## **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:**



**Solution:**

**Code:**

**ListofPlayers.js :**

import React from 'react';

const ListofPlayers = () => {

  const players = [

    { name: 'Virat', score: 85 },

    { name: 'Rohit', score: 90 },

    { name: 'Dhoni', score: 65 },

    { name: 'Pant', score: 45 },

    { name: 'Hardik', score: 75 },

    { name: 'Jadeja', score: 68 },

    { name: 'Kohli', score: 95 },

    { name: 'Rahul', score: 58 },

    { name: 'Bumrah', score: 88 },

    { name: 'Gill', score: 92 },

    { name: 'Shami', score: 55 }

  ];

  const lowScorers = players.filter(player => player.score < 70);

  return (

    <div>

      <h2>All Players</h2>

      <ul>

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

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

        ))}

      </ul>

      <h3>Players with Score below 70</h3>

      <ul>

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

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

        ))}

      </ul>

    </div>

  );

};

export default ListofPlayers;

**IndianPlayers.js:**

import React from 'react';

const IndianPlayers = () => {

  const players = ['Virat', 'Rohit', 'Dhoni', 'Pant', 'Hardik', 'Jadeja', 'Kohli', 'Rahul', 'Bumrah', 'Gill', 'Shami'];

  const oddTeam = players.filter((\_, index) => index % 2 !== 0);

  const evenTeam = players.filter((\_, index) => index % 2 === 0);

  const T20Players = ['Surya', 'Iyer', 'Ashwin'];

  const RanjiTrophyPlayers = ['Pujara', 'Saha', 'Karun'];

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

  return (

    <div>

      <h2>Indian Players</h2>

      <h3>Odd Team</h3>

      <ul>

        {oddTeam.map((player, index) => <li key={index}>{player}</li>)}

      </ul>

      <h3>Even Team</h3>

      <ul>

        {evenTeam.map((player, index) => <li key={index}>{player}</li>)}

      </ul>

      <h3>All Combined Players (T20 + Ranji Trophy)</h3>

      <ul>

        {allPlayers.map((player, index) => <li key={index}>{player}</li>)}

      </ul>

    </div>

  );

};

export default IndianPlayers;

**App.js:**

import React from 'react';

import ListofPlayers from './ListofPlayers';

import IndianPlayers from './IndianPlayers';

function App() {

  const flag = true;

  return (

    <div className="App">

      <h1>Welcome to Cricket App</h1>

      {flag ? <ListofPlayers /> : <IndianPlayers />}

    </div>

  );

}

export default App;

