1. **ReactJS-HOL**

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.

Code:

import React, { useState } from 'react';

import './App.css';

import ListofPlayers from './components/ListofPlayers';

import Scorebelow70 from './components/Scorebelow70';

import { OddPlayers, EvenPlayers, IndianPlayers, ListofindianPlayers } from './components/IndianPlayers';

function **App**() {

*// Players array with names and scores*

    const players = [

        { name: 'Jack', score: 50 },

        { name: 'Michael', score: 70 },

        { name: 'John', score: 40 },

        { name: 'Ann', score: 61 },

        { name: 'Elisabeth', score: 61 },

        { name: 'Sachin', score: 95 },

        { name: 'Dhoni', score: 100 },

        { name: 'Virat', score: 84 },

        { name: 'Jadeja', score: 64 },

        { name: 'Raina', score: 75 },

        { name: 'Rohit', score: 80 }

    ];

*// Indian team array for destructuring*

    const IndianTeam = ['Sachin1', 'Dhoni2', 'Virat3', 'Rohit4', 'Yuvraj5', 'Raina6'];

*// State to control which component to display*

    const [flag, **setFlag**] = useState(true);

    const **toggleFlag** = () => {

        setFlag(!flag);

    };

    return (

        <div>

            <div *style*={{ textAlign: 'center', margin: '20px' }}>

                <h2>ES6 Cricket App Demo</h2>

                <button

*onClick*={toggleFlag}

*style*={{

                        padding: '10px 20px',

                        fontSize: '16px',

                        backgroundColor: flag ? '#4CAF50' : '#2196F3',

                        color: 'white',

                        border: 'none',

                        borderRadius: '5px',

                        cursor: 'pointer'

                    }}

                >

                    {flag ? 'Switch to Indian Team View' : 'Switch to Players List View'}

                </button>

                <p>Current Flag Value: {flag.toString()}</p>

            </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(IndianTeam)}

                        <hr />

                        <h1> Even Players</h1>

                        {EvenPlayers(IndianTeam)}

                    </div>

                    <hr />

                    <div>

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

                        <ListofindianPlayers *IndianPlayers*={IndianPlayers} />

                    </div>

                </div>

            )}

        </div>

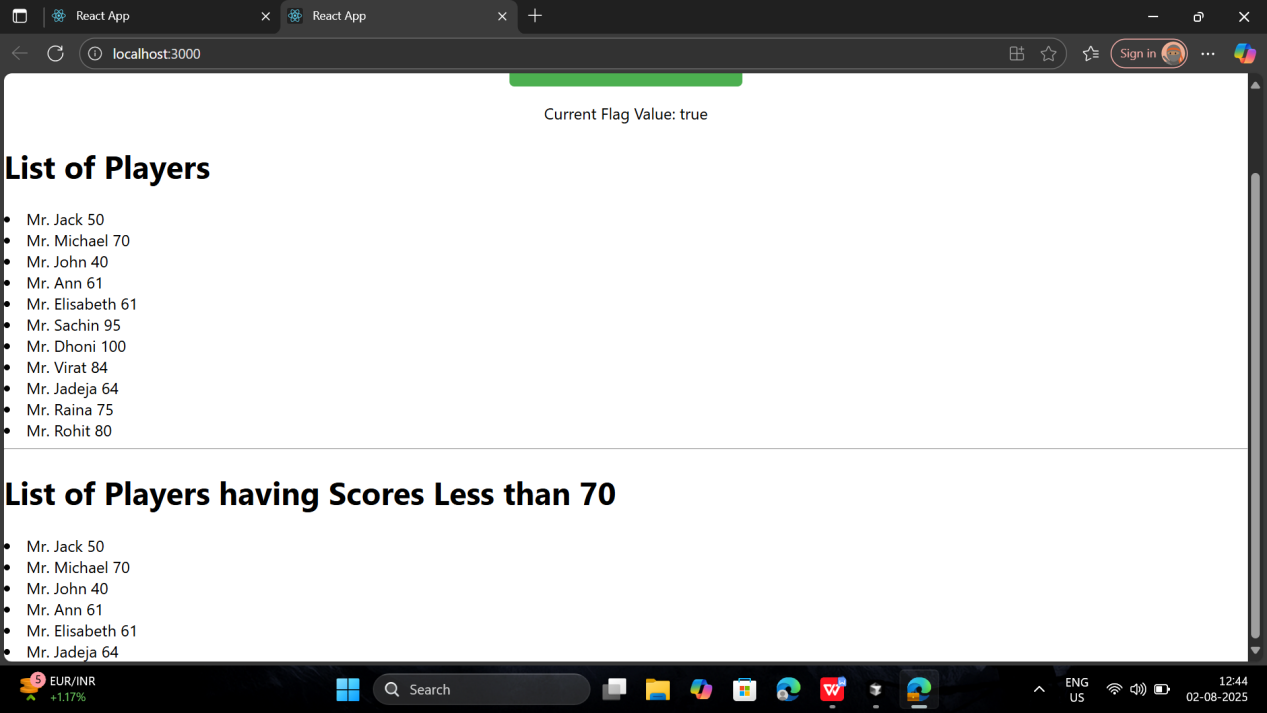
    );

