React- Week 7

## 1. **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

ListofPlayers.ja

import React from 'react';

const players = [

  { name: "Virat", score: 80 },

  { name: "Rohit", score: 65 },

  { name: "Dhoni", score: 90 },

  { name: "Kohli", score: 60 },

  { name: "Jadeja", score: 50 },

  { name: "Hardik", score: 85 },

  { name: "Rahul", score: 45 },

  { name: "Gill", score: 75 },

  { name: "Pant", score: 35 },

  { name: "Ashwin", score: 95 },

  { name: "Bumrah", score: 55 }

];

const ListofPlayers = () => {

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

  return (

    <div>

      <h2>All Players</h2>

      {players.map((player, i) => (

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

      ))}

      <h2>Filtered Players (score &lt; 70)</h2>

      {filtered.map((player, i) => (

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

      ))}

    </div>

  );

};

export default ListofPlayers;

IndianPlayers.ja

import React from 'react';

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

const RanjiPlayers = ["Pujara", "Rahane", "Shreyas"];

const IndianPlayers = () => {

  const merged = [...T20players, ...RanjiPlayers];

  const [odd, even] = [

    merged.filter((\_, i) => i % 2 !== 0),

    merged.filter((\_, i) => i % 2 === 0)

  ];

  return (

    <div>

      <h2>Merged Players</h2>

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

      <h2>Even Team</h2>

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

      <h2>Odd Team</h2>

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

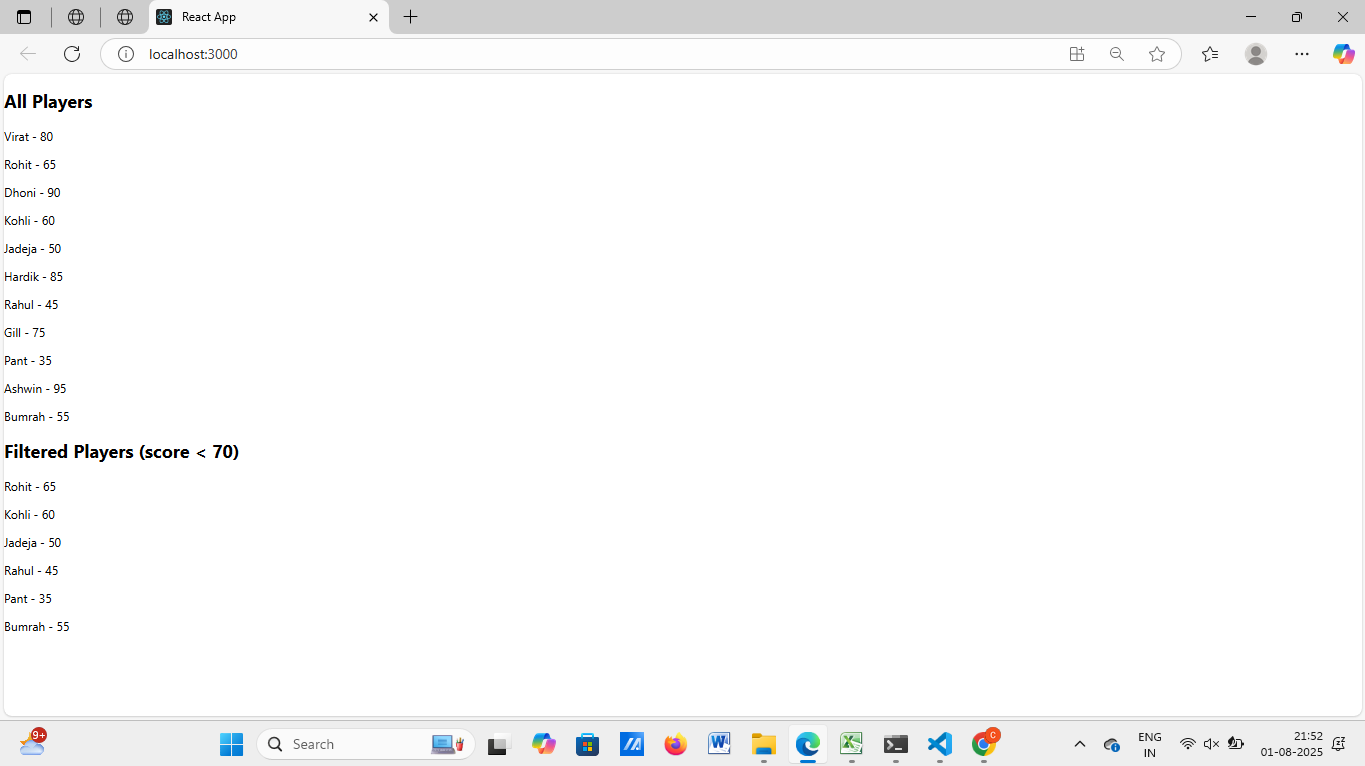
    </div>

  );

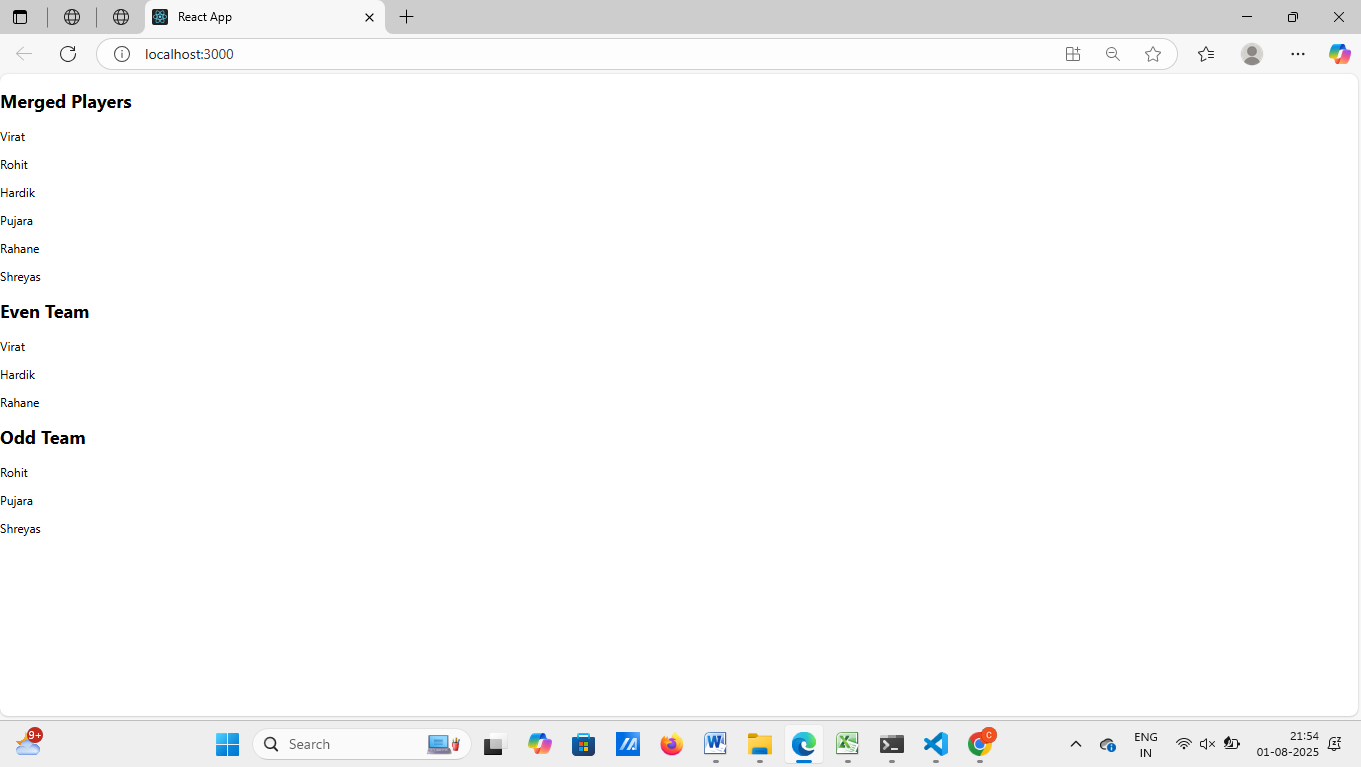
};

export default IndianPlayers;

