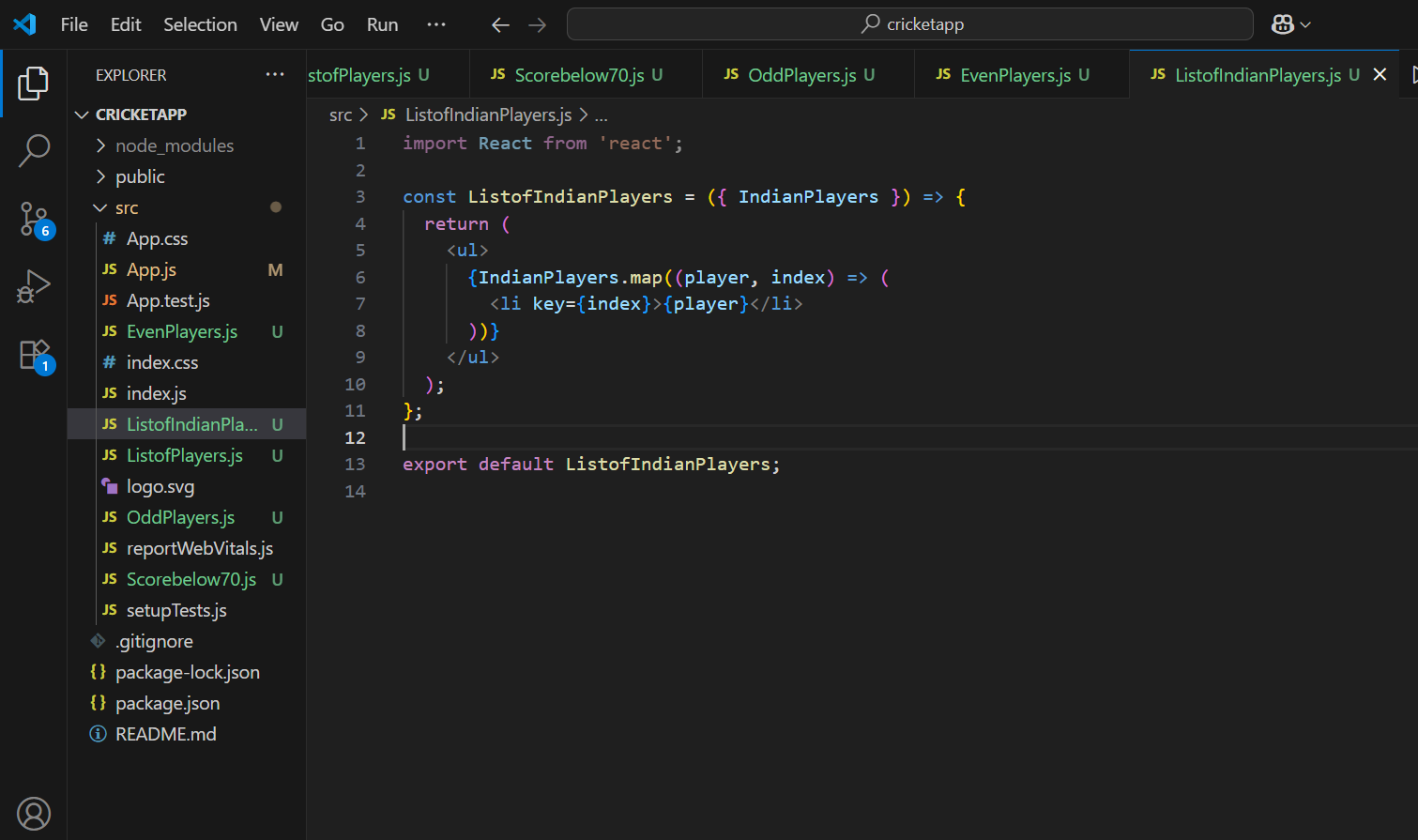
    );

  }

}

export default App;

**3.ListofIndianPlayers.js**

****

import React from 'react';

const ListofIndianPlayers = ({ IndianPlayers }) => {

  return (

    <ul>

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

        <li key={index}>{player}</li>

      ))}

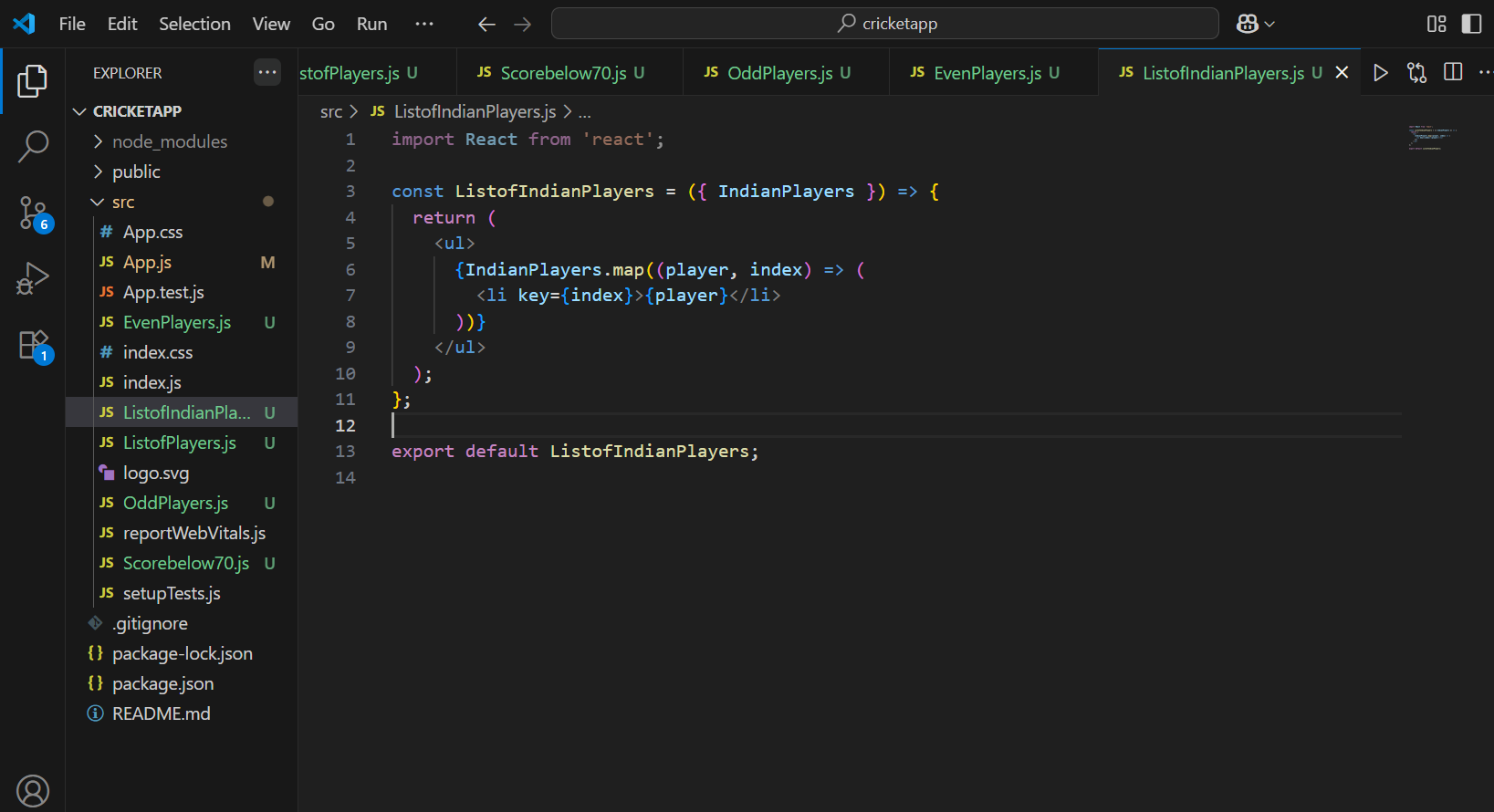
    </ul>

  );

};

export default ListofIndianPlayers;

**4.ListofPlayers.js**

****

import React from 'react';

const ListofPlayers = ({ players }) => {

  return (

    <ul>

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

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

      ))}

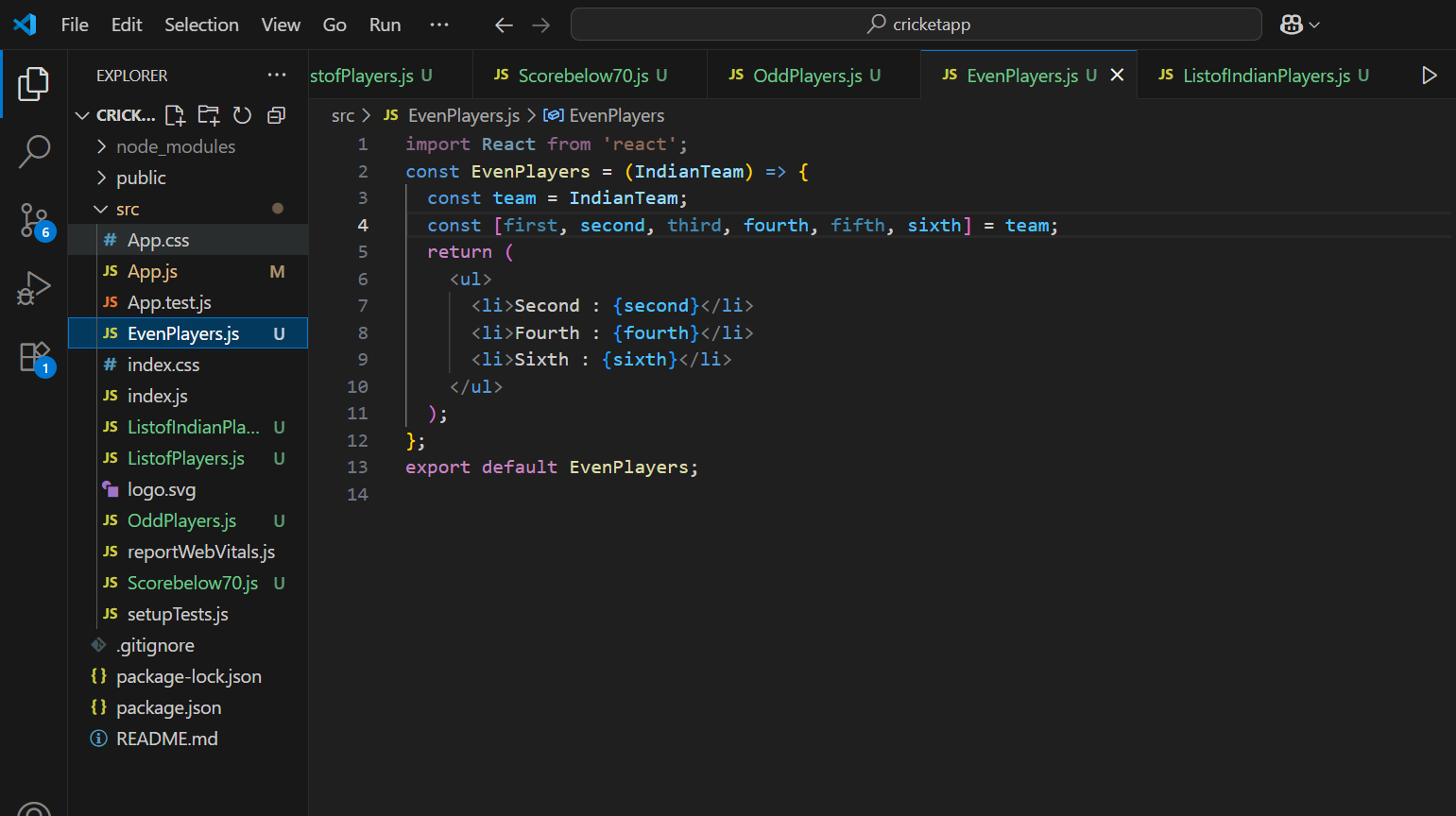
    </ul>

  );

};

export default ListofPlayers;