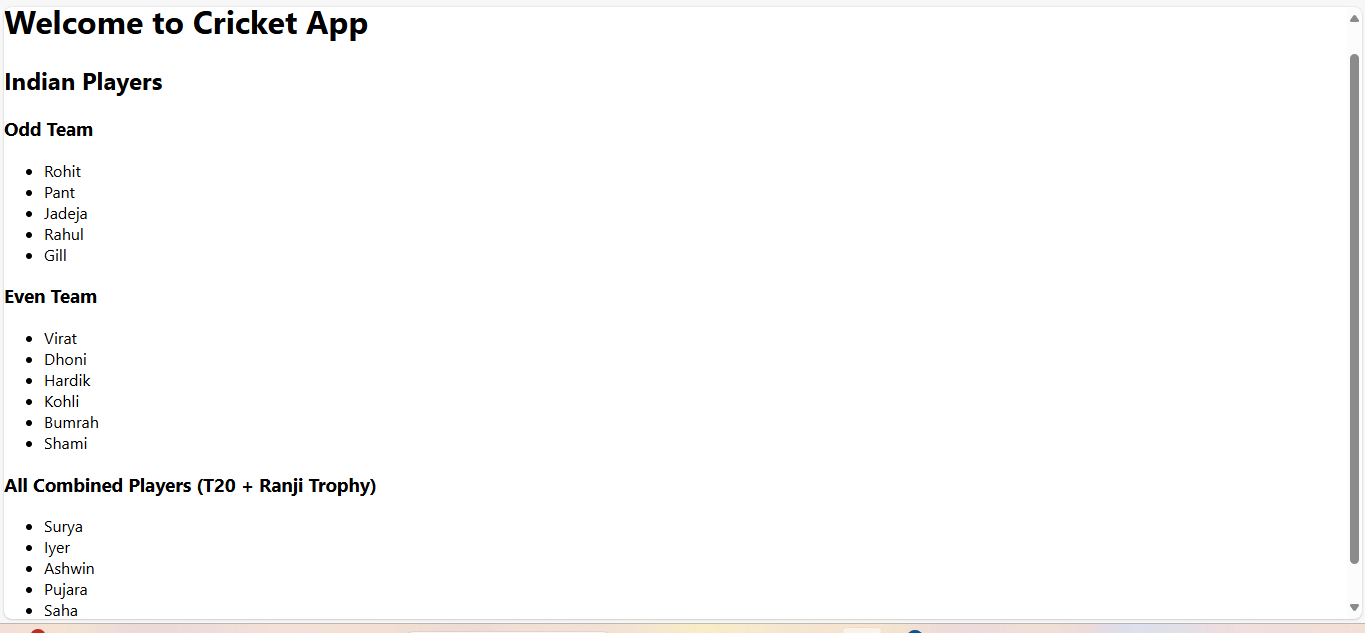
**when Flag = true;**

**Output:**



**When flag = false:**

**Output:**



## **10,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 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.

Output:



**Hint:**





**Solution:**

**Code:**

**App.js:**

import React from 'react';

function App() {

  const heading = <h1>Office Space Rental Application</h1>;

  const officeImageUrl = "https://via.placeholder.com/400x200?text=Office+Image";

  const office = {

    name: "TechSpace Solutions",

    rent: 55000,

    address: "2nd Floor, Innovate Tower, Bangalore"

  };

  const officeList = [

    {

      name: "Elite Office Hub",

      rent: 45000,

      address: "3rd Floor, Business Park, Chennai"

    },

    {

      name: "Urban Workbay",

      rent: 75000,

      address: "5th Floor, Cyber Heights, Hyderabad"

    },

    {

      name: "NextGen Offices",

      rent: 62000,

      address: "1st Floor, Skyline Plaza, Pune"

    },

    {

      name: "Budget Business Suites",

      rent: 39000,

      address: "Near Tech Junction, Kochi"

    }

  ];

  const getRentColor = (rent) => {

    return {

      color: rent > 60000 ? 'green' : 'red'

    };

  };

  return (

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

      {heading}

      <img src={officeImageUrl} alt="Office" style={{ width: '100%', maxWidth: '600px', marginBottom: '20px' }} />

      <h2>Featured Office:</h2>

      <div>

        <p><strong>Name:</strong> {office.name}</p>

        <p><strong>Rent:</strong> <span style={getRentColor(office.rent)}>{office.rent}</span></p>