}

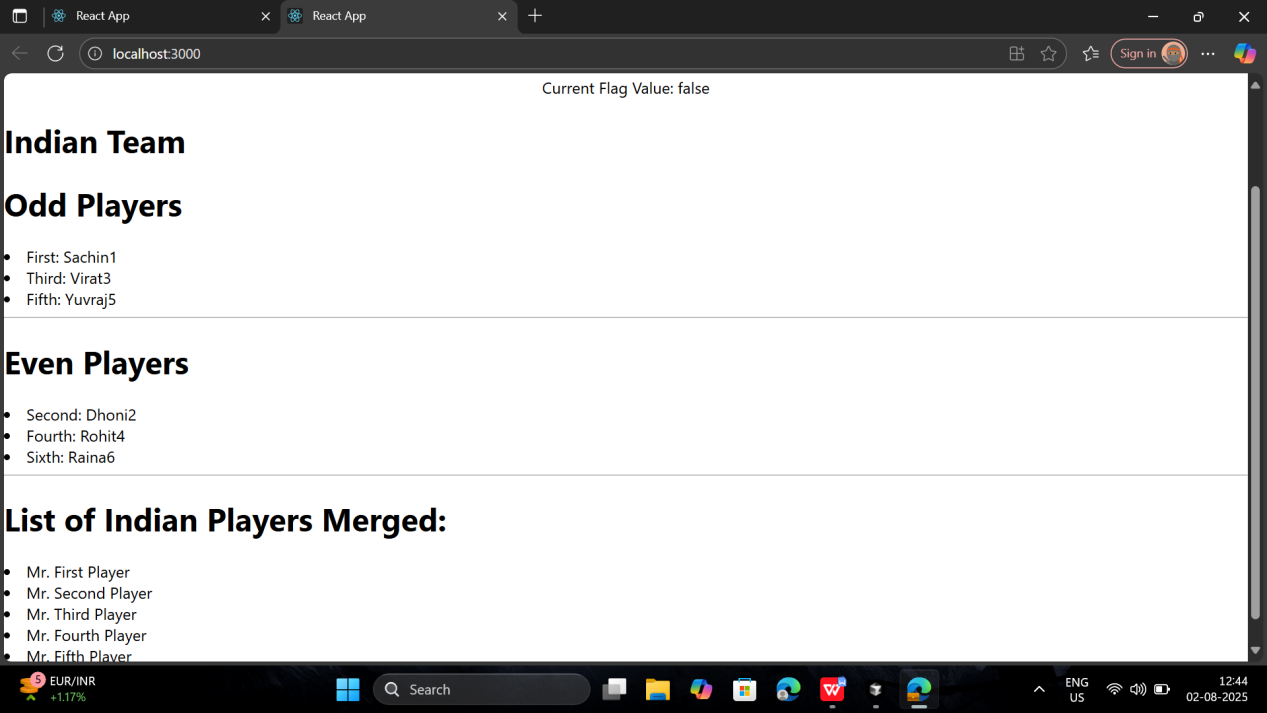
export default App;

**Output:**

When Flag=true



When Flag=false



1. **ReactJS-HOL**

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

Create a React Application named “officespacerentalapp” which uses React JSX to create elements, 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:

import React from 'react';

import './App.css';

function App() {

*// Office space data*

  const officeSpaces = [

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

    { Name: "Tech Park", Rent: 75000, Address: "Bangalore" },

    { Name: "Business Center", Rent: 45000, Address: "Mumbai" },

    { Name: "Corporate Plaza", Rent: 80000, Address: "Delhi" }

  ];

*// Image source (you can replace with actual image URL)*

  const sr = "https://images.unsplash.com/photo-1497366216548-37526070297c?w=500&h=300&fit=crop";

  return (

    <div *className*="App">

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

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

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

        let colors = [];

        if (item.Rent <= 60000) {

          colors.push('textRed');

        } else {

          colors.push('textGreen');

        }

        return (

          <div *key*={index} *className*="office-item">

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

            <h3 *className*={colors[0]}>Rent: Rs. {item.Rent}</h3>

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

          </div>

        );