**5.EvenPlayers.js**

****

import React from 'react';

const EvenPlayers = (IndianTeam) => {

  const team = IndianTeam;

  const [first, second, third, fourth, fifth, sixth] = team;

  return (

    <ul>

      <li>Second : {second}</li>

      <li>Fourth : {fourth}</li>

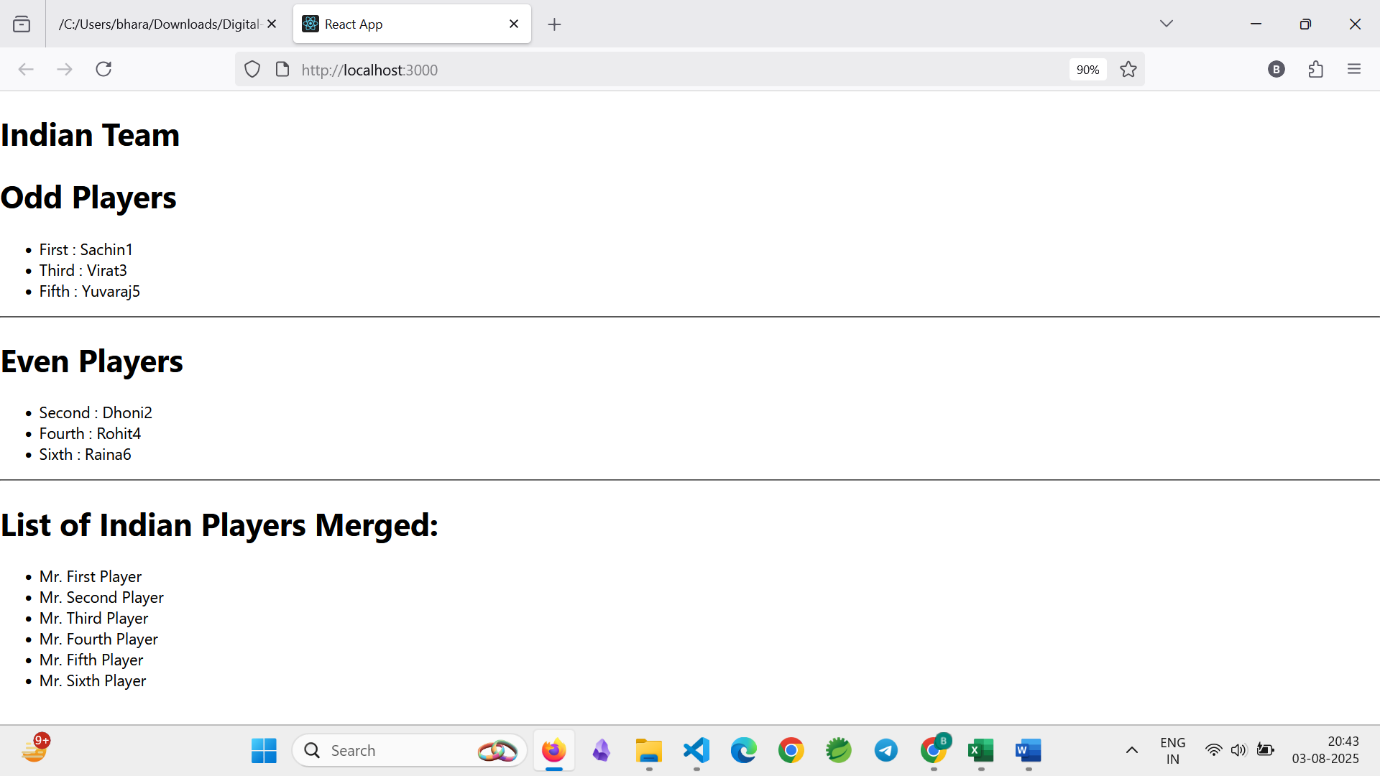
      <li>Sixth : {sixth}</li>

    </ul>

  );

};

export default EvenPlayers;

**OUTPUT:**

**--------------------------------------------------------------------------------------------------------------------------** Objectives

* Define JSX
* Explain about ECMA Script
* Explain React.createElement()
* Explain how to create React nodes with JSX
* Define how to render JSX to DOM
* Explain how to use JavaScript expressions in JSX
* Explain how to use inline CSS in JSX

In this hands-on lab, you will learn how to:

* Use JSX syntax in React applications
* Use inline CSS in JSX

Prerequisites

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

Notes

Estimated time to complete this lab: 60 minutes.

Create a React Application named “officespacerentalapp” which uses React JSX to createelements, 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

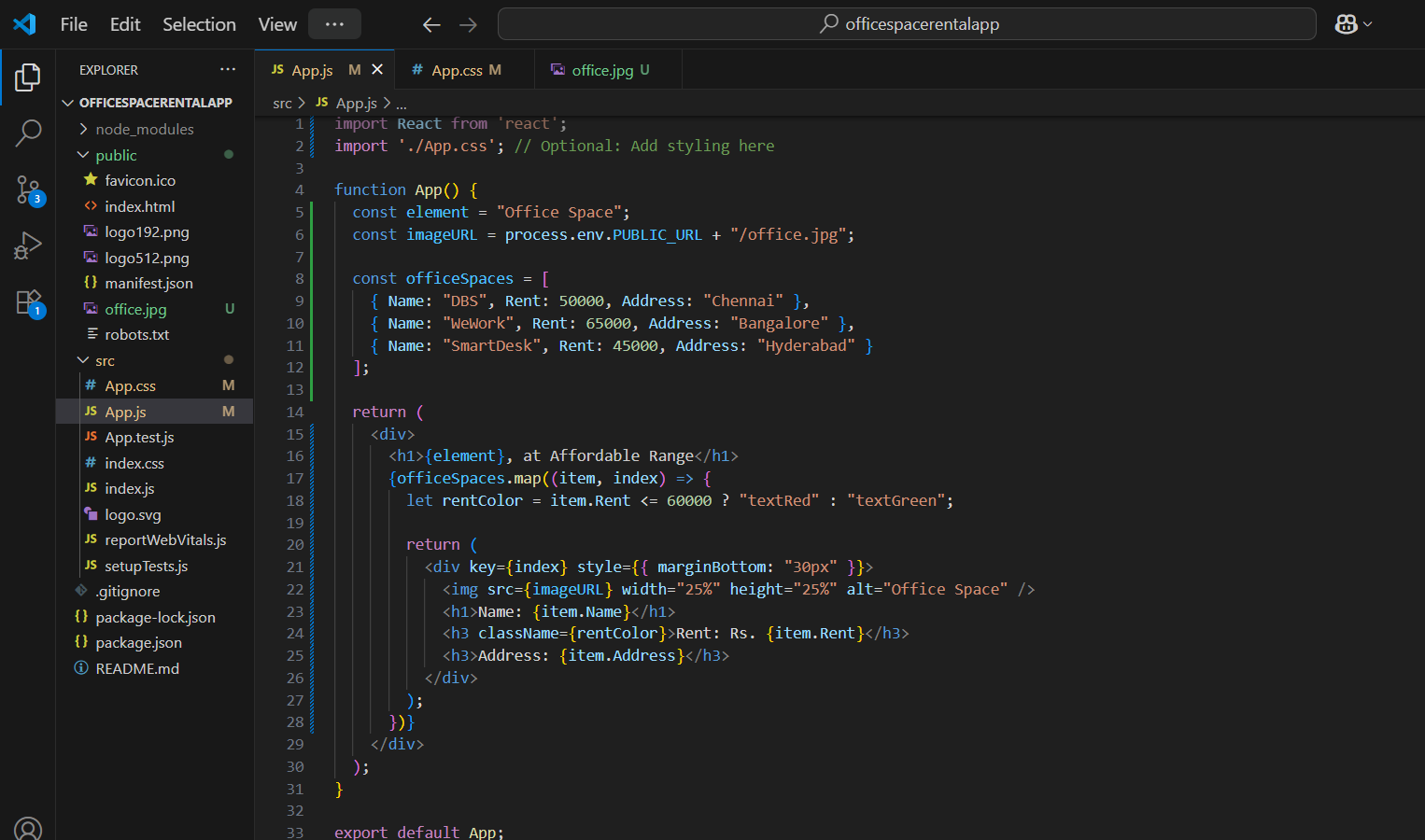
Create an object of office to display the details like Name, Rent and Address.

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.

**Code:**

**App.js**

****

import React from 'react';

import './App.css'; // Optional: Add styling here

function App() {

const element = "Office Space";

const imageURL = process.env.PUBLIC\_URL + "/office.jpg";

const officeSpaces = [

{ Name: "DBS", Rent: 50000, Address: "Chennai" },

{ Name: "WeWork", Rent: 65000, Address: "Bangalore" },

{ Name: "SmartDesk", Rent: 45000, Address: "Hyderabad" }

];

return (

<div>

<h1>{element}, at Affordable Range</h1>

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

let rentColor = item.Rent <= 60000 ? "textRed" : "textGreen";

return (

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

<img src={imageURL} width="25%" height="25%" alt="Office Space" />

<h1>Name: {item.Name}</h1>

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

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

</div>

);

})}

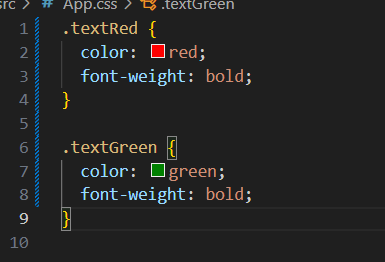
</div>

);

}

export default App;

**App.css**

****

.textRed {

  color: red;

  font-weight: bold;