When flag=true,



When flag=false,



## 2. **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

App.js

import React from 'react';

import './App.css';

const offices = [

  {

    name: "Tech Park",

    rent: 55000,

    address: "Coimbatore, Tamil Nadu",

    image: "https://via.placeholder.com/300x200?text=Tech+Park"

  },

  {

    name: "IT Hub",

    rent: 75000,

    address: "Chennai, Tamil Nadu",

    image: "https://via.placeholder.com/300x200?text=IT+Hub"

  },

  {

    name: "Business Center",

    rent: 45000,

    address: "Bangalore, Karnataka",

    image: "https://via.placeholder.com/300x200?text=Business+Center"

  }

];

function App() {

  return (

    <div className="App">

      <h1>🏢 Office Space Rental</h1>

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

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

          <img src={office.image} alt={office.name} width="300" height="200" />

          <h2>{office.name}</h2>

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

          <p style={{ color: office.rent > 60000 ? 'green' : 'red' }}>

            <strong>Rent:</strong> ₹{office.rent}

          </p>

        </div>

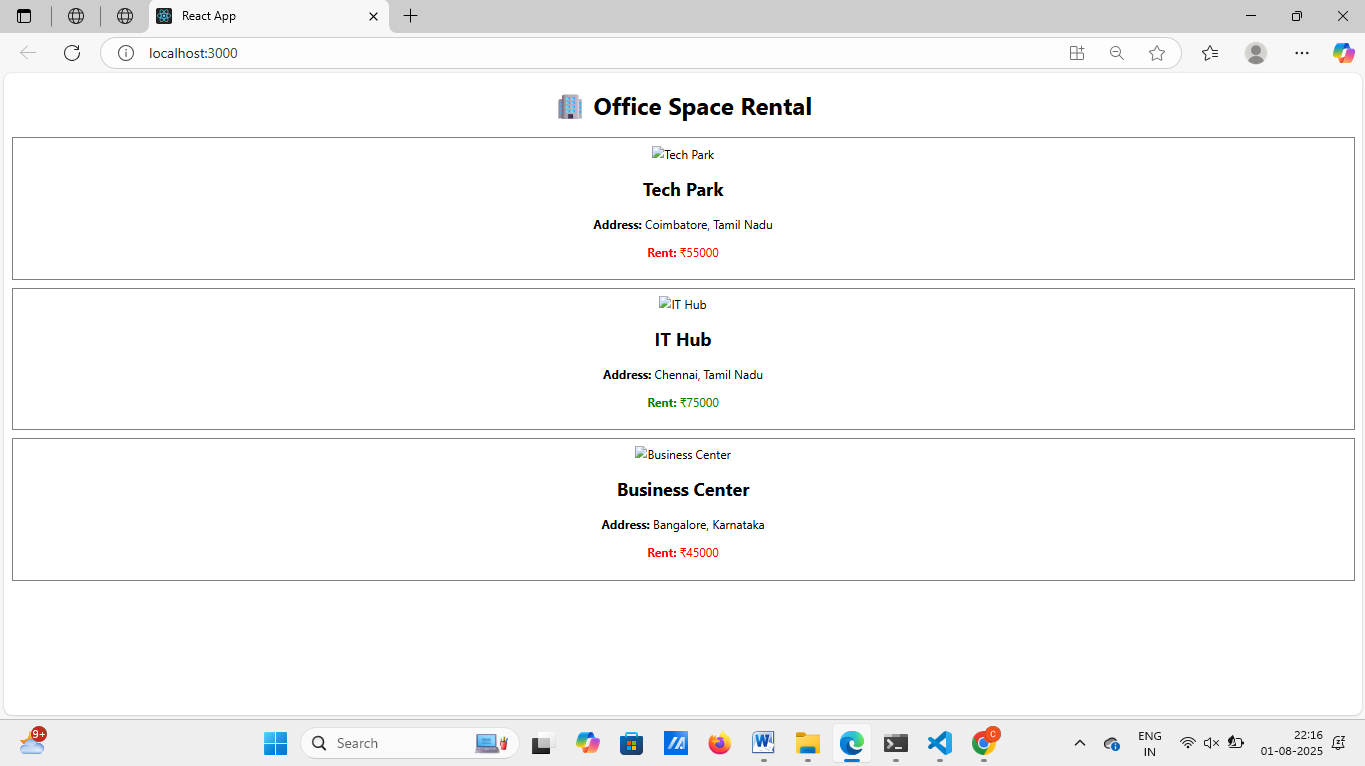
      ))}

    </div>

  );

}

export default App;



3. 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.

CurrencyConverter.js

import React, { useState } from 'react';

const CurrencyConvertor = () => {

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

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

const handleSubmit = () => {

const rate = 0.011; // 1 INR = 0.011 EUR approx

setEuro((parseFloat(rupees) \* rate).toFixed(2));

};

return (

<div>

<h2>Currency Convertor</h2>

<input

type="text"

placeholder="Enter INR"