      })}

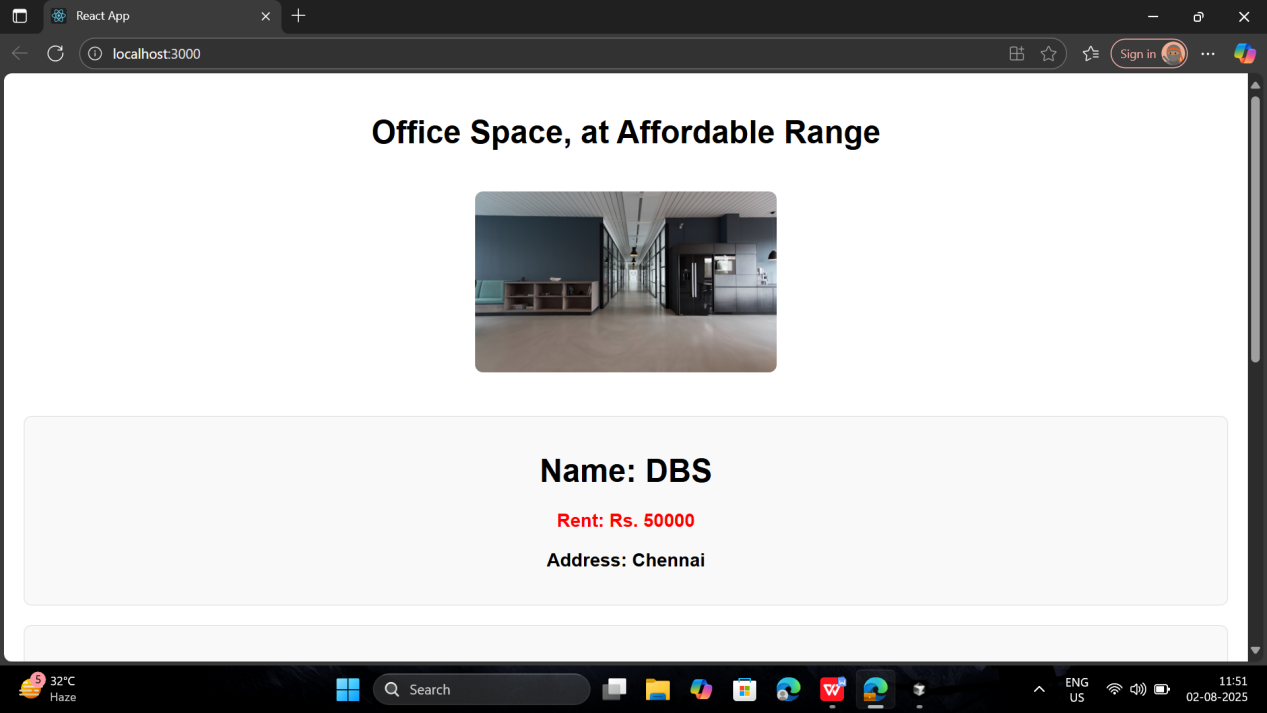
    </div>

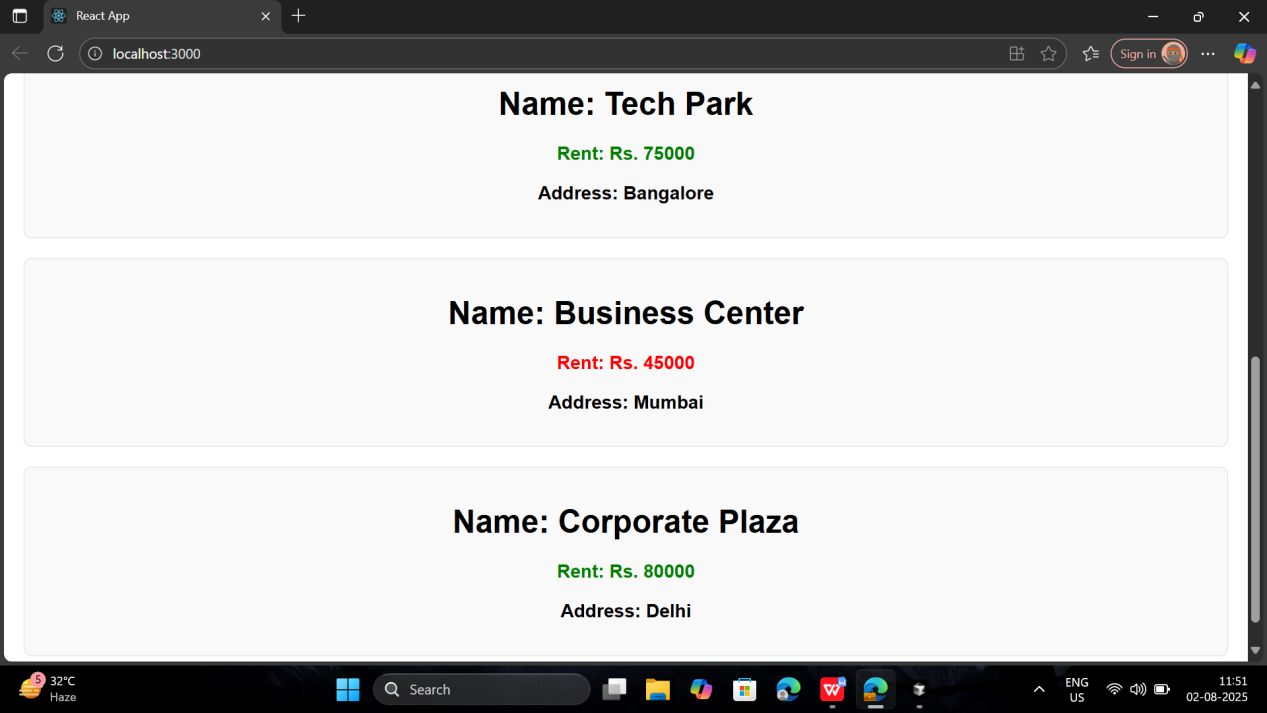
  );