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

      </div>

      <hr />

      <h2>Other Available Offices</h2>

      {officeList.map((o, index) => (

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

          <p><strong>Name:</strong> {o.name}</p>

          <p><strong>Rent:</strong> <span style={getRentColor(o.rent)}>{o.rent}</span></p>

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

        </div>

      ))}

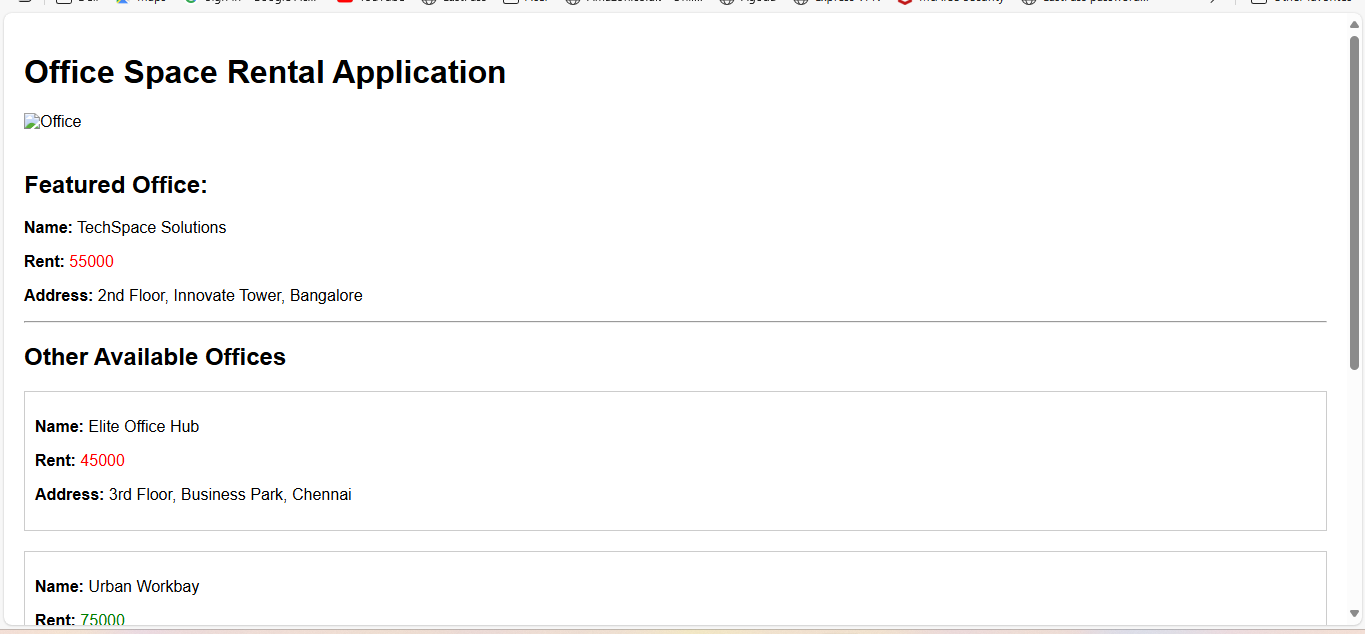
    </div>

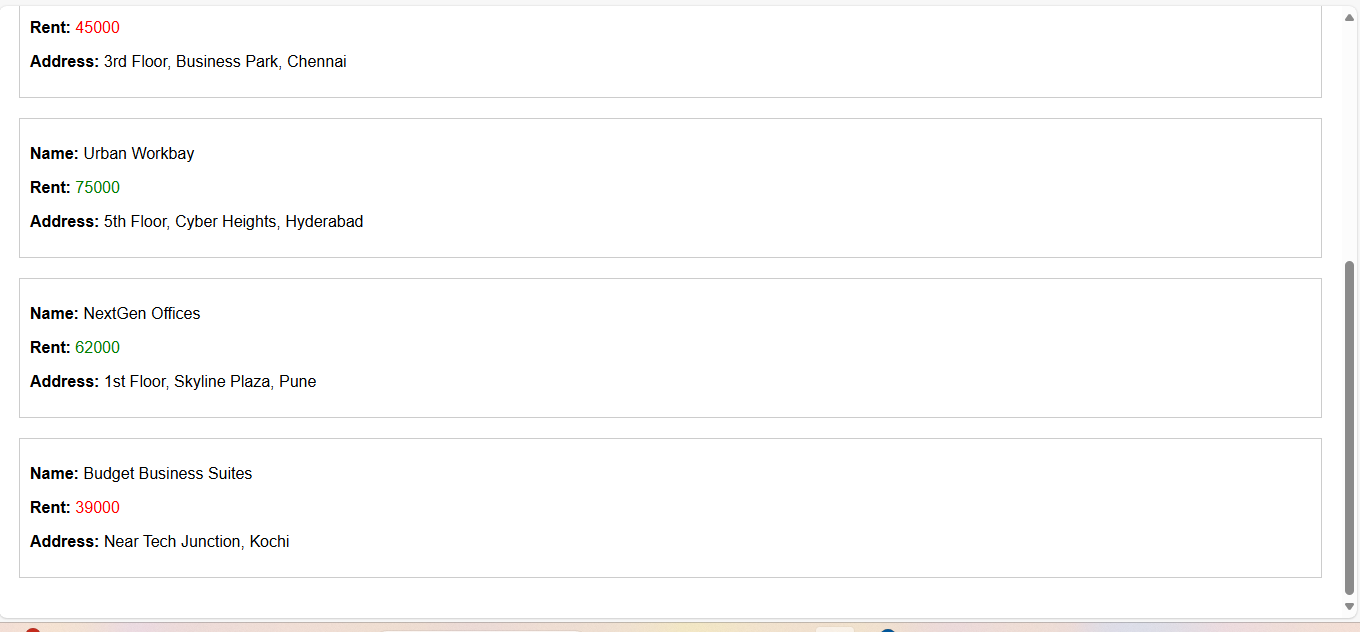
  );