}

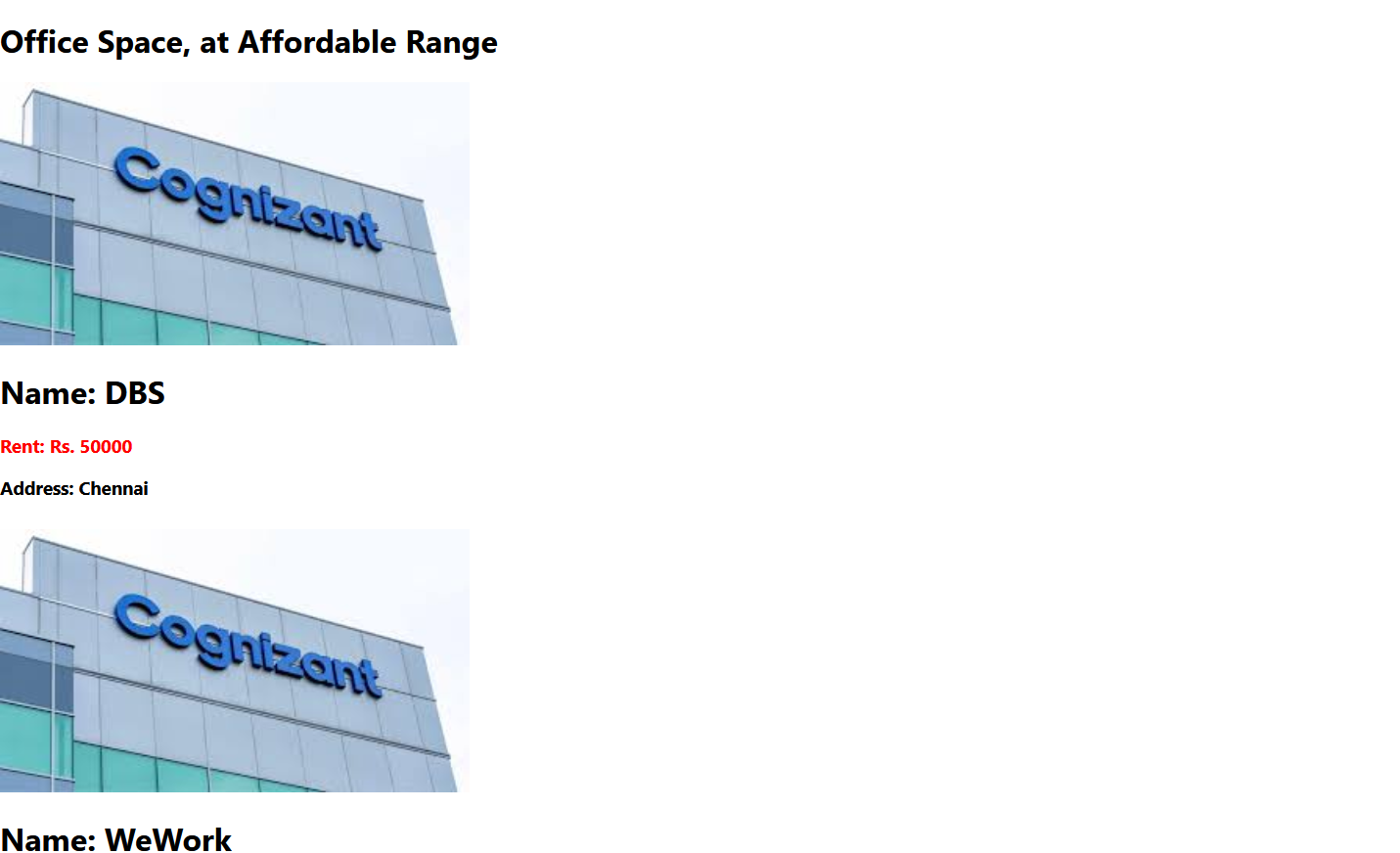
.textGreen {

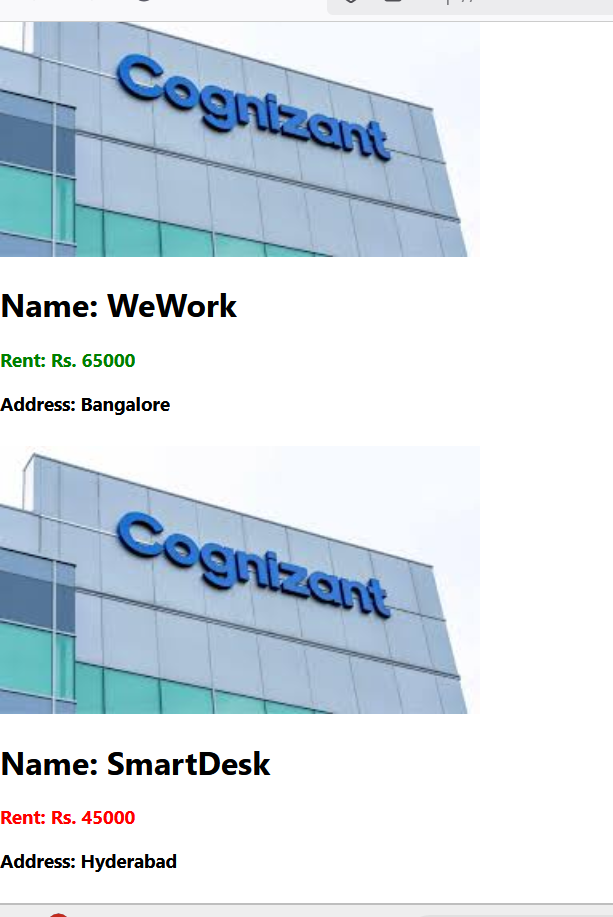
  color: green;

  font-weight: bold;

}

**Output:**

**OUTPUT:**



**Objectives**

* Explain React events
* Explain about event handlers
* Define Synthetic event
* Identify React event naming convention

In this hands-on lab, you will learn how to:

* Implement Event handling concept in React applications
* Use this keyword
* Use synthetic event

**Prerequisites**

The following is required to complete this hands-on lab:

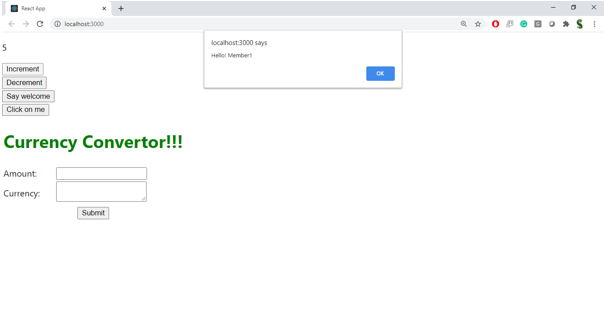
* Node.js
* NPM
* Visual Studio Code

**Notes**

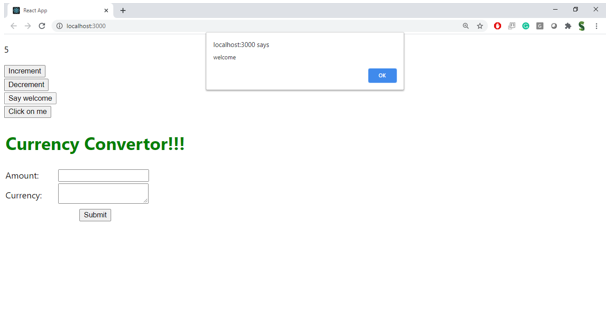
Estimated time to complete this lab: **90 minutes.**

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

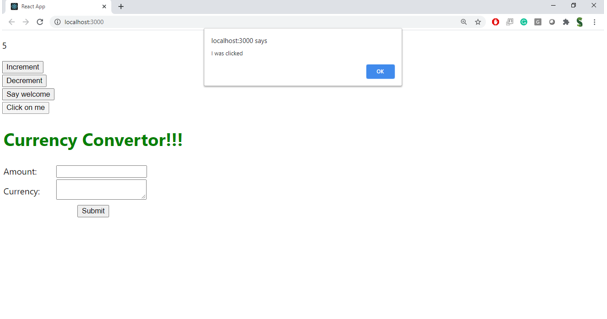
1. Create “Increment” button to increase the value of the counter and “Decrement” button to decrease the value of the counter. The “Increase” button should invoke multiple methods.
   1. To increment the value
   2. Say Hello followed by a static message.



1. Create a button “Say Welcome” which invokes the function which takes “welcome” as an argument.

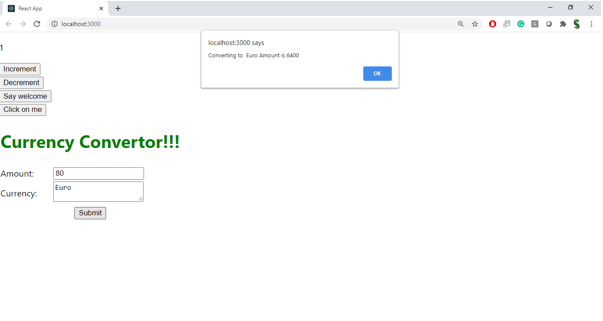


1. Create a button which invokes synthetic event “OnPress” which display “I was clicked”



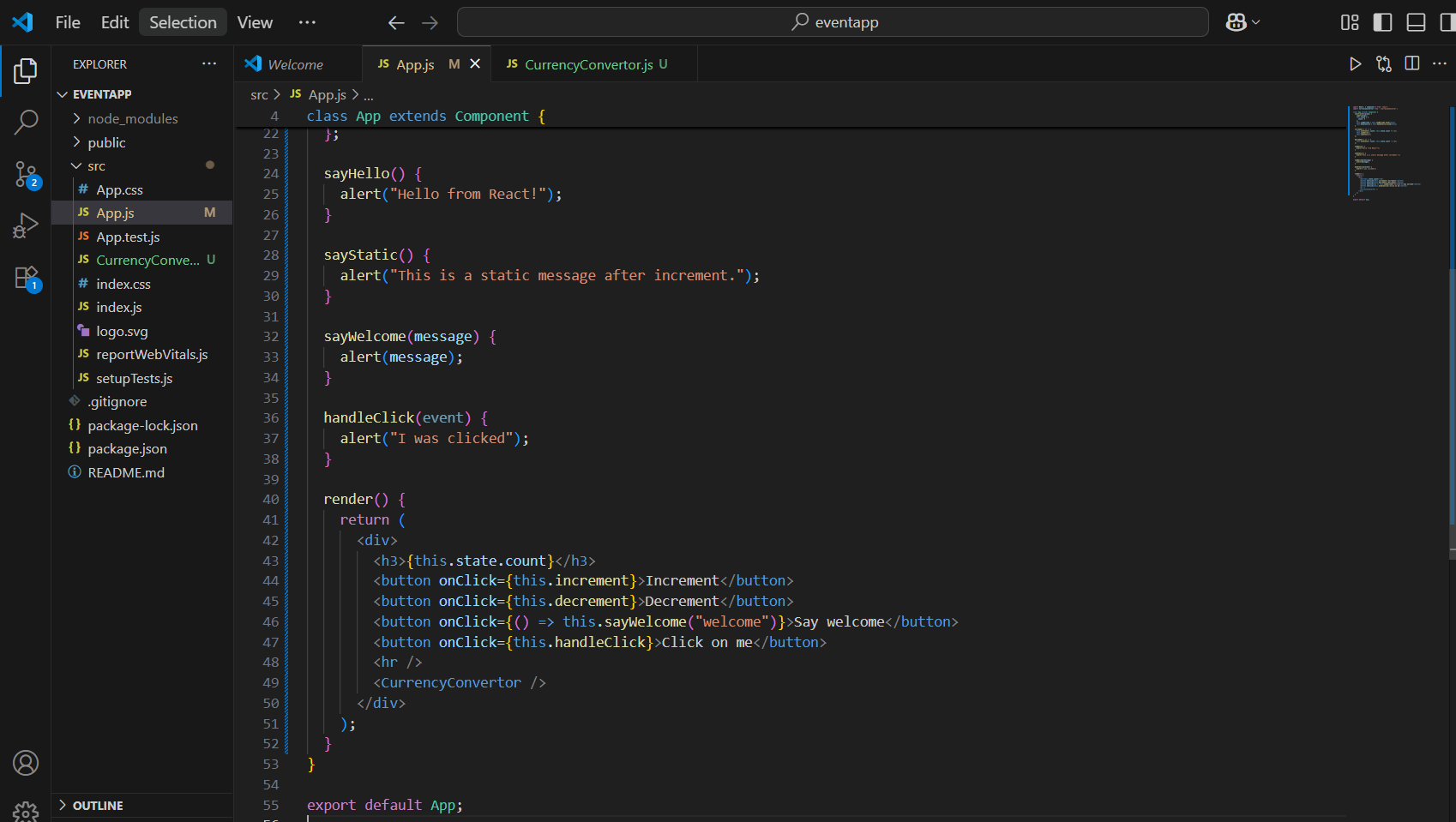
Create a “CurrencyConvertor” component which will convert the Indian Rupees to Euro when the Convert button is clicked.

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



**Code:**

**App.js:**

****

import React, { Component } from 'react';

import CurrencyConvertor from './CurrencyConvertor';

class App extends Component {

  constructor(props) {

    super(props);

    this.state = {

      count: 5

    };

    this.sayWelcome = this.sayWelcome.bind(this);

    this.handleClick = this.handleClick.bind(this);

  }

  increment = () => {

    this.setState({ count: this.state.count + 1 });

    this.sayHello();

    this.sayStatic();

  };

  decrement = () => {

    this.setState({ count: this.state.count - 1 });

  };

  sayHello() {

    alert("Hello from React!");

  }

  sayStatic() {

    alert("This is a static message after increment.");

  }

  sayWelcome(message) {

    alert(message);

  }

  handleClick(event) {

    alert("I was clicked");

  }

  render() {

    return (

      <div>

        <h3>{this.state.count}</h3>

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

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

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

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

        <hr />

        <CurrencyConvertor />

      </div>

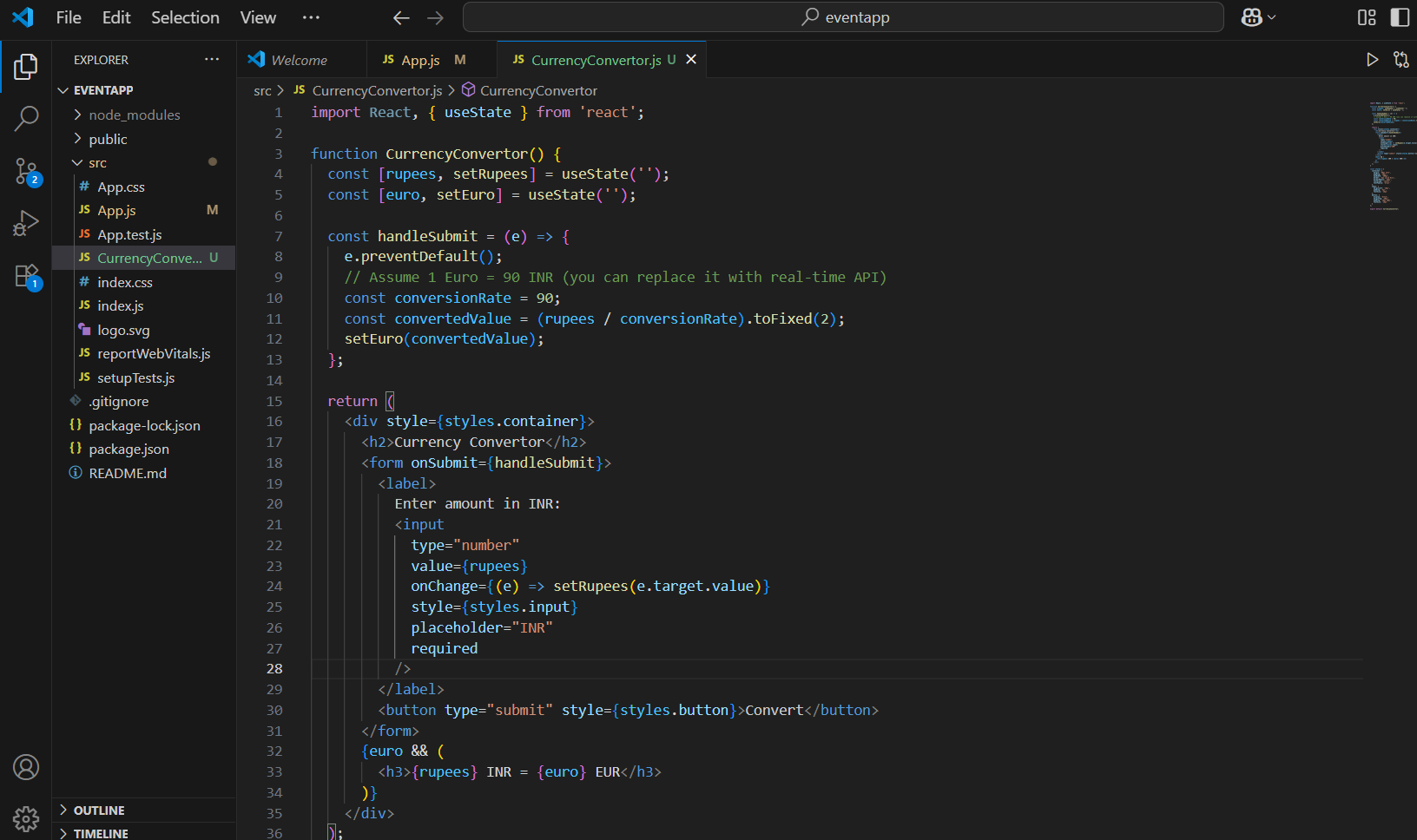
    );