}

export default App;

**Output:**





1. **ReactJS-HOL**

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.

Code:

import React, { Component } from 'react';

import './App.css';

class App extends Component {

  constructor(*props*) {

*super*(props);

    this.state = {

      counter: 5,

      amount: '',

      currency: ''

    };

  }

*// Task 1: Increment and Decrement with multiple methods*

  incrementCounter = () => {

    this.setState(*prevState* => ({

      counter: prevState.counter + 1

    }));

  }

  sayHello = () => {

    alert('localhost:3000 says hello member1');

  }

  handleIncrement = () => {

    this.incrementCounter();

    this.sayHello();

  }

  decrementCounter = () => {

    this.setState(*prevState* => ({

      counter: prevState.counter - 1

    }));

  }

*// Task 2: Say Welcome with argument*

  sayWelcome = (*message*) => {

    alert(`localhost:3000 says ${message}`);

  }

*// Task 3: Synthetic event*

  handleClick = (*event*) => {

    alert('localhost:3000 says I was clicked');

  }

*// Currency Converter*

  handleAmountChange = (*event*) => {

    this.setState({

      amount: event.target.value

    });

  }

  handleCurrencyChange = (*event*) => {

    this.setState({

      currency: event.target.value

    });

  }

  handleSubmit = (*event*) => {

    event.preventDefault();

    const { amount, currency } = this.state;

    if (currency.toLowerCase() === 'euro') {

      const euroAmount = (parseFloat(amount) \* 0.011).toFixed(2);

      alert(`localhost:3000 says Converting to Euro Amount is ${euroAmount}`);

    }

  }

  render() {

    return (

      <div *className*="App">

        <header *className*="App-header">

          <h1>Event Handling Examples</h1>

          {*/\* Counter Section \*/*}

          <div>

            <h2>Counter: {this.state.counter}</h2>

            <button *onClick*={this.handleIncrement}>Increment</button>

            <button *onClick*={this.decrementCounter}>Decrement</button>

            <button *onClick*={() => this.sayWelcome('welcome')}>Say Welcome</button>

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

          </div>

          {*/\* Currency Converter Section \*/*}

          <div>

            <h2>Currency Convertor!!!</h2>

            <form *onSubmit*={this.handleSubmit}>

              <div>

                <label>Amount: </label>

                <input

*type*="number"

*value*={this.state.amount}

*onChange*={this.handleAmountChange}

*placeholder*="Enter amount"

                />

              </div>

              <div>

                <label>Currency: </label>

                <input

*type*="text"

*value*={this.state.currency}

*onChange*={this.handleCurrencyChange}

*placeholder*="Enter currency"

                />

              </div>

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

            </form>

          </div>

        </header>

      </div>

    );