value={rupees}

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

/>

<button onClick={handleSubmit}>Convert</button>

<p>EUR: {euro}</p>

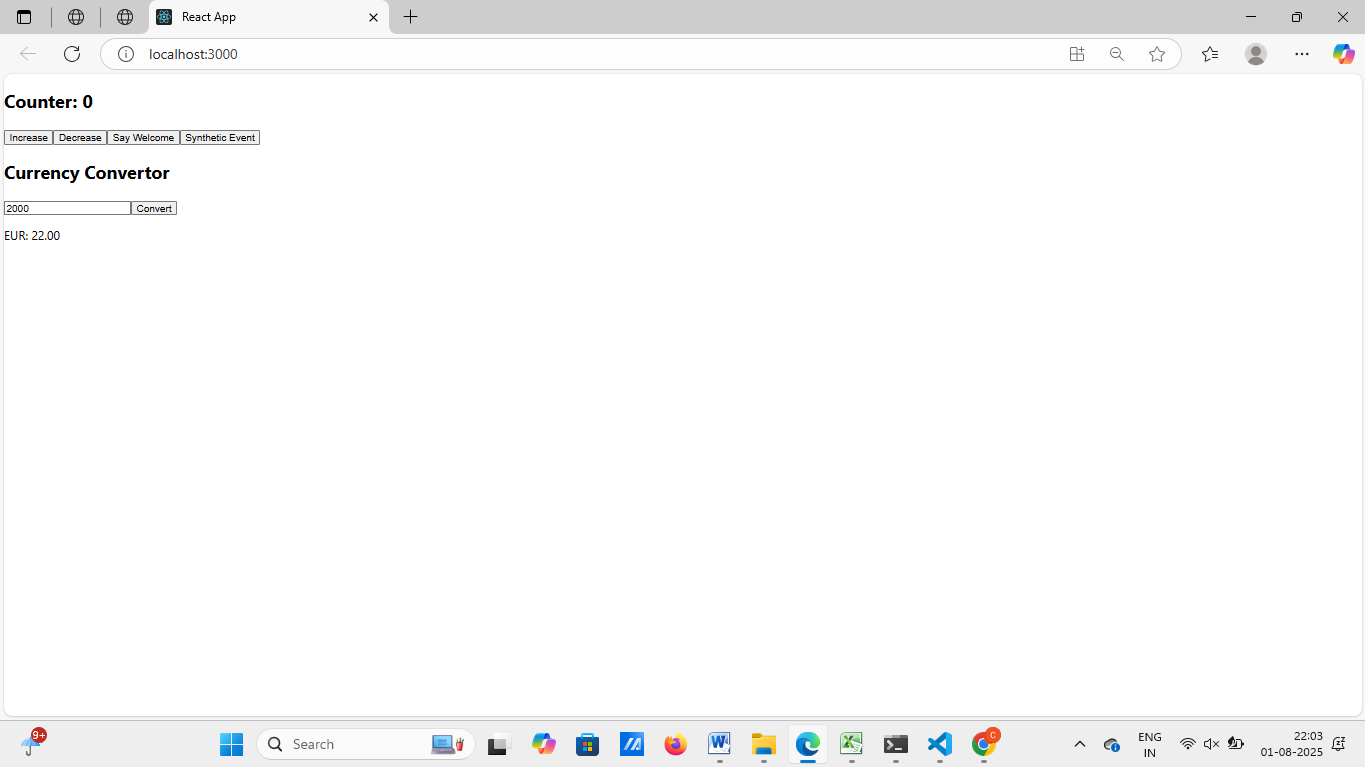
</div>

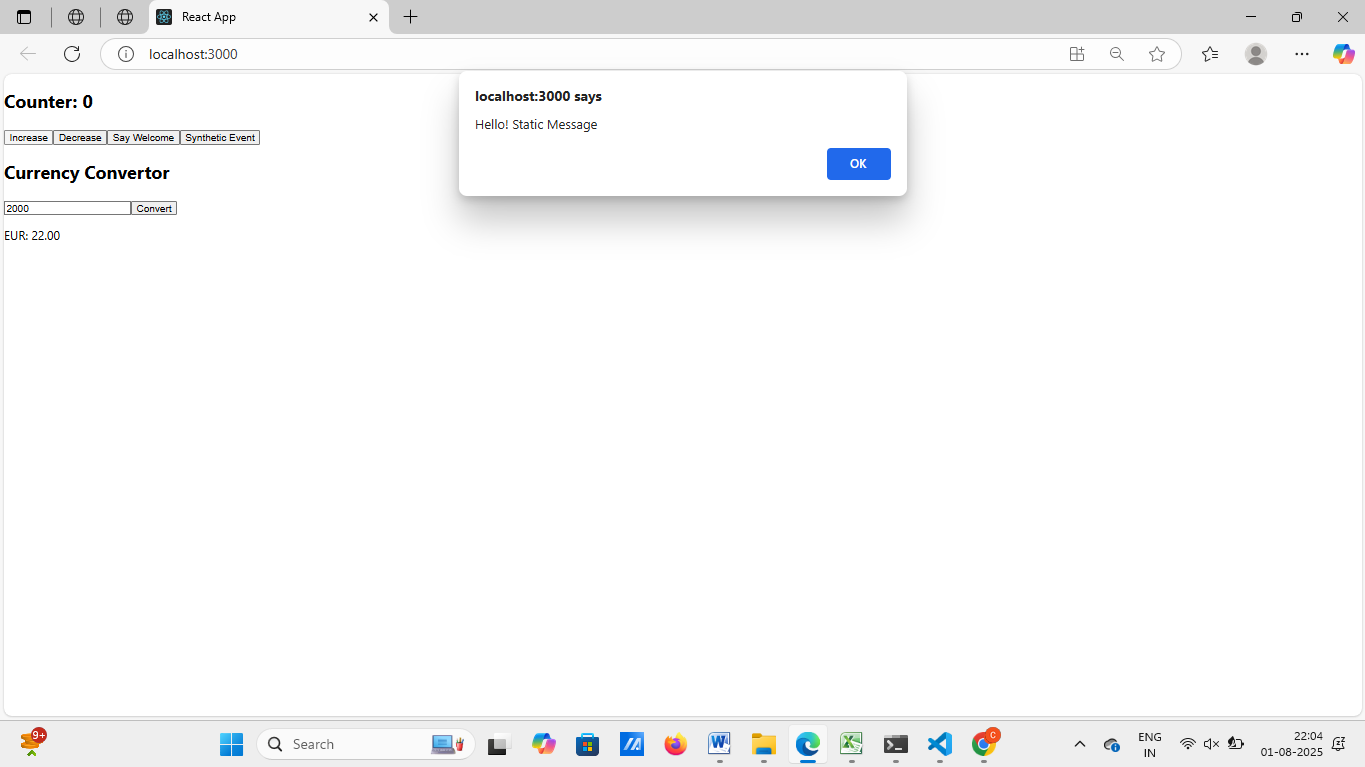
);