}

export default App;

**Output:**





## **11,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.



**Solution:**

**Code:**

**App.js:**

import React, { Component } from 'react';

import CurrencyConverter from './CurrencyConverter';

class App extends Component {

  constructor(props) {

    super(props);

    this.state = {

      counter: 0,

    };

    // Binding methods to use "this"

    this.increment = this.increment.bind(this);

    this.sayHello = this.sayHello.bind(this);

    this.combinedFunction = this.combinedFunction.bind(this);

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

  }

  increment() {

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

  }

  decrement = () => {

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

  }

  sayHello() {

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

  }

  combinedFunction() {

    this.increment();

    this.sayHello();

  }

  sayWelcome(message) {

    alert("Message: " + message);

  }

  handleSyntheticEvent = (e) => {

    e.preventDefault();

    alert("I was clicked (Synthetic Event)");

  };

  render() {

    return (

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

        <h1>Event Examples in React</h1>

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

        <button onClick={this.combinedFunction}>Increase</button>

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

        <br /><br />

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

        <br /><br />

        <button onClick={this.handleSyntheticEvent}>Click Me (Synthetic Event)</button>

        <br /><br />

        <CurrencyConverter />

      </div>

    );

  }

}

export default App;

**CurrencyConverter.js:**

import React, { useState } from 'react';

const CurrencyConverter = () => {

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

const [convertedValue, setConvertedValue] = useState(null);

const [currency, setCurrency] = useState('EUR');

const exchangeRates = {

EUR: 89.5, // 1 Euro = 89.5 INR (example rate)

USD: 83.2, // 1 Dollar = 83.2 INR

GBP: 105.3 // 1 Pound = 105.3 INR

};

const handleSubmit = (e) => {

e.preventDefault();

const rupeeValue = parseFloat(rupees);

if (!isNaN(rupeeValue)) {

const converted = rupeeValue / exchangeRates[currency];

setConvertedValue(converted.toFixed(2));

} else {

setConvertedValue(null);

alert("Please enter a valid number");

}

};

return (

<div style={{ marginTop: '30px' }}>

<h2>Currency Converter (INR ➡️ Foreign Currency)</h2>

<form onSubmit={handleSubmit}>

<input

type="text"

placeholder="Enter INR amount"