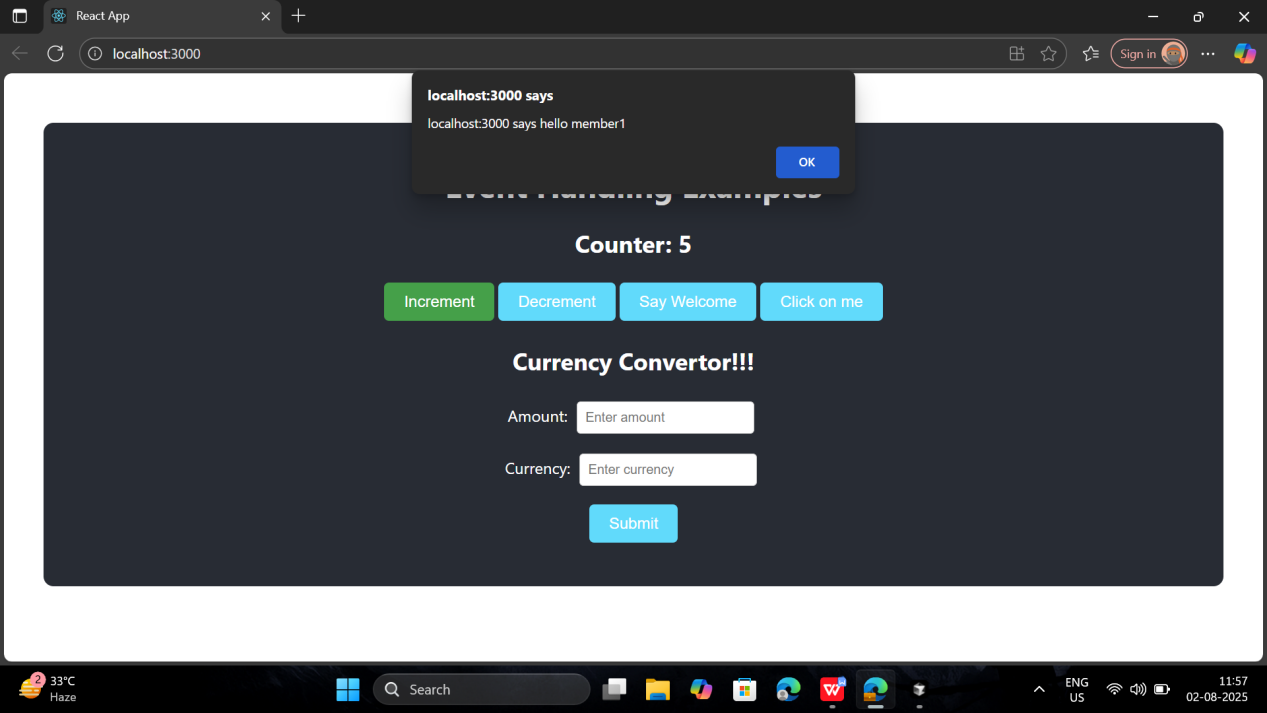
  }

}

export default App;

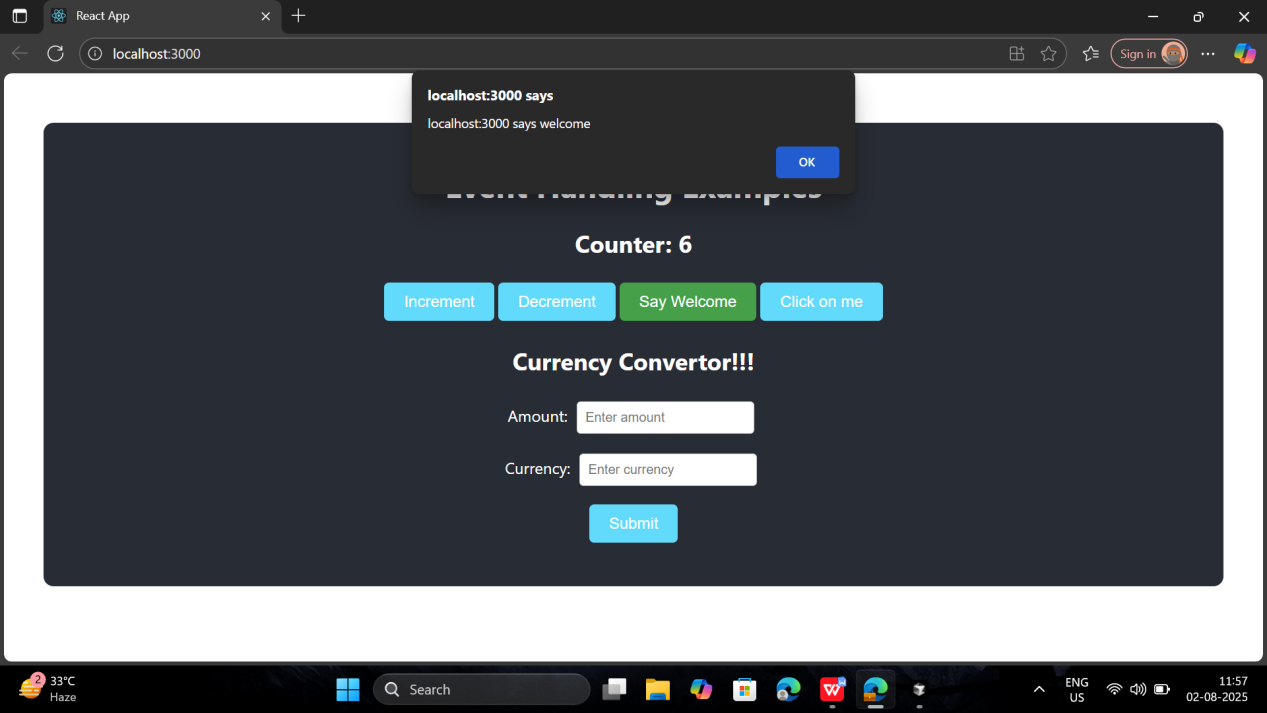
* 1. To increment the value
  2. Say Hello followed by a static message.