  }

}

export default App;

**CurrencyConverter.js:**

****

import React, { useState } from 'react';

function CurrencyConvertor() {

  const [rupees, setRupees] = useState('');

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

  const handleSubmit = (e) => {

    e.preventDefault();

    // Assume 1 Euro = 90 INR (you can replace it with real-time API)

    const conversionRate = 90;

    const convertedValue = (rupees / conversionRate).toFixed(2);

    setEuro(convertedValue);

  };

  return (

    <div style={styles.container}>

      <h2>Currency Convertor</h2>

      <form onSubmit={handleSubmit}>

        <label>

          Enter amount in INR:

          <input

            type="number"

            value={rupees}

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

            style={styles.input}

            placeholder="INR"

            required

          />

        </label>

        <button type="submit" style={styles.button}>Convert</button>

      </form>

      {euro && (

        <h3>{rupees} INR = {euro} EUR</h3>

      )}

    </div>

  );

}

const styles = {

  container: {

    margin: '50px auto',

    padding: '30px',

    maxWidth: '400px',

    border: '2px solid #ccc',

    borderRadius: '10px',

    textAlign: 'center',

    fontFamily: 'Arial'

  },

  input: {

    marginLeft: '10px',

    padding: '5px',

    fontSize: '16px'

  },

  button: {

    display: 'block',

    marginTop: '20px',

    padding: '10px 20px',

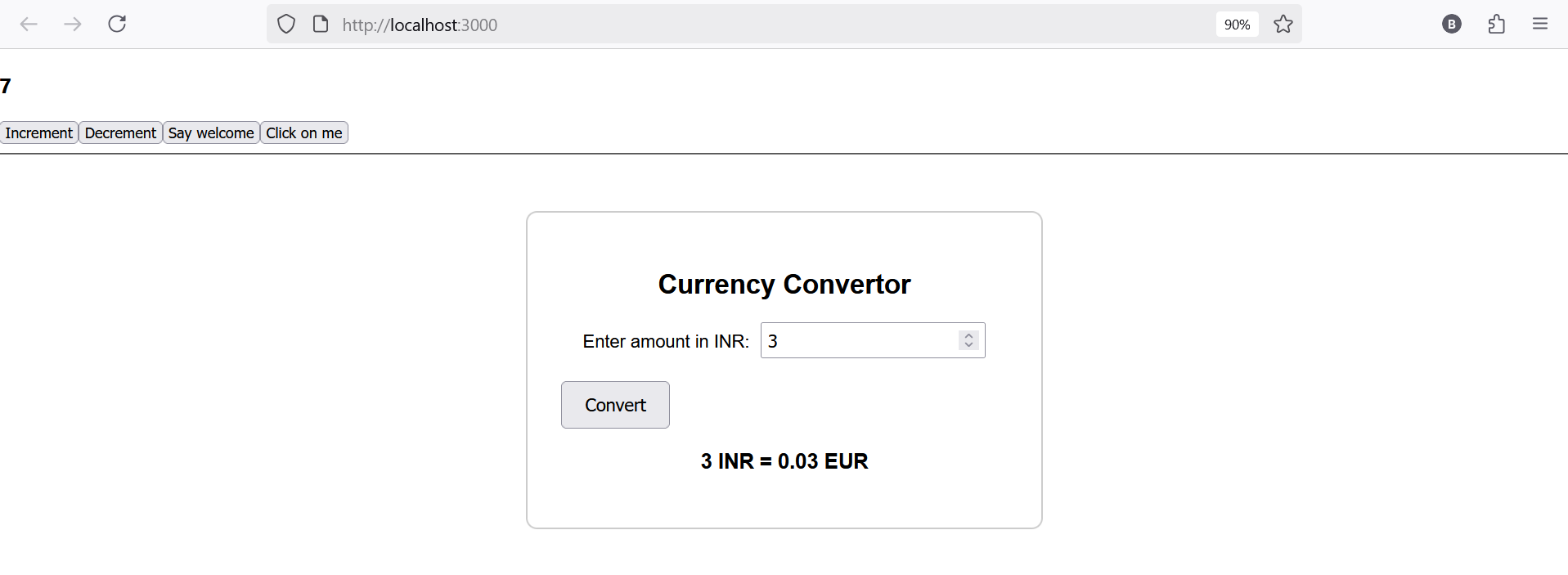
    fontSize: '16px'

  }

};

export default CurrencyConvertor;

**OUTPUT:**

****

**ReactJS HOL-12**

**Objectives**

* **Explain about conditional rendering in React**
* **Define element variables**
* **Explain how to prevent components from rendering**

**In this hands-on lab, you will learn how to:**

* **Implement conditional rendering in React applications**

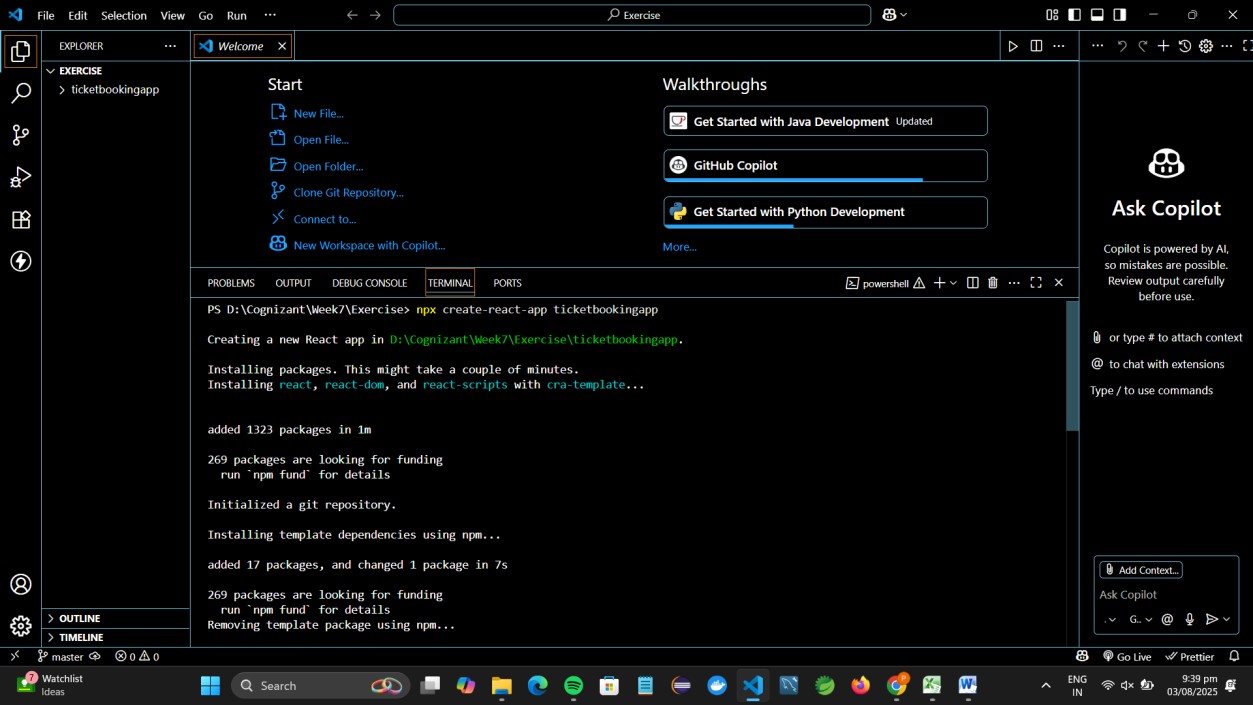
**Create a React Application named “ticketbookingapp” where the guest user can browse the page where the flight details are displayed whereas the logged in user only can book tickets.**

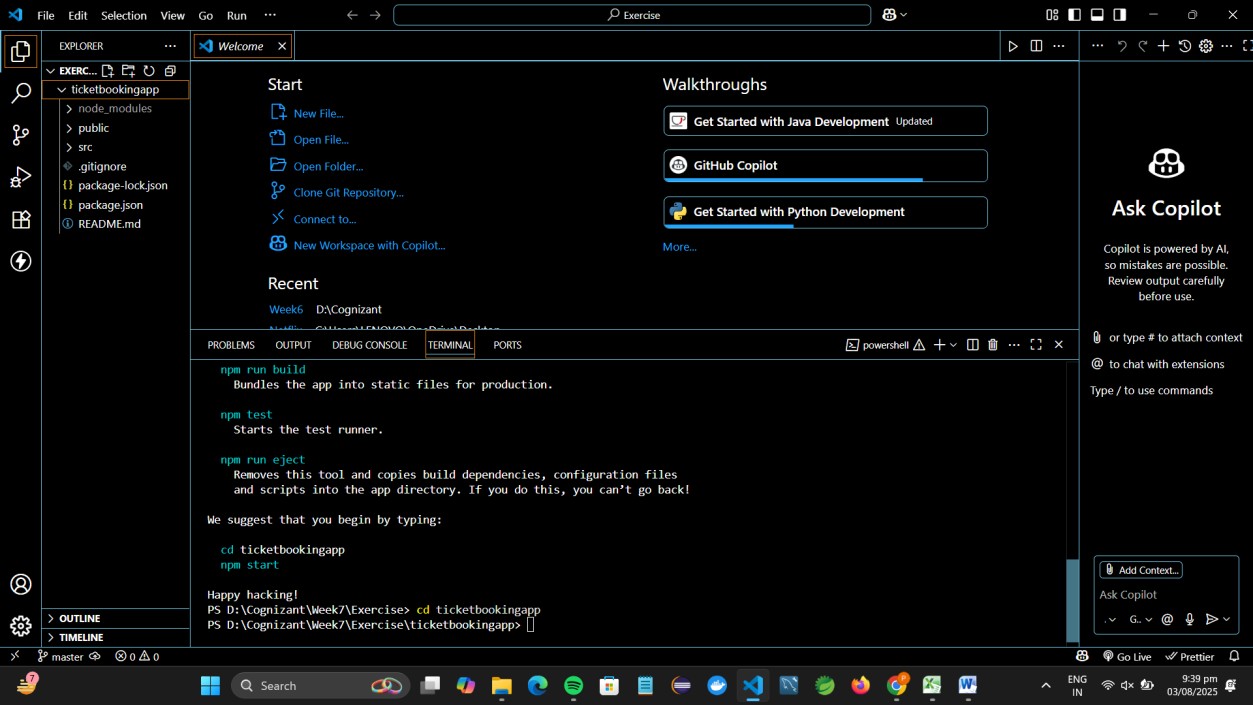
**The Login and Logout buttons should accordingly display different pages. Once the user is logged in the User page should be displayed. When the user clicks on Logout, the Guest page should be displayed.**