value={rupees}

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

style={{ marginRight: '10px' }}

/>

<select

value={currency}

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

style={{ marginRight: '10px' }}

>

<option value="EUR">Euro (EUR)</option>

<option value="USD">Dollar (USD)</option>

<option value="GBP">Pound (GBP)</option>

</select>

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

</form>

{convertedValue !== null && (

<p>

Equivalent in {currency}: <strong>{convertedValue} {currency}</strong>

</p>

)}

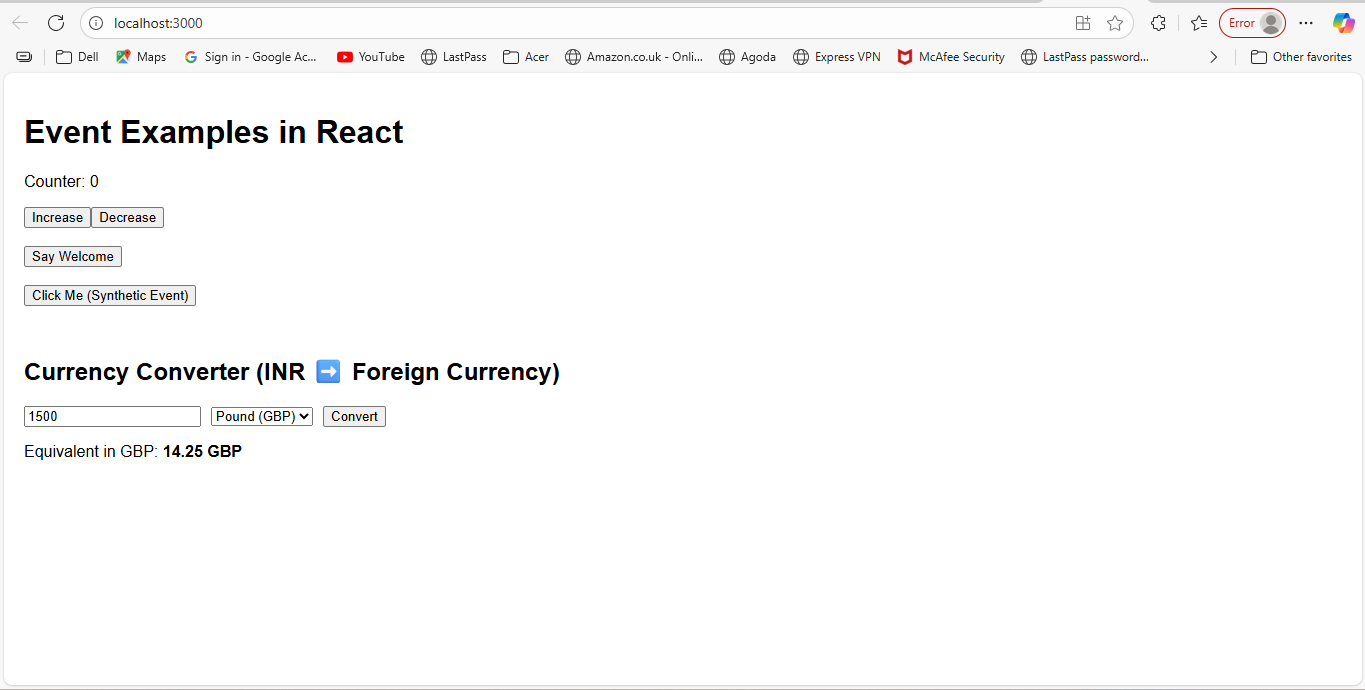
</div>

);

};

export default CurrencyConverter;

**Output:**



## **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

## **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 “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.





**Hint:**







**Solution:;**

**Code:**

**App.js:**

import React, { useState } from 'react';

import SignInPage from './SignInPage';

import UserPage from './UserPage';

import LogoutPage from './LogoutPage';

function App() {

  const [page, setPage] = useState('signin'); // 'signin', 'user', 'logout'

  const handleLogin = () => {

    setPage('user');

  };

  const handleLogout = () => {

    setPage('logout');

  };

  const handleGoHome = () => {

    setPage('signin');

  };

  let content;

  if (page === 'signin') {

    content = <SignInPage onLogin={handleLogin} />;

  } else if (page === 'user') {

    content = <UserPage onLogout={handleLogout} />;

  } else if (page === 'logout') {

    content = <LogoutPage onGoHome={handleGoHome} />;

  }

  return (

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

      <h1>Ticket Booking App</h1>

      {content}

    </div>

  );

