**WEEK 7**

**REACT**

**ReactJS HOL-9**

**CODE:**

**App.js**

import React from 'react';

import ListOfPlayers from './ListOfPlayers';

import { OddPlayers, EvenPlayers, ListofIndianPlayers, IndianTeam } from './IndianPlayers';

function App() {

  const playersData = [

    { 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 }

  ];

  const oddEvenPlayers = ['Sachin1', 'Dhoni2', 'Virat3', 'Rohit4', 'Yuvraj5', 'Raina6']; // Example data for odd/even players

  // Use a flag variable for conditional rendering

  const flag = true; // Set to true or false to see different outputs [cite: 28, 30, 31, 32]

  return (

    <div className="App">

      {flag ? ( // Conditional rendering using if-else

        <div>

          <ListOfPlayers players={playersData} />

          <hr />

          {/\* The filtering for scores below 70 is handled within ListOfPlayers component \*/}

        </div>

      ) : (

        <div>

          <h1>Indian Team</h1>

          <OddPlayers players={oddEvenPlayers} />

          <hr />

          <EvenPlayers players={oddEvenPlayers} />

          <hr />

          <ListofIndianPlayers IndianPlayers={IndianTeam} />

        </div>

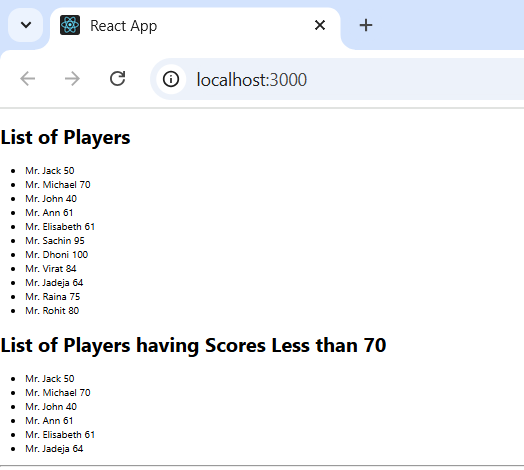
      )}

    </div>

  );

}

export default App;

**OUTPUT:**

**ReactJS HOL-10**

**CODE:**

**App.js**

import React from 'react';

import officeSpaceImage from './office-space.jpeg'; // Make sure you have an image file (e.g., office-space.jpeg) in src/

function App() {

  // Create an element to display the heading of the page. [cite: 117]

  const element = "Office Space";

  // Attribute to display the image of the office space

  const jsxatt = <img src={officeSpaceImage} width="25%" height="25%" alt="Office Space" />; // Image imported and used as attribute

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

  const officeDetails = {

    Name: "DBS",

    Rent: 50000,

    Address: "Chennai"

  };

  // Create a list of Object and loop through the office space item to display more data.

  const offices = [

    { Name: "Tech Hub", Rent: 75000, Address: "Bengaluru" },

    { Name: "Creative Spaces", Rent: 45000, Address: "Hyderabad" },

    { Name: "Work Oasis", Rent: 62000, Address: "Mumbai" },

  ];

  // Function to determine text color based on rent

  const getRentColor = (rent) => {

    let colors = [];

    if (rent <= 60000) {

      colors.push('red'); // Display the color of the Rent in Red if it’s below 60000 [cite: 121]

    } else {

      colors.push('green'); // and in Green if it’s above 60000. [cite: 121]

    }

    return colors[0]; // Return the color string

  };

  return (

    <div className="App" style={{ textAlign: 'center', fontFamily: 'Arial, sans-serif' }}>

      <h1>{element} , at Affordable Range</h1> {/\* Use JavaScript expressions in JSX [cite: 104] \*/}

      {jsxatt} {/\* Render JSX attribute/element  \*/}

      {/\* Displaying details of a single office object  \*/}

      <div>

        <h3>Name: {officeDetails.Name}</h3>

        <h3 style={{ color: getRentColor(officeDetails.Rent) }}>Rent: Rs. {officeDetails.Rent}</h3> {/\* Inline CSS in JSX  \*/}

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

      </div>

      <hr />

      <h2>More Office Spaces:</h2>

      {/\* Looping through a list of objects  \*/}

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

        <div key={index} style={{ marginBottom: '20px', border: '1px solid #eee', padding: '10px', borderRadius: '5px' }}>