**Step 1: Create React App**

**Open your terminal inside VS Code, and run:**

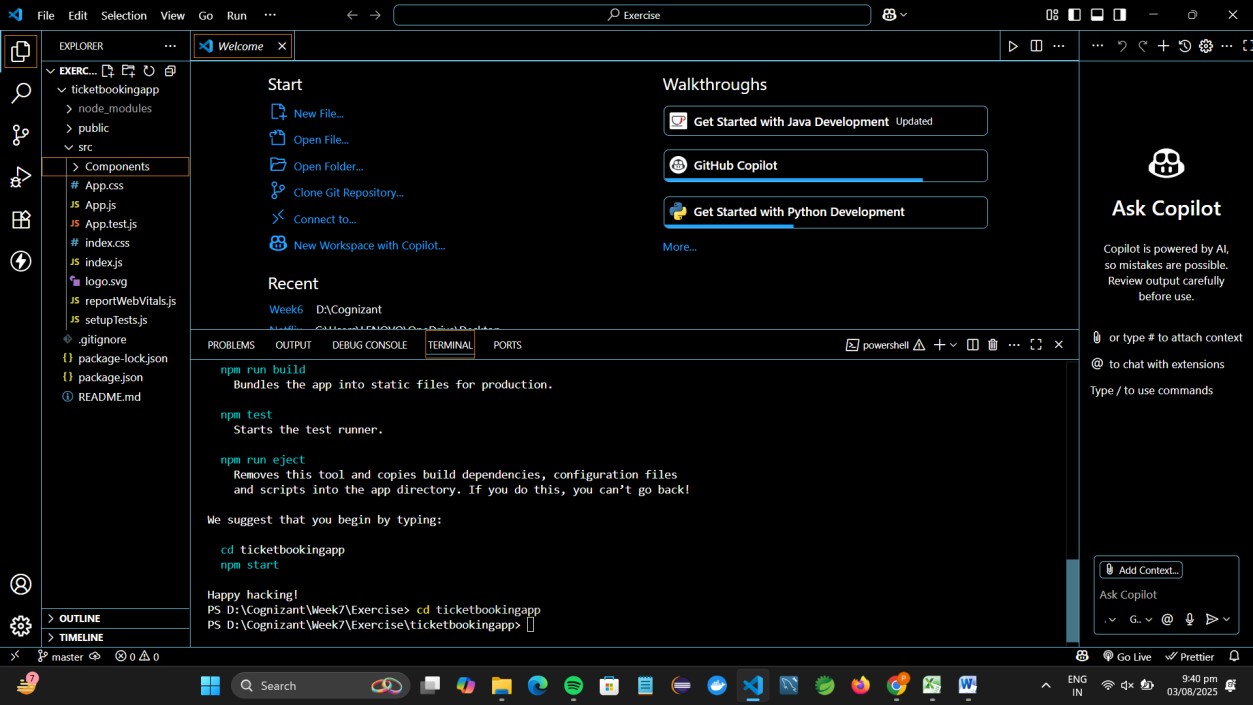
**npx create-react-app ticketbookingapp cd ticketbookingapp**