}

export default App;

**SigninPage.js:**

import React, { useState } from 'react';

const SignInPage = ({ onLogin }) => {

  const [username, setUsername] = useState('');

  const [password, setPassword] = useState('');

  const handleSubmit = (e) => {

    e.preventDefault();

    // Simulate basic login (no real auth here)

    if (username === 'user' && password === '1234') {

      onLogin();

    } else {

      alert("Invalid credentials");

    }

  };

  return (

    <div>

      <h2>Sign In</h2>

      <form onSubmit={handleSubmit}>

        <div>

          <label>Username: </label>

          <input type="text" value={username} onChange={(e) => setUsername(e.target.value)} required />

        </div>

        <br />

        <div>

          <label>Password: </label>

          <input type="password" value={password} onChange={(e) => setPassword(e.target.value)} required />

        </div>

        <br />

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

      </form>

    </div>

  );

};

export default SignInPage;

**UserPage.js:**

import React from 'react';

const UserPage = ({ onLogout }) => {

  return (

    <div>

      <h2>Welcome Back, User!</h2>

      <p>You can now book tickets:</p>

      <form>

        <label>

          Select Flight:

          <select>

            <option>Flight A - ₹5000</option>

            <option>Flight B - ₹6000</option>

            <option>Flight C - ₹5500</option>

          </select>

        </label>

        <br /><br />

        <button type="submit">Book Now</button>

      </form>

      <br />

      <button onClick={onLogout}>Logout</button>

    </div>

  );

};

export default UserPage;

**LogoutPage.js:**

import React from 'react';

const LogoutPage = ({ onGoHome }) => {

  return (

    <div>

      <h2>You have been logged out successfully.</h2>

      <button onClick={onGoHome}>Go to Sign In Page</button>

    </div>

  );

};

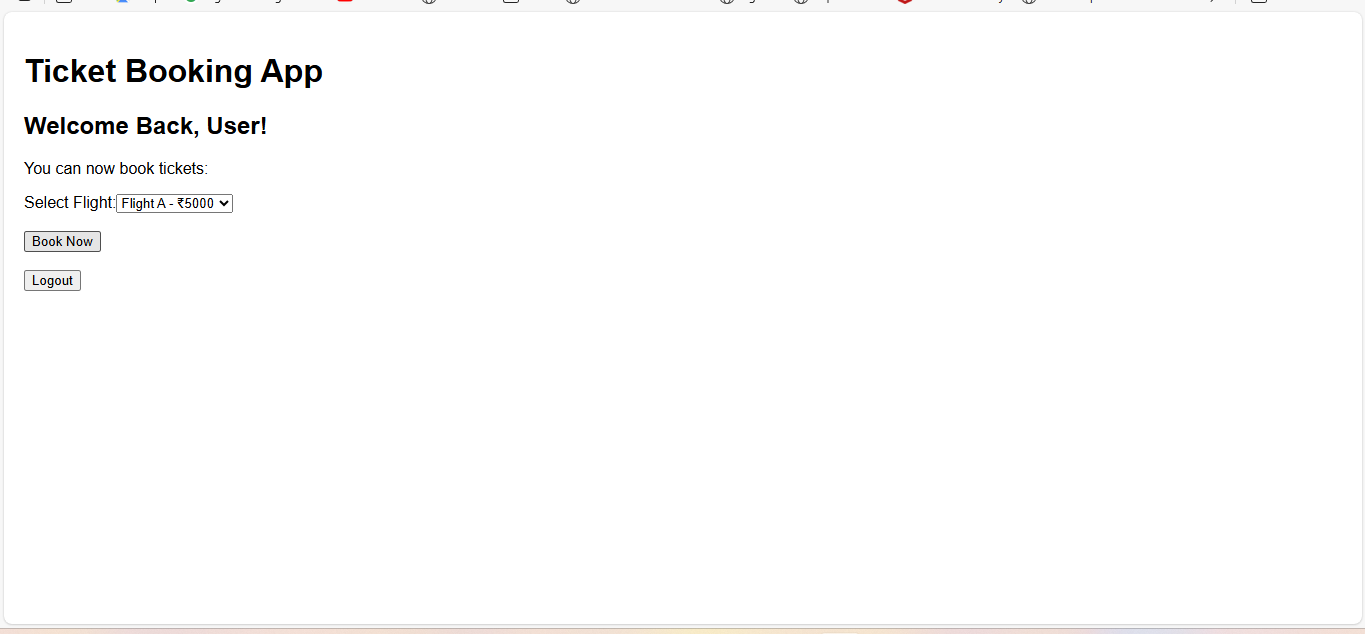
export default LogoutPage;