};

export default CurrencyConvertor;

output:



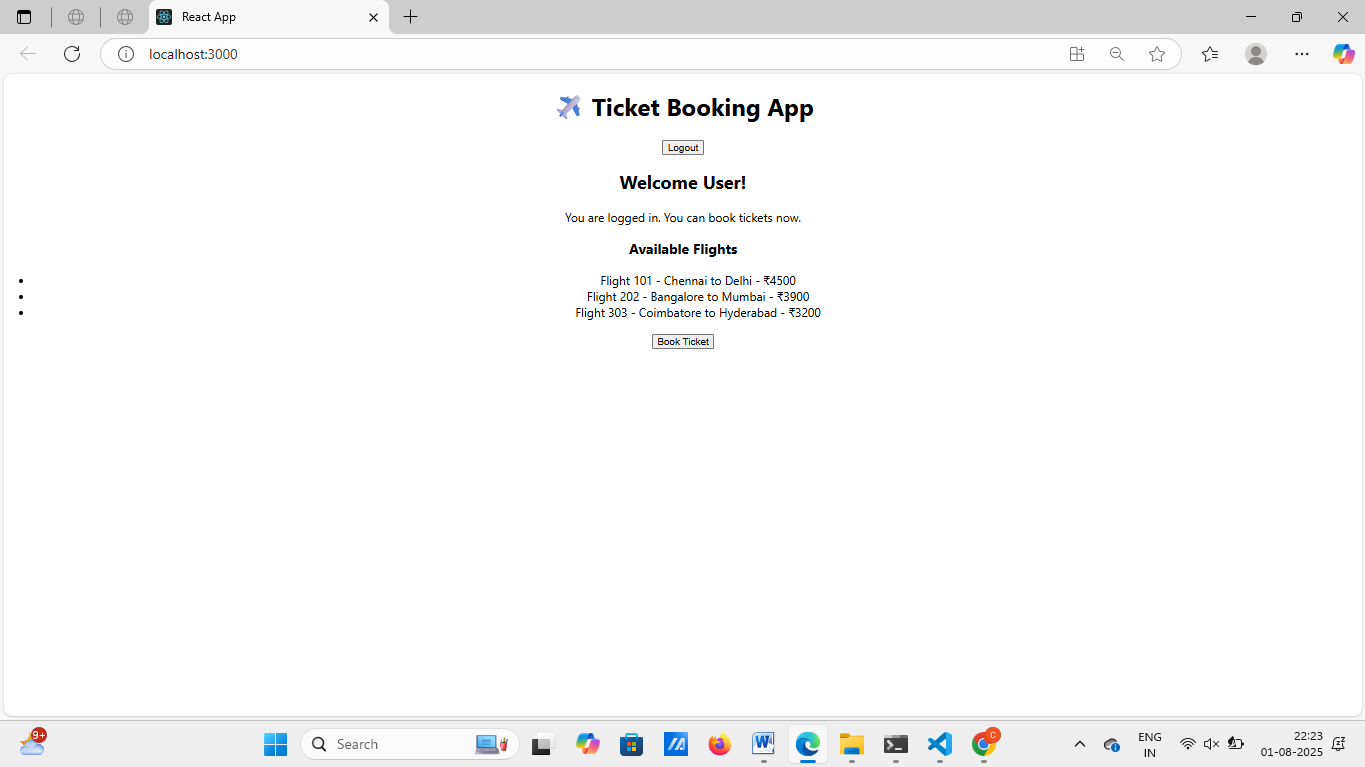