****

****

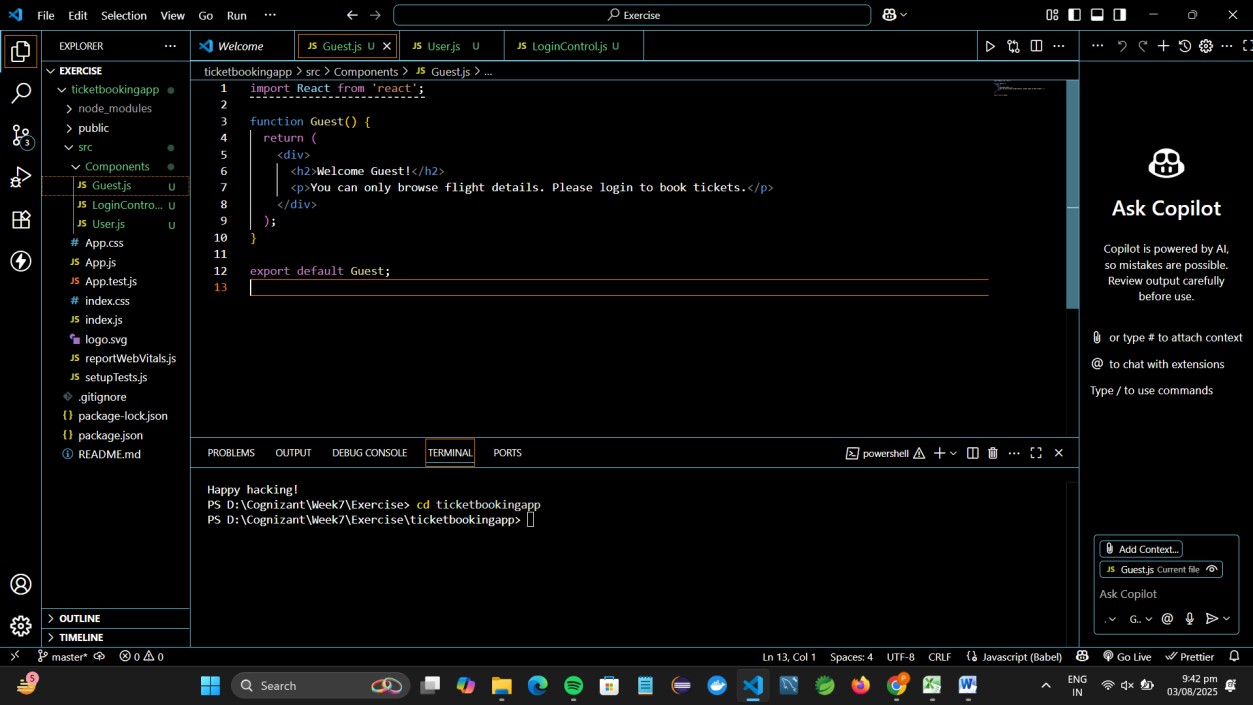
**Step 2: Create Components**

**Inside src, create a folder:**

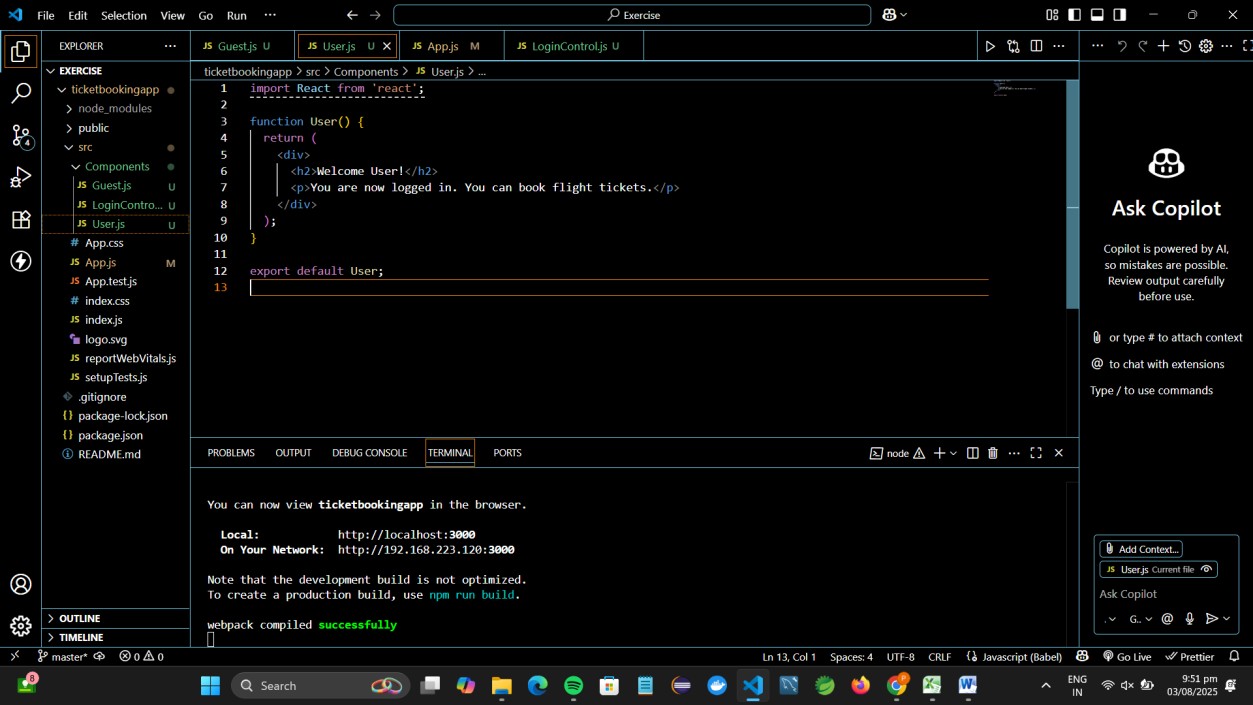
****

**Inside Components, create 3 files:**

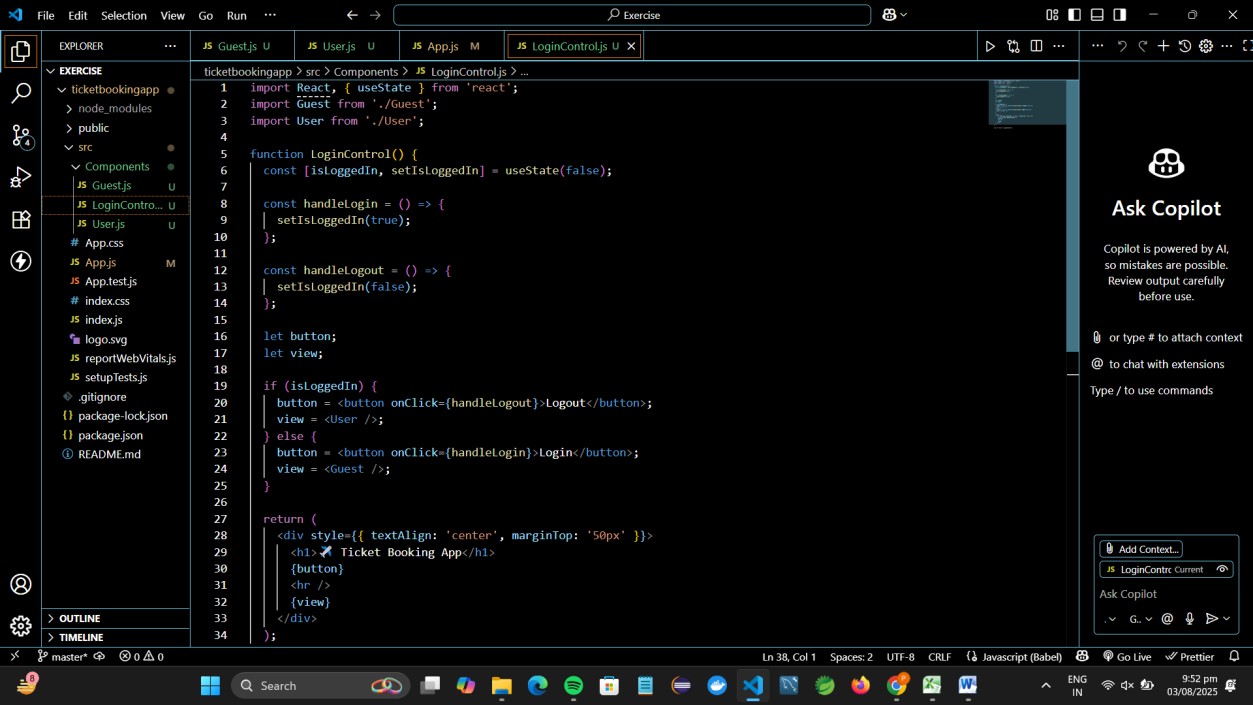
**Guest.js:**

****

**User.js:**

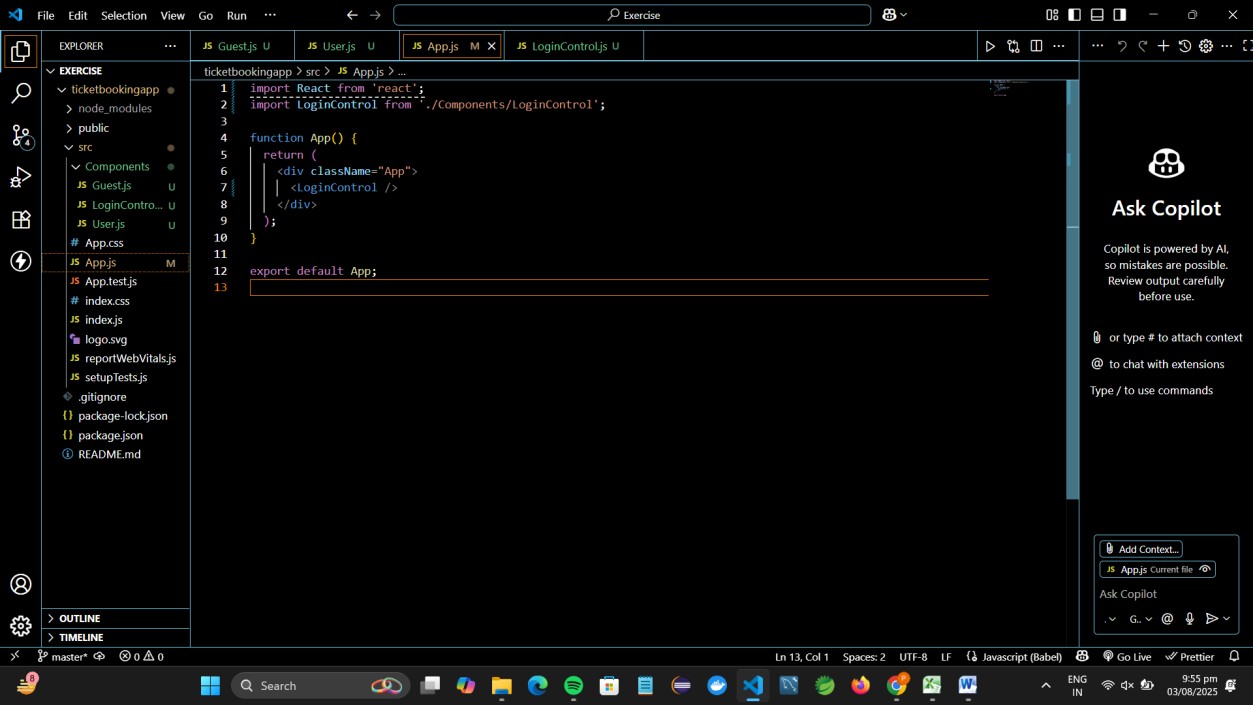
****

**LoginControl.js:**

****

**Step 3: Edit App.js**

**Replace content in src/App.js with:**

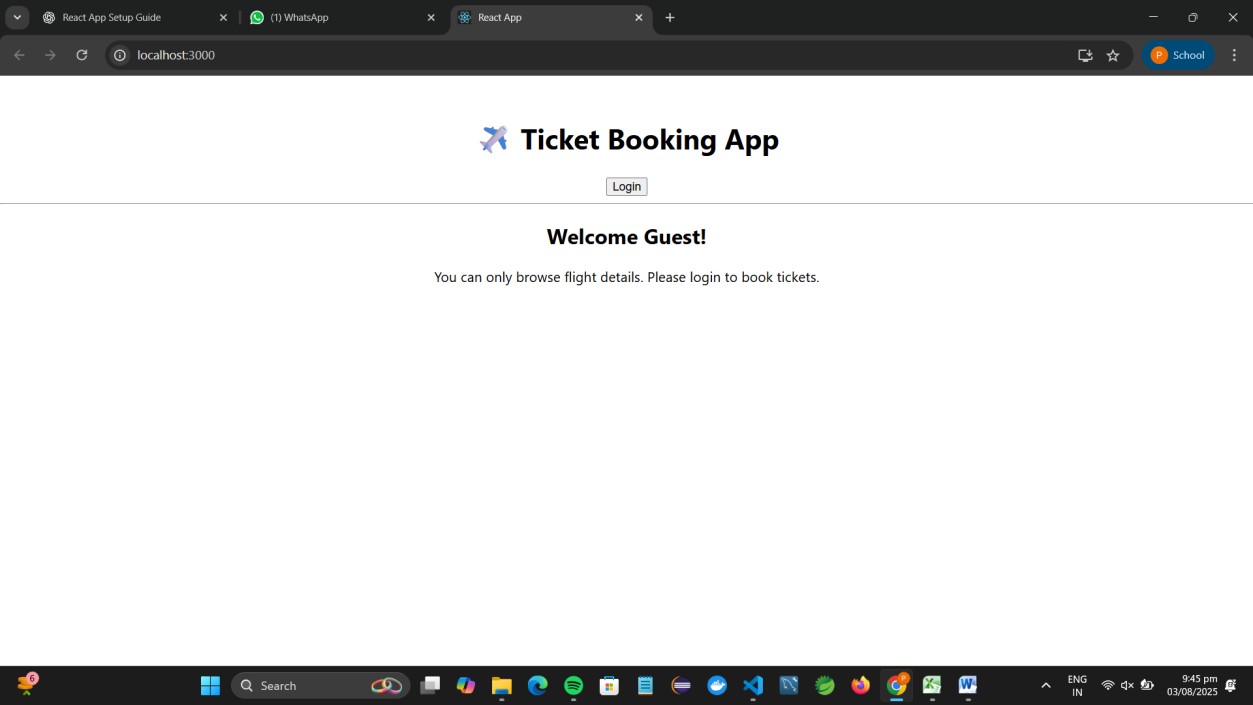
****

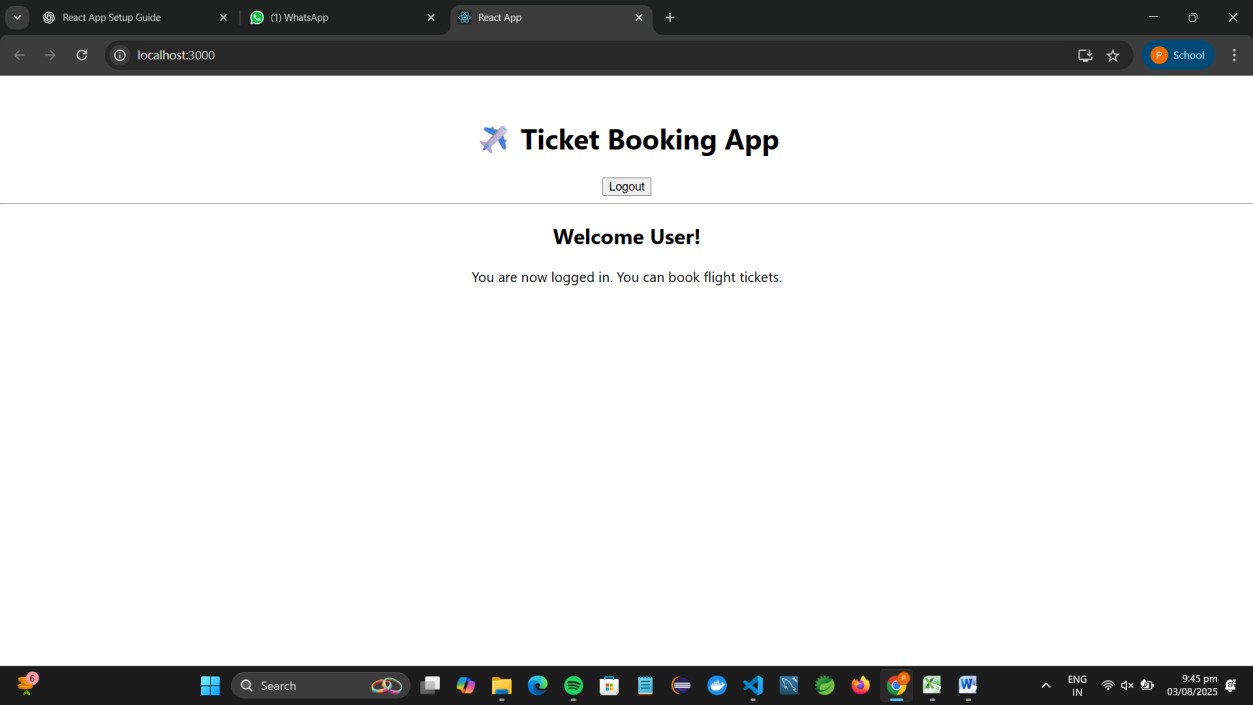
**Step 4: Run the App**

**In terminal, run:**

**npm start**

**Result:**

****

****

**13. ReactJS-HOL**

**Objectives**

* **Explain various ways of conditional rendering**
* **Explain how to render multiple components**
* **Define list component**
* **Explain about keys in React applications**
* **Explain how to extract components with keys**
* **Explain React Map, map() function**

**In this hands-on lab, you will learn how to:**

* **Implement conditional rendering in React applications**

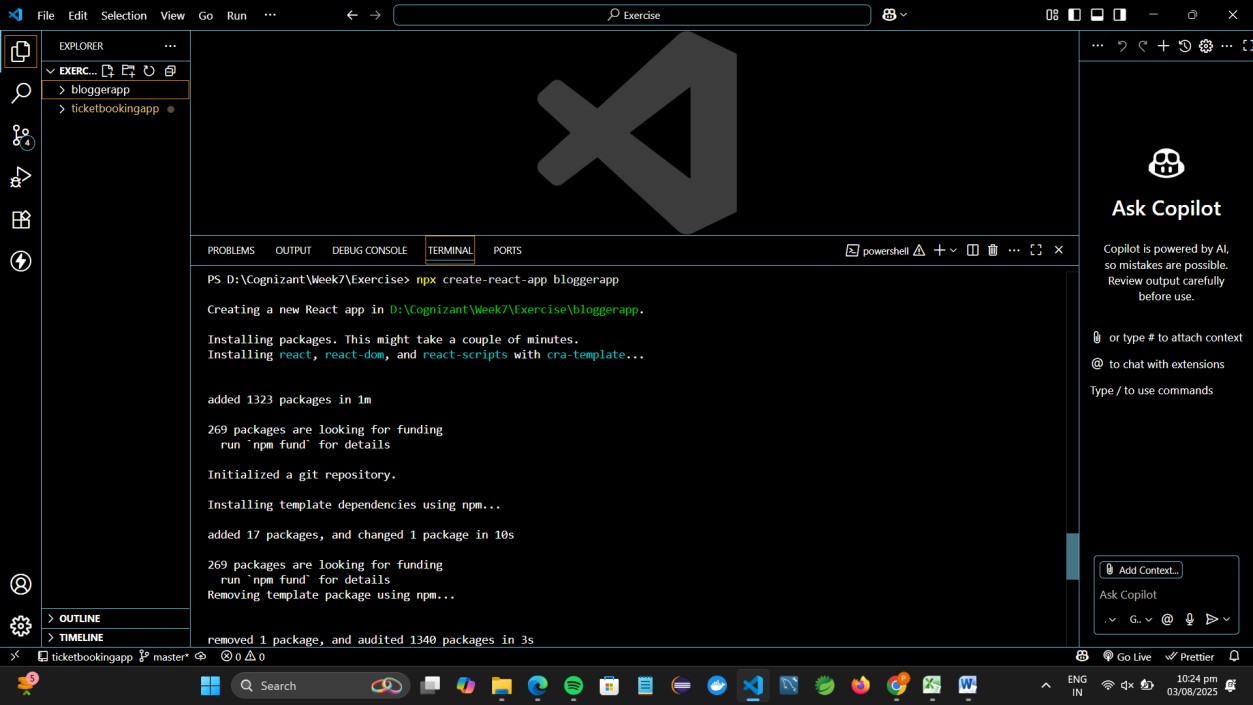
**Create a React App named “bloggerapp” in with 3 components.**

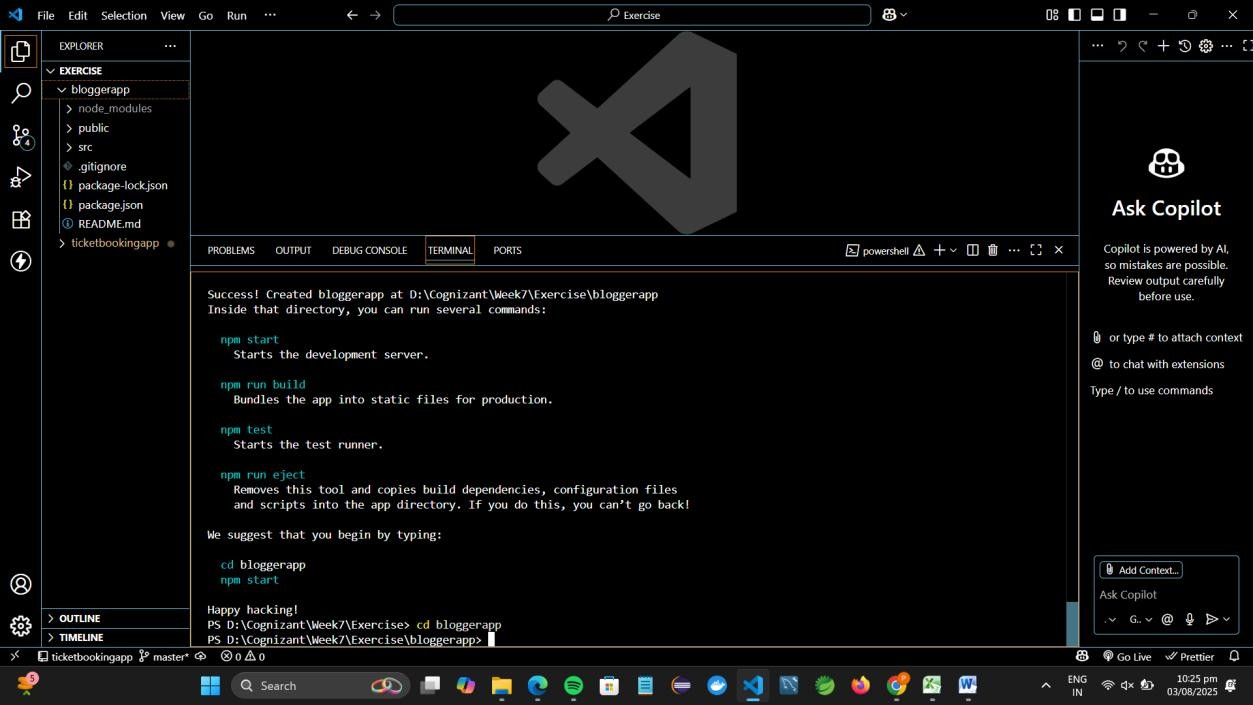
1. **Book Details**
2. **Blog Details**
3. **Course Details**