          <h3>Name: {office.Name}</h3>

          <h3 style={{ color: getRentColor(office.Rent) }}>Rent: Rs. {office.Rent}</h3> {/\* Inline CSS  \*/}

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

        </div>

      ))}

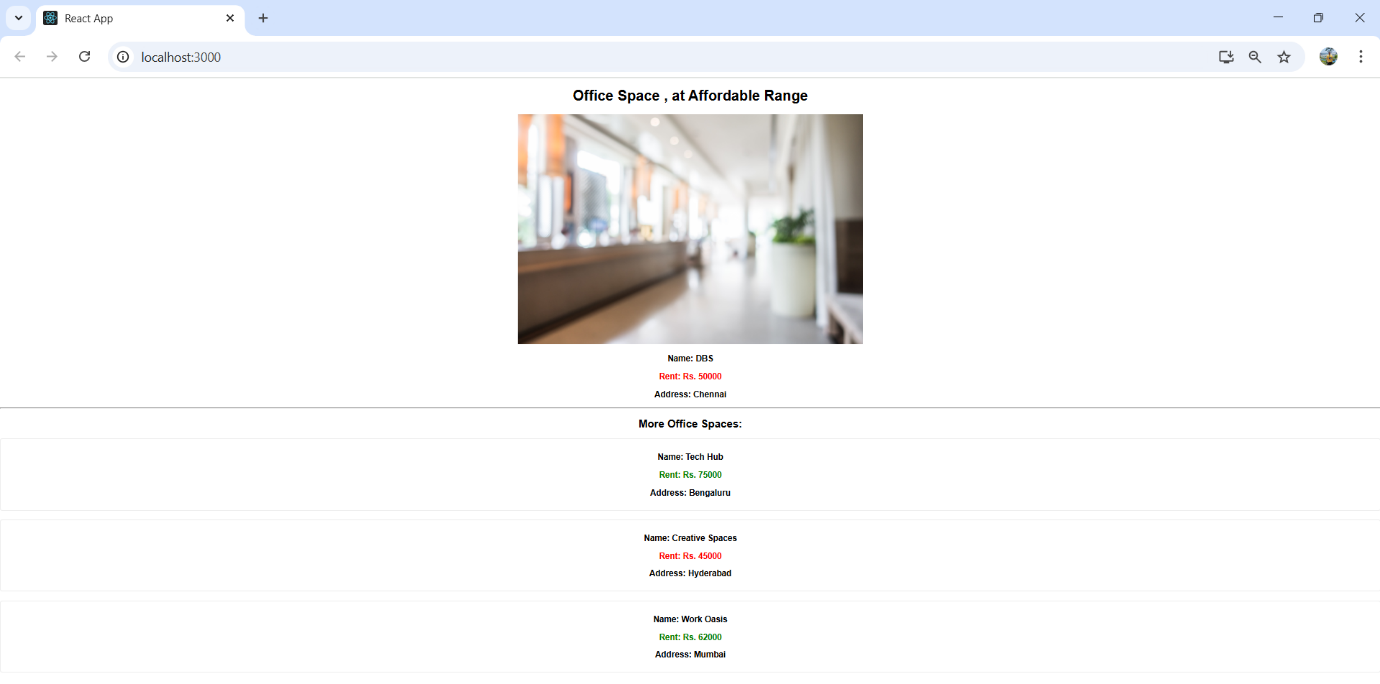
    </div>

  );

}

export default App;

**OUTPUT:**

****

**ReactJS HOL-11**

**CODE:**

**App.js**

import React, { useState } from 'react';

// CurrencyConvertor component [cite: 78]

function CurrencyConvertor() {

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

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

  const handleRupeesChange = (event) => {

    const value = event.target.value;

    setRupees(value);

    // Convert to Euro on change, or you can do it on button click as per requirements

    // For simplicity, we'll convert on button click as specified.