**Output:**



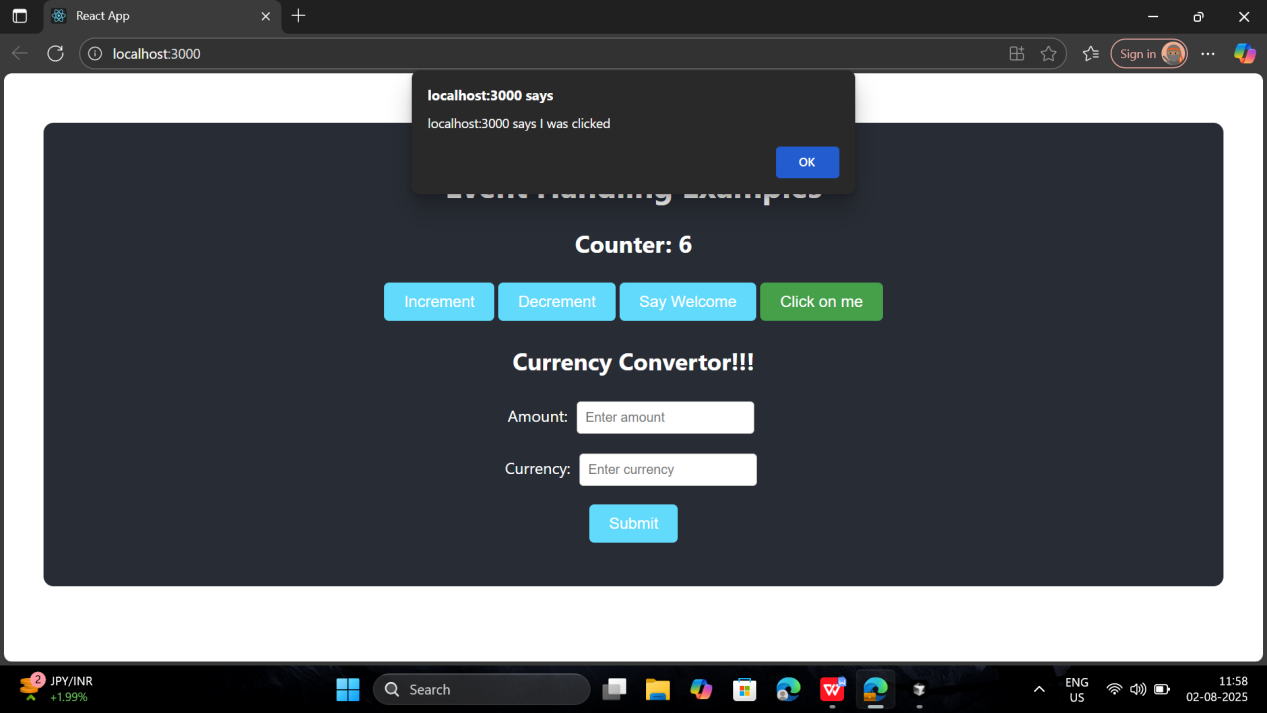
1. Create a button “Say Welcome” which invokes the function which takes “welcome” as an argument.