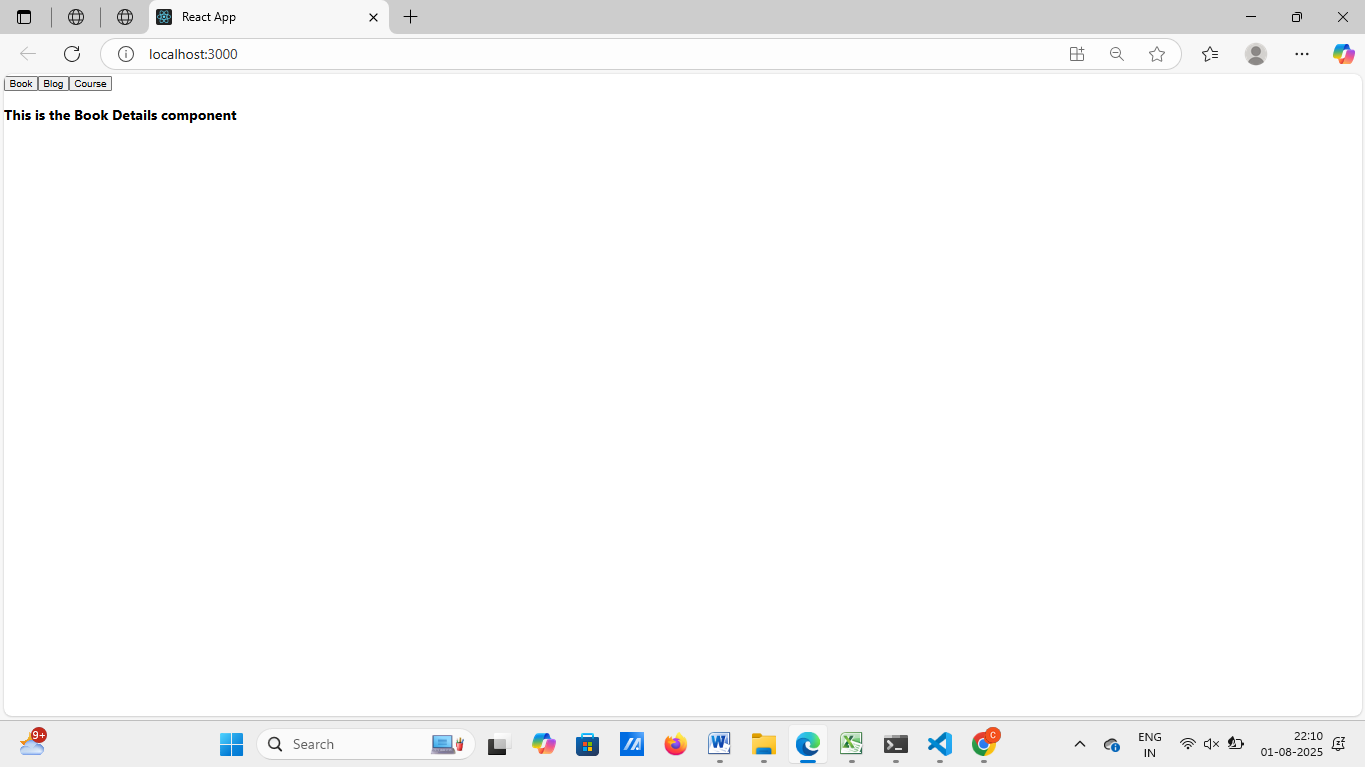
4. Estimated time to complete this lab: **60 minutes.**