**Implement this with as many ways possible of Conditional Rendering.**

**Step 1: Create the React App**

**npx create-react-app bloggerapp cd bloggerapp**

****

****

**Step 2: Create Components Folder**

**Inside src, create a folder:**

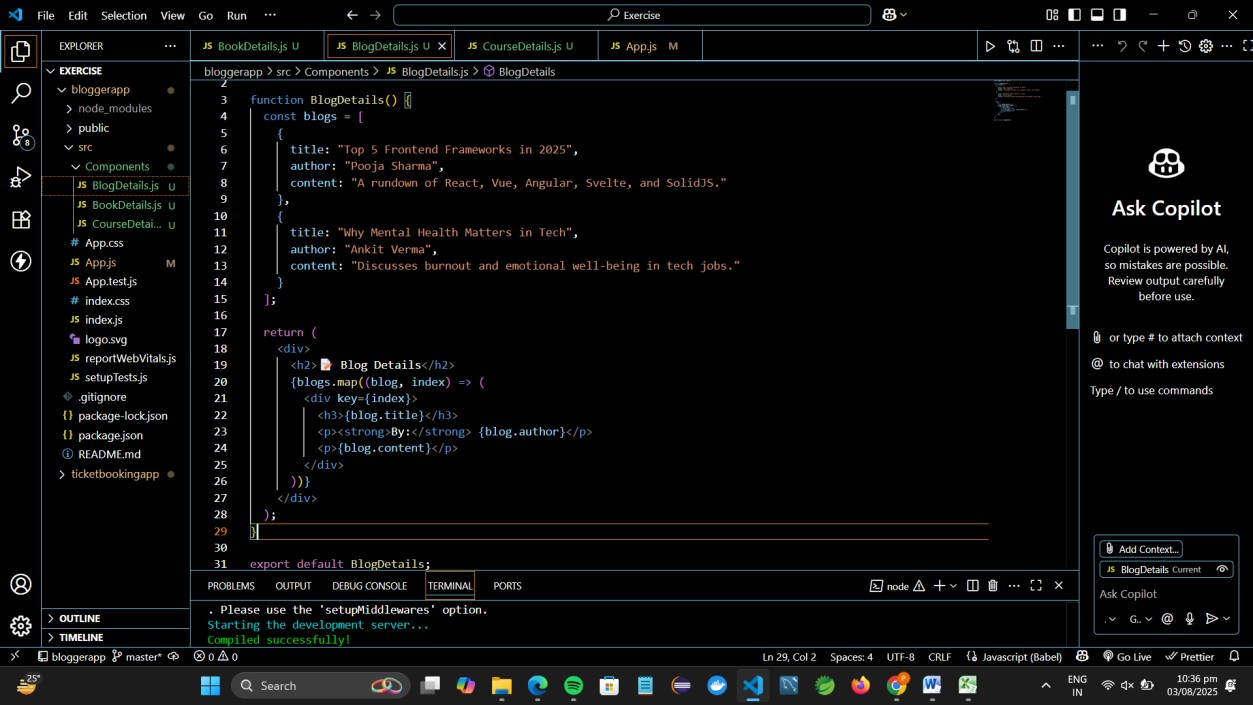
**Components**

**Create 3 components inside it:**

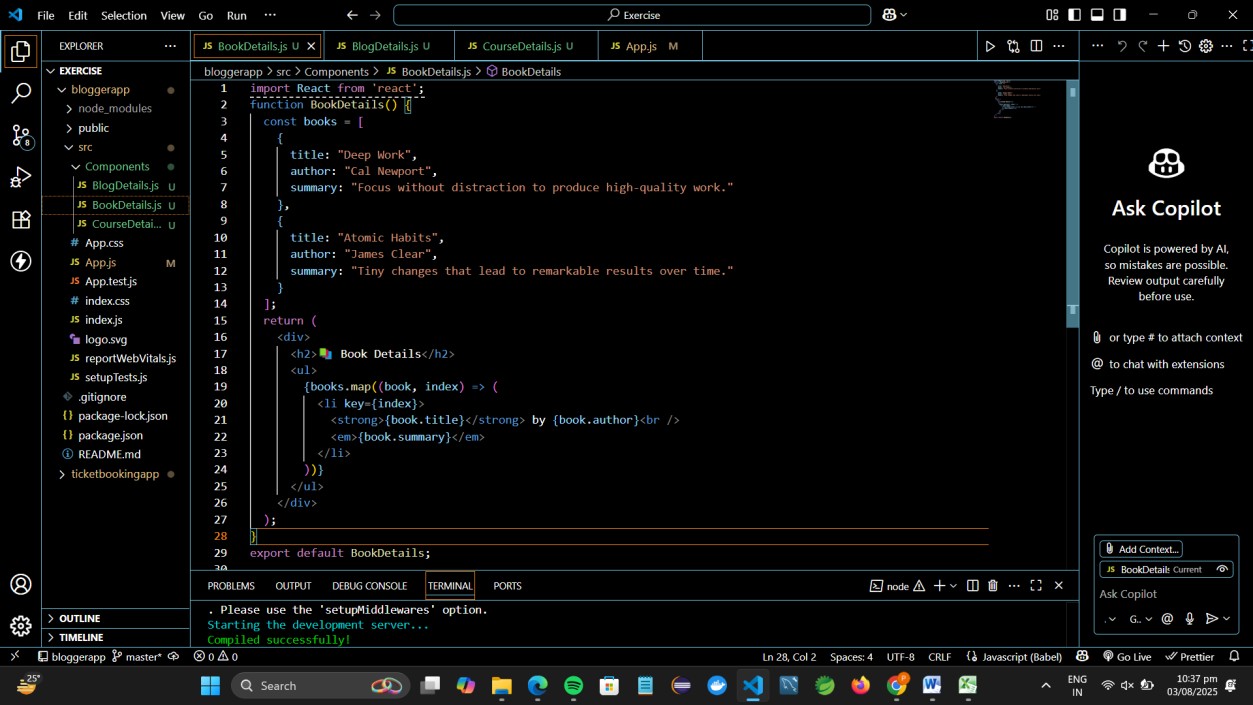
* **BookDetails.js**
* **BlogDetails.js**
* **CourseDetails.js**