**Output:**



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

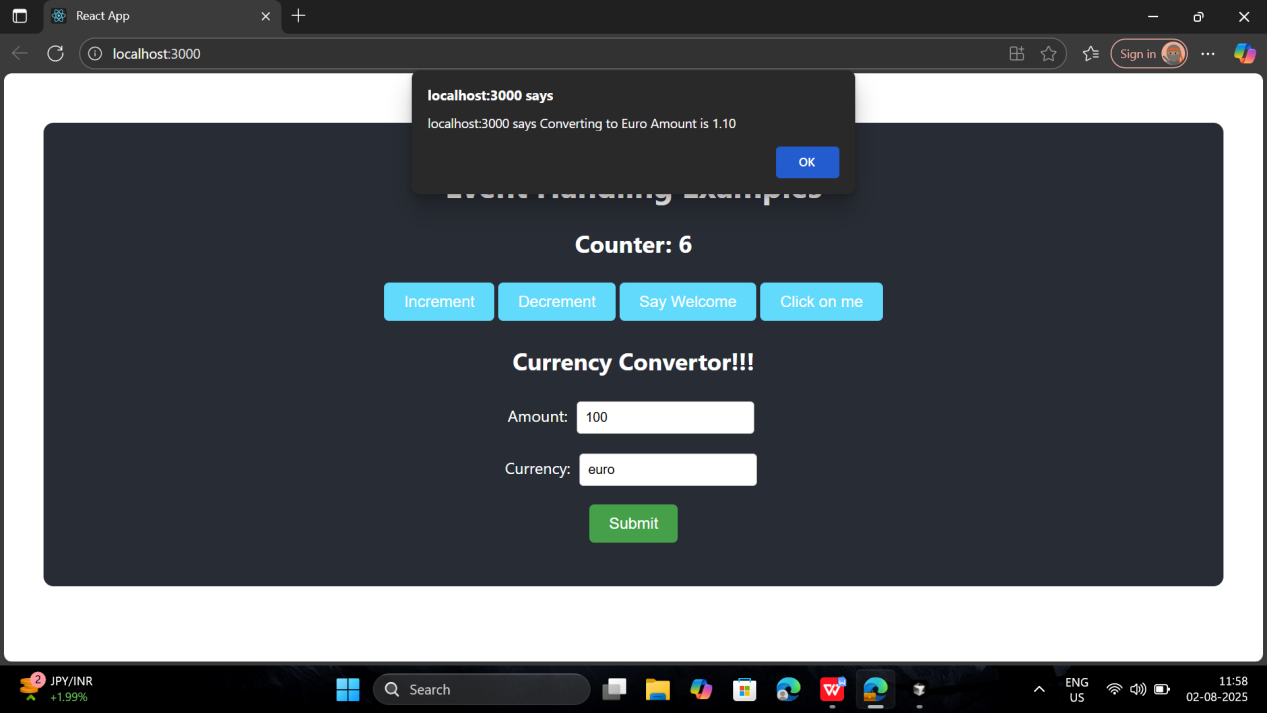
**Output:**



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.

**Output:**



1. **ReactJS-HOL**

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

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.

Code:

import React, { useState } from 'react';

import './App.css';

function **LoginButton**(*props*) {

  return (

    <button *onClick*={*props*.onClick}>

      Login

    </button>

  );

}

function **LogoutButton**(*props*) {

  return (

    <button *onClick*={*props*.onClick}>

      Logout

    </button>

  );

}

function **UserGreeting**() {

  return <h1>Welcome back</h1>;

}

function **GuestGreeting**() {

  return <h1>Please sign up.</h1>;

}

function **Greeting**(*props*) {

  const isLoggedIn = *props*.isLoggedIn;

  if (isLoggedIn) {

    return <UserGreeting />;

  }

  return <GuestGreeting />;

}

function **App**() {

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

  const **handleLoginClick** = () => {

    setIsLoggedIn(true);

  };

  const **handleLogoutClick** = () => {

    setIsLoggedIn(false);

  };