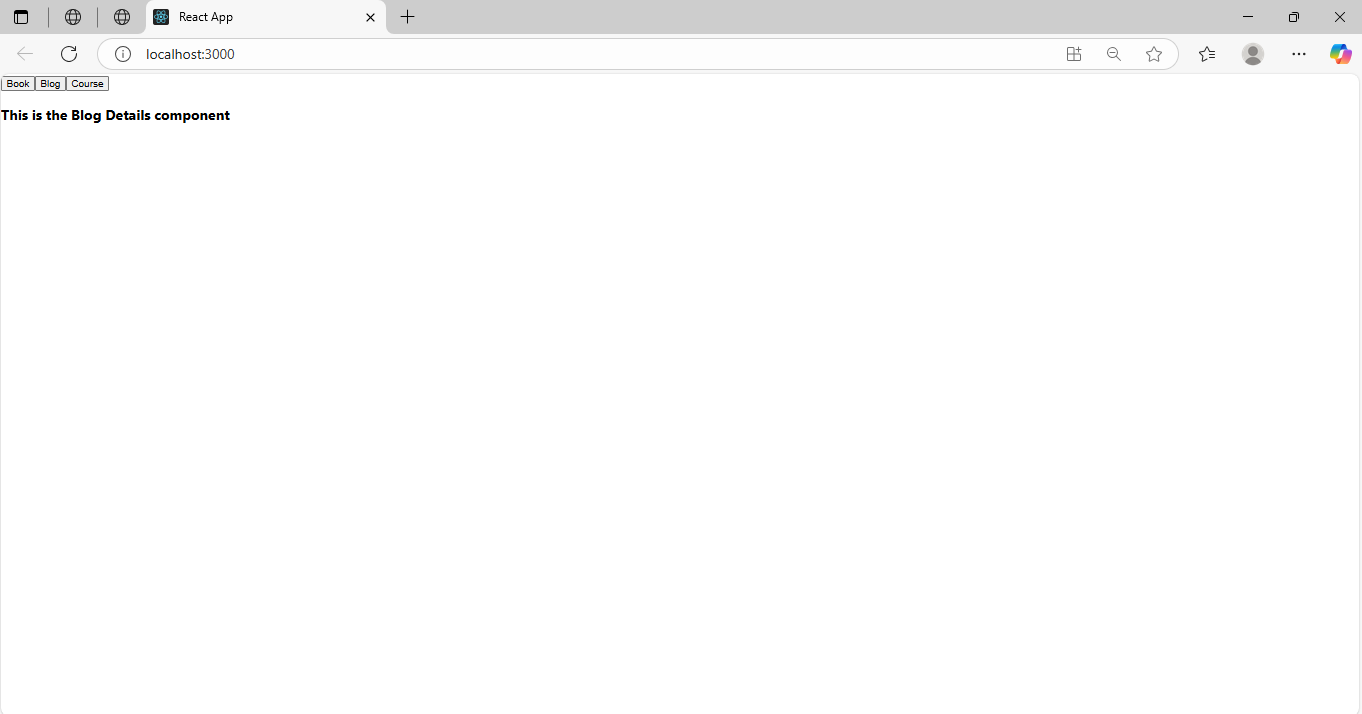
Create a React App named “bloggerapp” in with 3 components.

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

Implement this with as many ways possible of Conditional Rendering.







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

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.

import React, { useState } from 'react';

import './App.css';

import BookDetails from './BookDetails';

import BlogDetails from './BlogDetails';

import CourseDetails from './CourseDetails';

function App() {

const [selected, setSelected] = useState('');

// Element variable method

let content;

if (selected === 'book') {

content = <BookDetails />;

} else if (selected === 'blog') {

content = <BlogDetails />;

} else if (selected === 'course') {

content = <CourseDetails />;

} else {

content = <p>Please select an option to view details.</p>;

}

return (

<div className="App">

<h1>📝 BloggerApp</h1>

<div>

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

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

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

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

</div>

<hr />

{/\* 1. Using if-else with element variable \*/}

{content}

{/\* 2. Using ternary operator \*/}

<div>

<h3>Using Ternary:</h3>

{selected === 'book' ? <BookDetails /> : null}

</div>

{/\* 3. Using logical AND (&&) operator \*/}

<div>

<h3>Using && operator:</h3>

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

</div>

{/\* 4. Using switch-case inside a function \*/}

<div>

<h3>Using switch-case:</h3>

{renderComponentBySwitch(selected)}

</div>

</div>

);

}

// Helper function: switch-case rendering

function renderComponentBySwitch(key) {

switch (key) {

case 'book':

return <BookDetails />;

case 'blog':

return <BlogDetails />;

case 'course':

return <CourseDetails />;

default:

return null;

}

}

export default App;