  };

  const handleSubmit = (event) => { // Handle the Click event of the button to invoke the handleSubmit event

    event.preventDefault(); // Prevent default form submission behavior

    const conversionRate = 0.011; // Example conversion rate (1 INR = 0.011 EUR approx)

    const convertedEuro = parseFloat(rupees) \* conversionRate;

    setEuro(convertedEuro.toFixed(2)); // Display with 2 decimal places

  };

  return (

    <div style={{ marginTop: '30px', border: '1px solid #ccc', padding: '20px', borderRadius: '8px' }}>

      <h2>Currency Convertor (INR to Euro)</h2>

      <form onSubmit={handleSubmit}>

        <label>

          Indian Rupees:

          <input

            type="number"

            value={rupees}

            onChange={handleRupeesChange}

            placeholder="Enter amount in INR"

          />

        </label>

        <br /><br />

        <button type="submit">Convert to Euro</button>

      </form>

      {euro && <h3>Converted Euro: €{euro}</h3>}

    </div>

  );

}

class App extends React.Component {

  constructor(props) {

    super(props);

    this.state = {

      counter: 0,

    };

    // Bind 'this' for methods if not using arrow functions in class methods

    this.incrementCounter = this.incrementCounter.bind(this); // Use this keyword

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

  }

  // Method to increment the value

  incrementCounter() {

    this.setState(prevState => ({

      counter: prevState.counter + 1,

    }));

  }

  // Method to say hello followed by a static message

  sayHello() {

    console.log("Hello, this is a static message!");

    alert("Hello, this is a static message!");

  }

  // Method to increment value and say hello (multiple methods invoked)

  handleIncreaseAndSayHello = () => { // Using arrow function to auto-bind 'this'

    this.incrementCounter();

    this.sayHello();

  }

  // Method that takes an argument

  sayWelcome(message) {

    console.log(message);

    alert(message);

  }

  // Synthetic event example

  handleSyntheticEvent = (event) => { // Define Synthetic event [cite: 58]

    console.log("I was clicked!", event); // Synthetic event

    alert("I was clicked!");

  };

  render() {

    return (

      <div className="App" style={{ textAlign: 'center', marginTop: '50px' }}>

        <h1>Event Handling Examples</h1>

        {/\* Increment and Decrement Buttons  \*/}

        <div>

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

          <button onClick={this.incrementCounter}>Increment</button> {' '}

          <button onClick={() => this.setState(prevState => ({ counter: prevState.counter - 1 }))}>Decrement</button>

        </div>

        <hr />

        {/\* Increase button invoking multiple methods  \*/}

        <div>

          <h2>Multiple Method Invocation</h2>

          <button onClick={this.handleIncreaseAndSayHello}>Increase (and Say Hello)</button>

        </div>

        <hr />

        {/\* Button with argument  \*/}

        <div>

          <h2>Function with Argument</h2>

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

        </div>

        <hr />

        {/\* Synthetic Event  \*/}

        <div>

          <h2>Synthetic Event (onClick)</h2>

          {/\* Note: onPress is not a standard DOM event; onClick is the equivalent in React. \*/}

          <button onClick={this.handleSyntheticEvent}>On Press (Click)</button>

        </div>

        <hr />

        {/\* Currency Convertor Component [cite: 78] \*/}

        <CurrencyConvertor />

      </div>

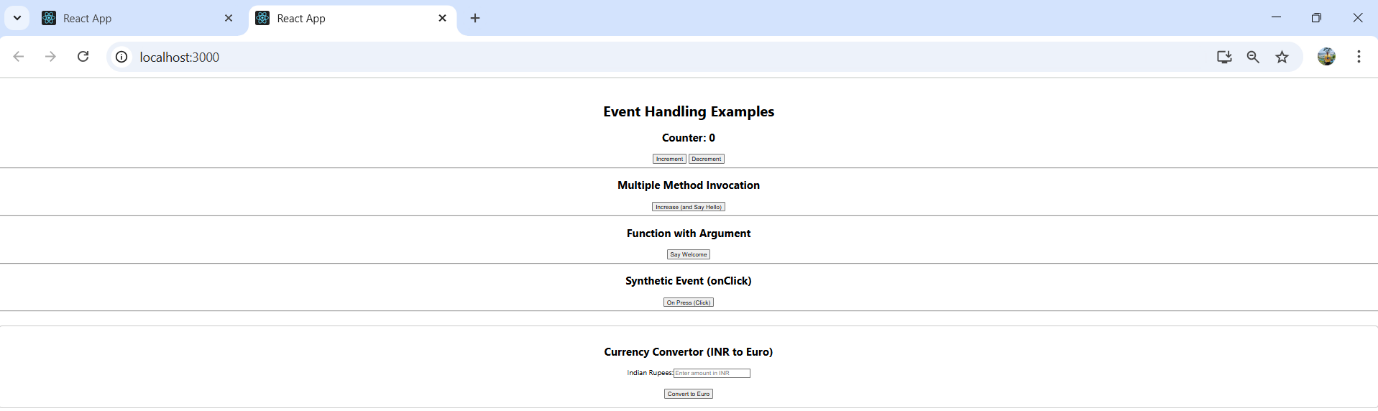
    );