****

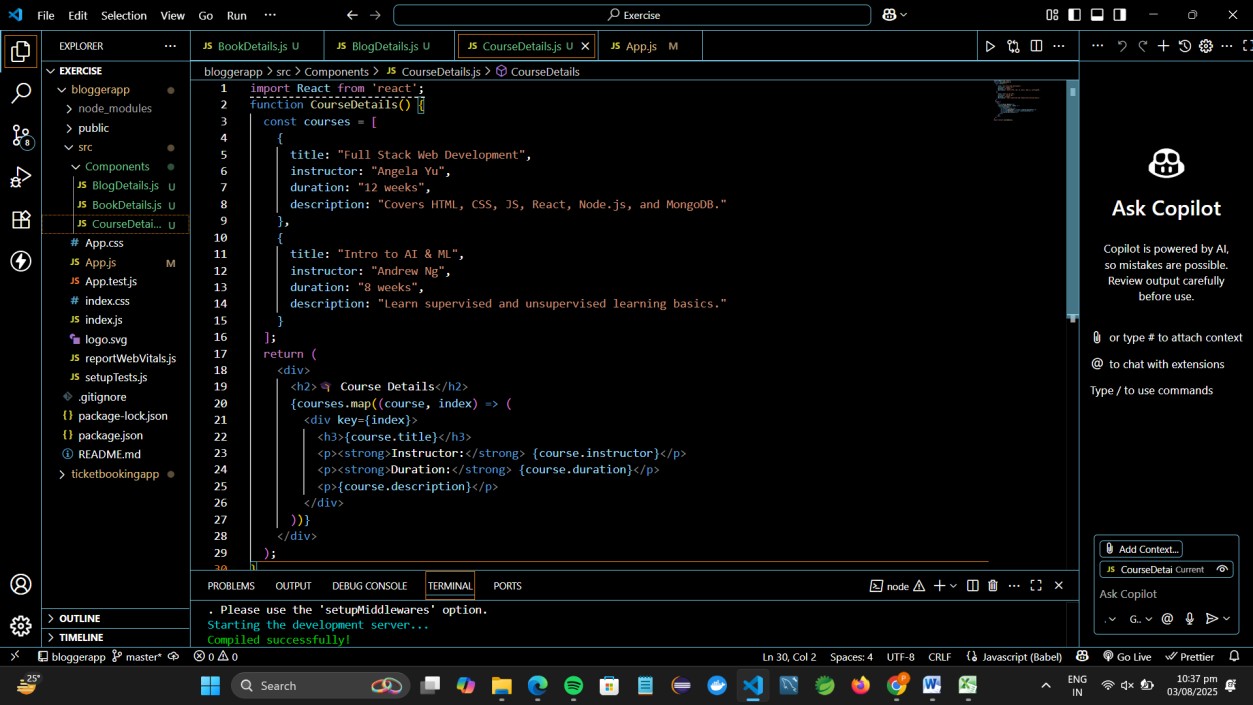
**BlogDetails.js:**

****

**BookDetails.js:**

****

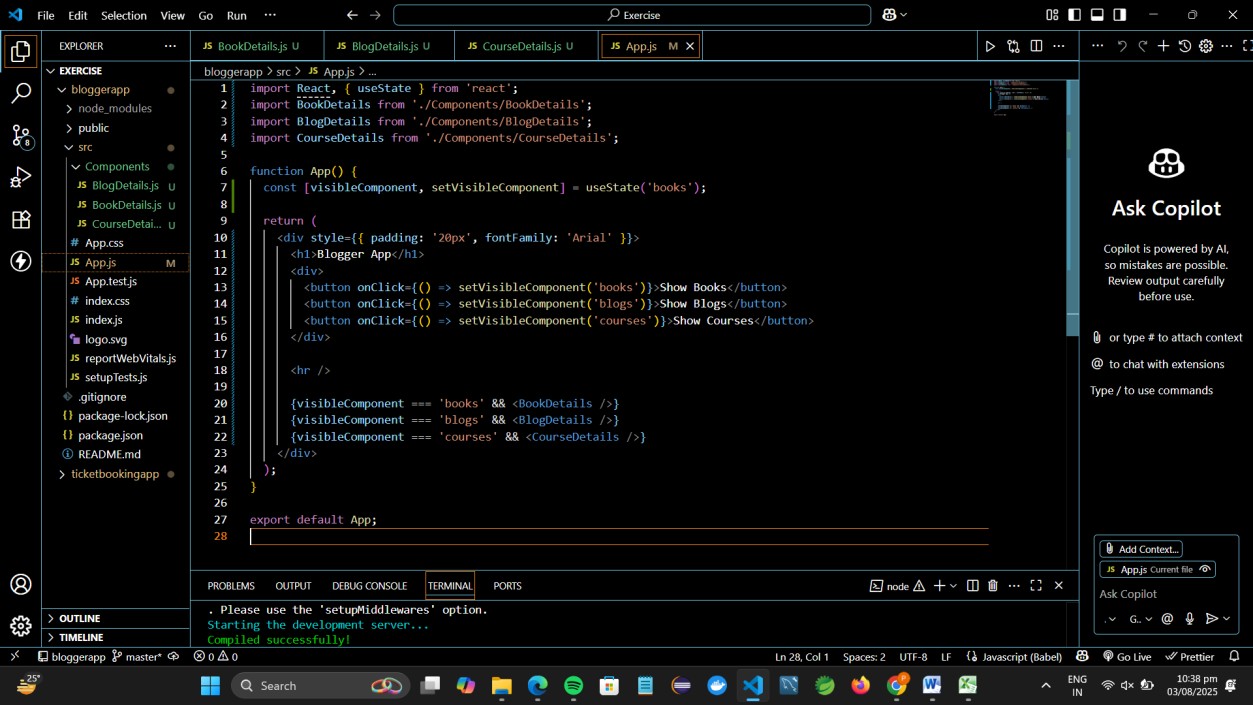
**CourseDetails.js:**

****

**Step 3: Update App.js to Use All Components**

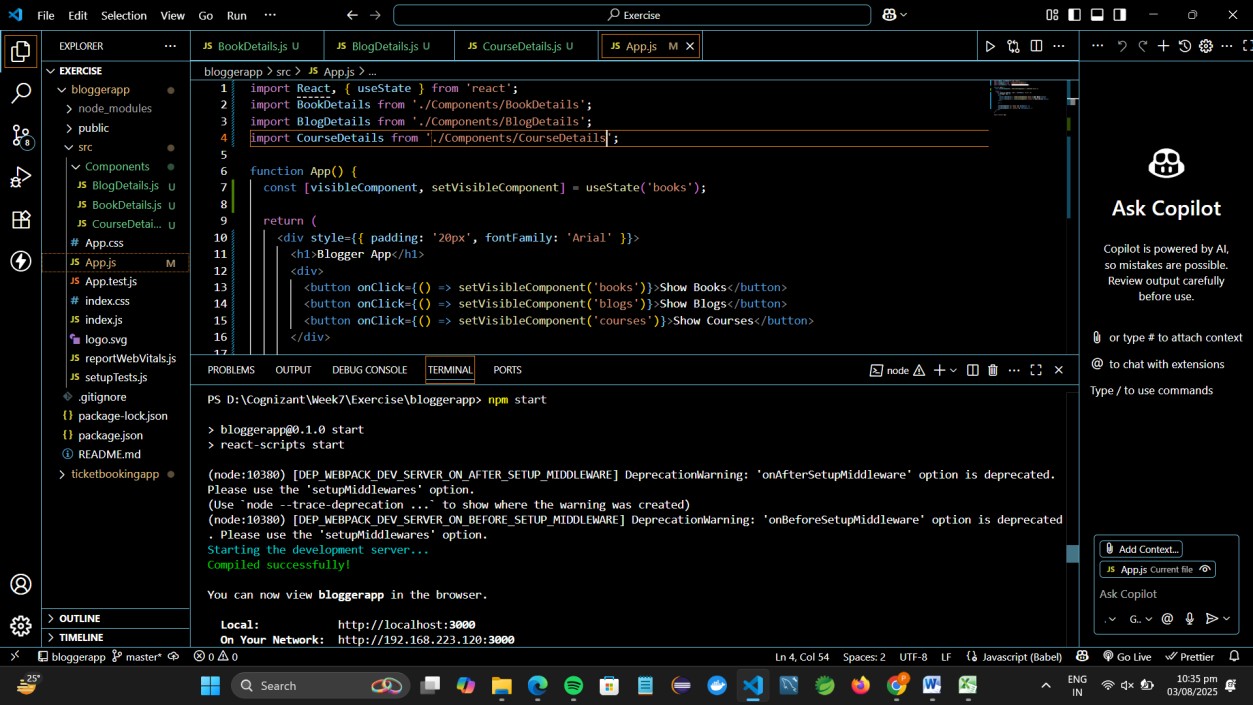
**Replace everything in src/App.js with:**

**App.js:**

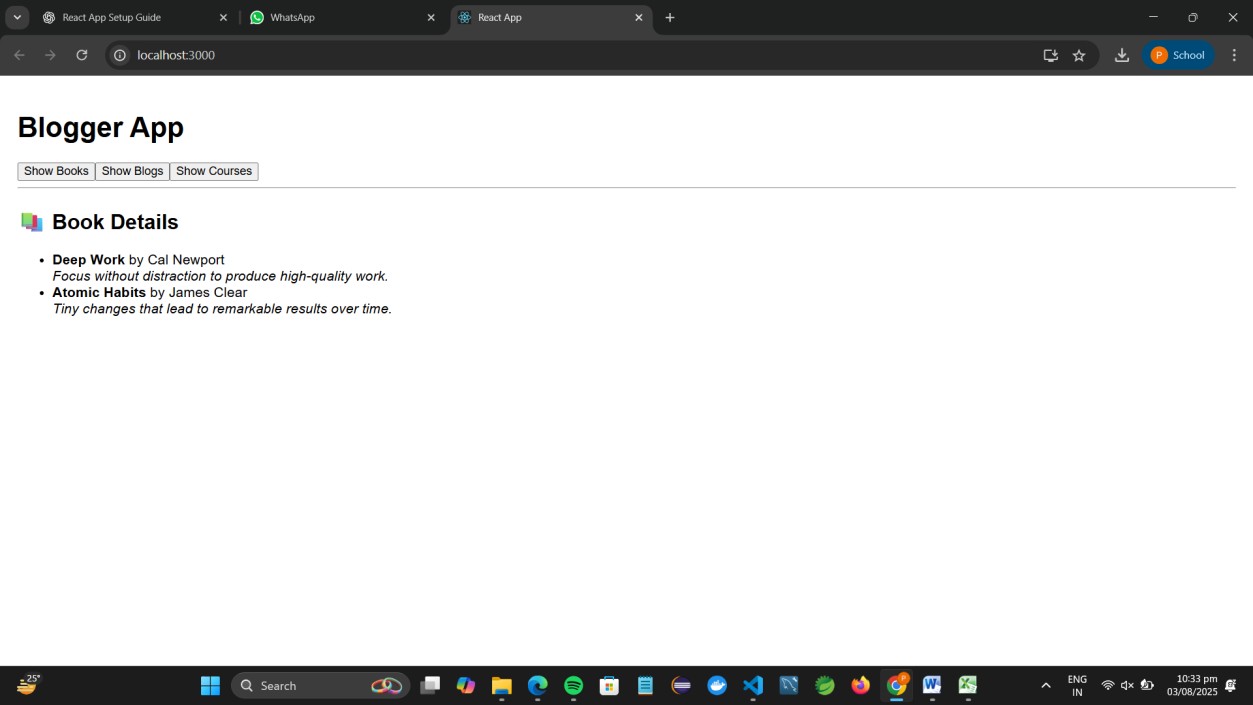
****

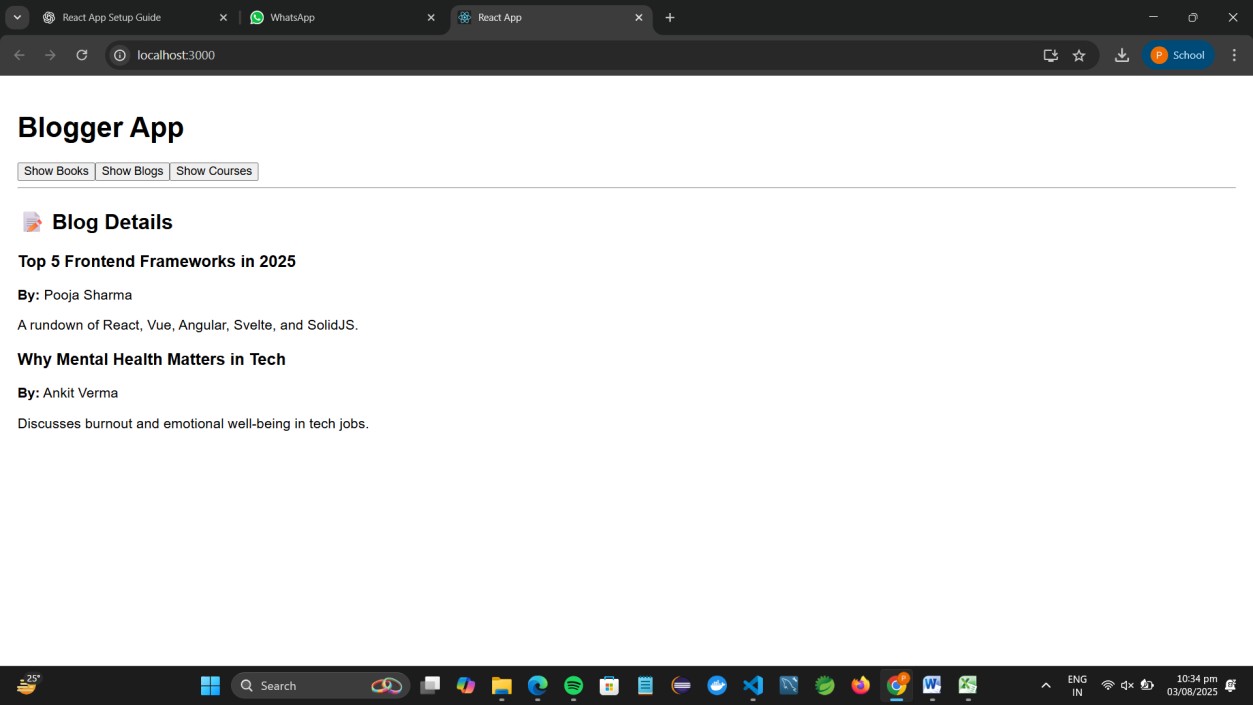
**Step 4: Run the App**

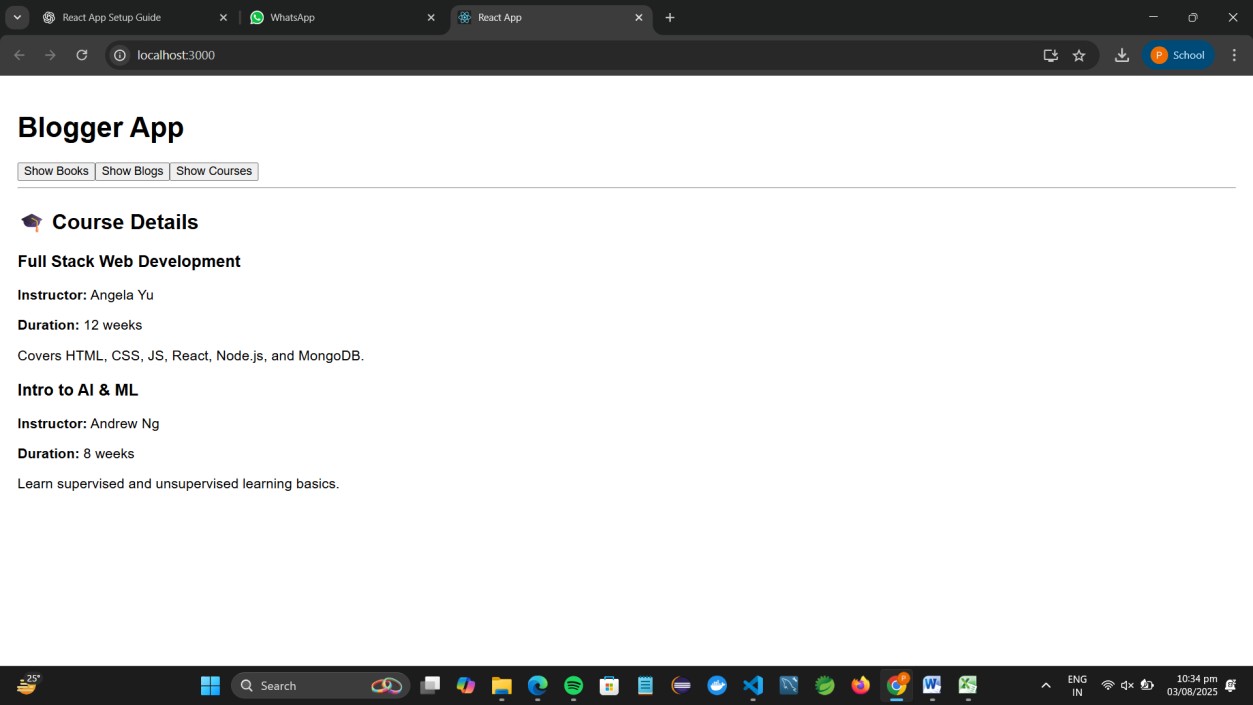
**npm start**

****

**Result:**

****

****

****