  return (

    <div *className*="App">

      <Greeting *isLoggedIn*={isLoggedIn} />

      {isLoggedIn ? (

        <LogoutButton *onClick*={handleLogoutClick} />

      ) : (

        <LoginButton *onClick*={handleLoginClick} />

      )}

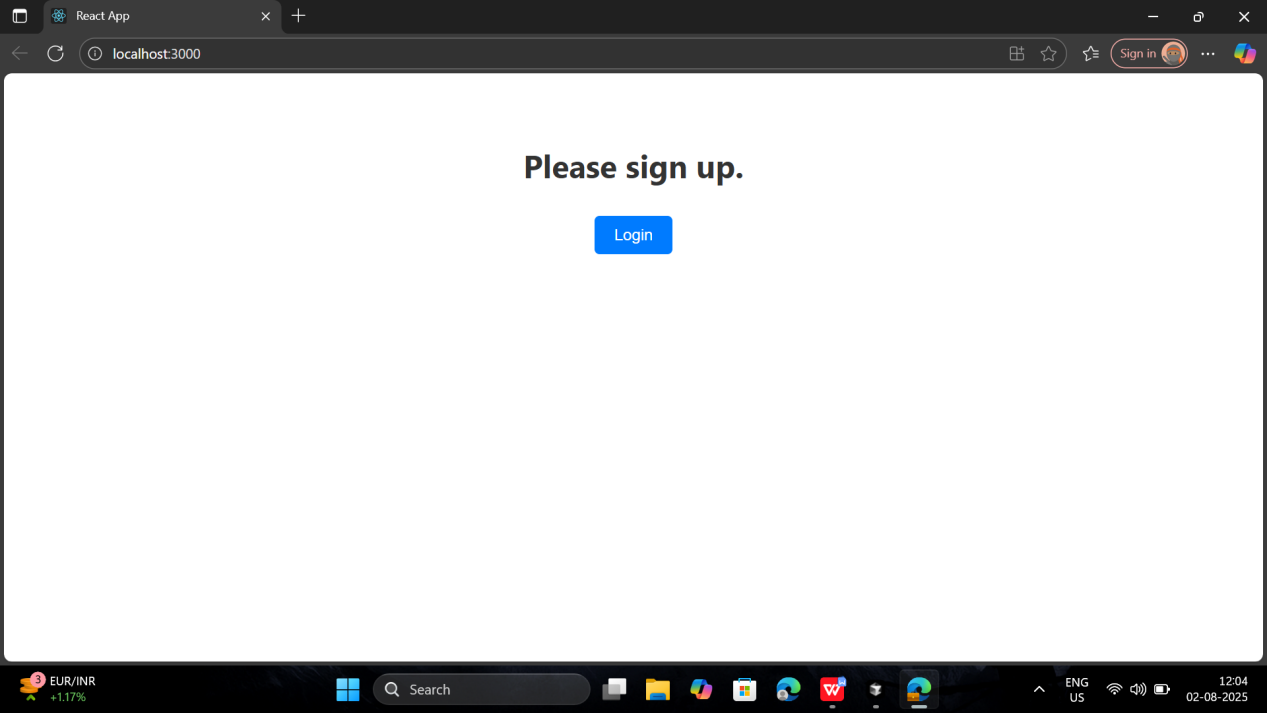
    </div>

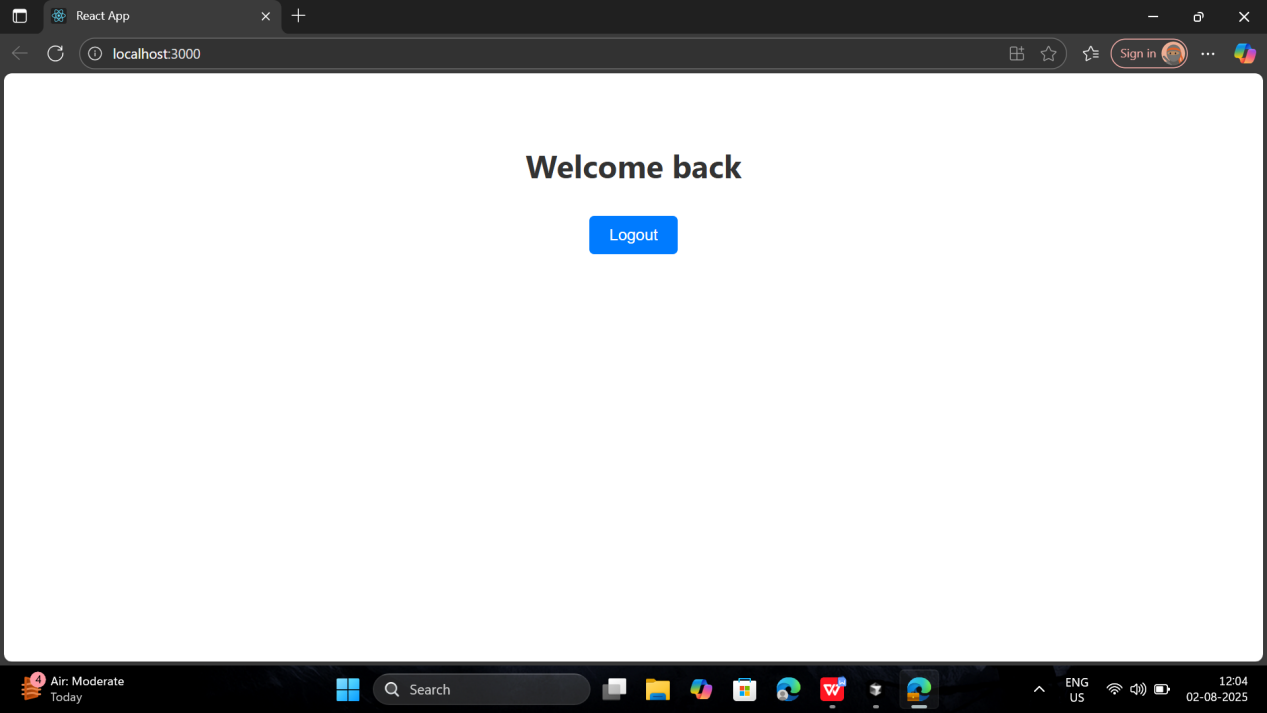
  );

}

export default App;

**Output:**





1. **ReactJS-HOL**

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.

Code:

import React from 'react';

import './App.css';

import BookDetails from './BookDetails';

import BlogDetails from './BlogDetails';

import CourseDetails from './CourseDetails';

import { books, blogs, courses } from './data';

function App() {

  return (

    <div *className*="App">

      <div>

        <BookDetails *books*={books} />

        <BlogDetails *blogs*={blogs} />

        <CourseDetails *courses*={courses} />

      </div>

    </div>

  );

}

export default App;

**Output:**