  }

}

export default App;

**OUTPUT:**

****

**ReactJS HOL-12**

**CODE:**

**App.js**

import React, { useState } from 'react';

import Greeting from './Greeting';

import LoginButton from './LoginButton';

import LogoutButton from './LogoutButton';

function App() {

  const [isLoggedIn, setIsLoggedIn] = useState(false); // State to manage login status

  const handleLoginClick = () => { // Login button displays different page

    setIsLoggedIn(true); // Once logged in, User page displayed

  };

  const handleLogoutClick = () => { // Logout button displays different page

    setIsLoggedIn(false); // When user clicks logout, Guest page displayed

  };

  let button;

  if (isLoggedIn) {

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

  } else {

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

  }

  return (

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

      <Greeting isLoggedIn={isLoggedIn} />

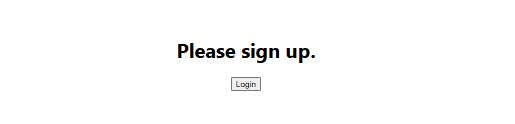
      {button}

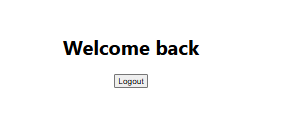
    </div>

  );

}

export default App;

**OUTPUT:**



**ReactJS HOL-13**

**CODE:**

**App.js**

**import React from 'react';**

**import BookDetails, { books } from './BookDetails'; // Import books data**

**import BlogDetails from './BlogDetails';**

**import CourseDetails from './CourseDetails';**

**import './App.css'; // Assuming some basic CSS for layout as per the image**

**function App() {**

**return (**

**<div className="App">**

**<div style={{ display: 'flex', justifyContent: 'space-around', border: '1px solid black', padding: '20px' }}>**

**<CourseDetails />**

**<div style={{ borderLeft: '2px solid green', height: 'auto', margin: '0 20px' }}></div> {/\* Visual separator \*/}**

**<BookDetails books={books} /> {/\* Pass books data to BookDetails \*/}**

**<div style={{ borderLeft: '2px solid green', height: 'auto', margin: '0 20px' }}></div> {/\* Visual separator \*/}**

**<BlogDetails />**

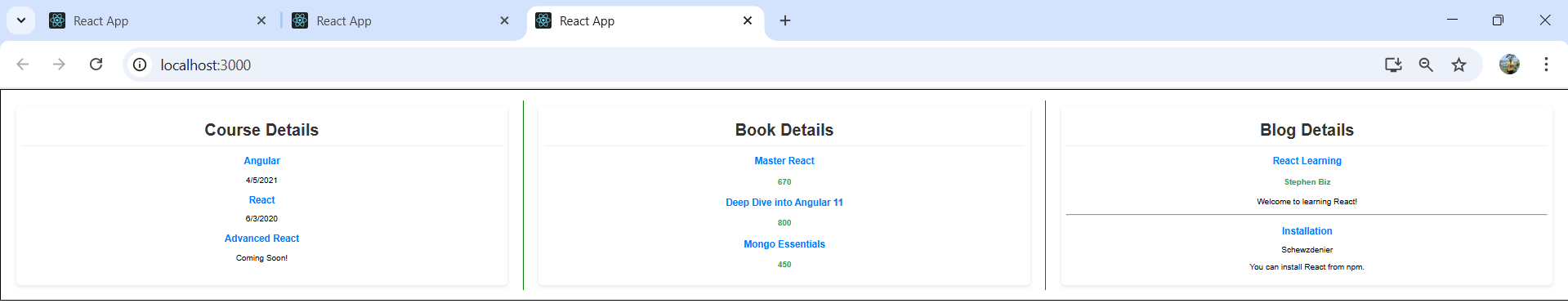
**</div>**

**</div>**

**);**

**}**

**export default App;**

**OUTPUT:**