**Output:**

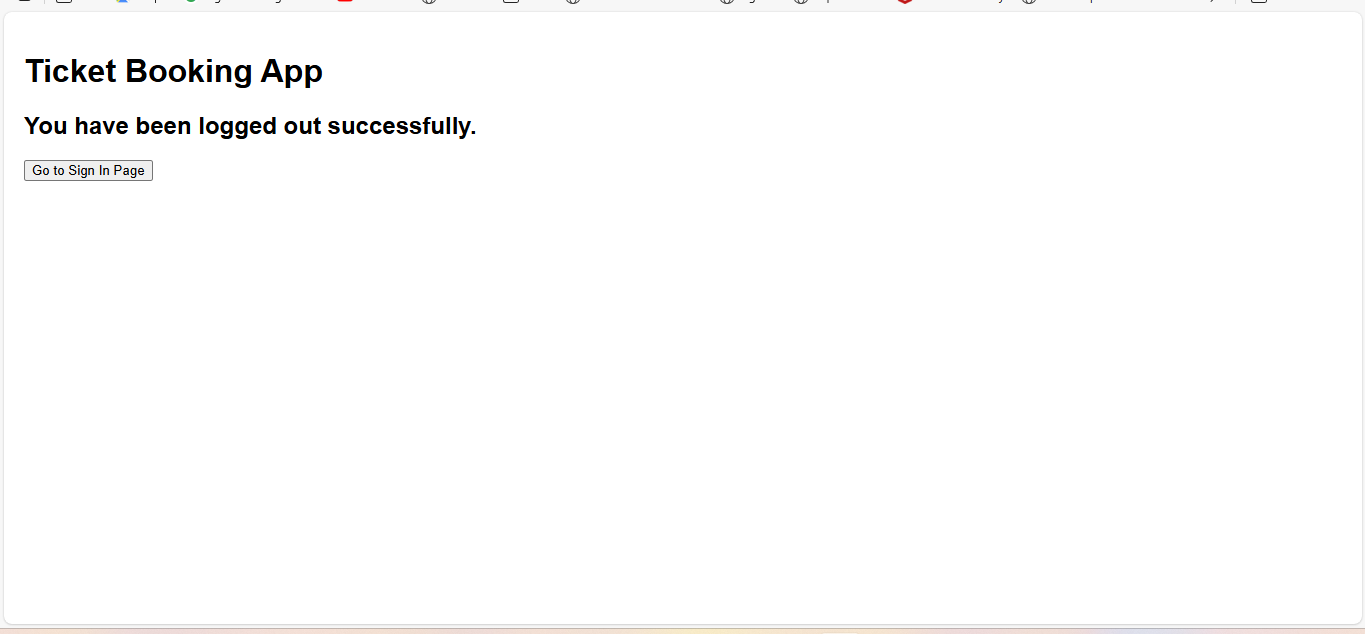
**SigninPage:**

****

**TicketBookingPage:**

****

**Logoutpage:**

****

## **13,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

## **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 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.



**Hint:**







**Solution:**

**Code:**

**App.js:**

import React, { useState } from 'react';

import BookDetails from './BookDetails';

import BlogDetails from './BlogDetails';

import CourseDetails from './CourseDetails';

function App() {

const [showComponent, setShowComponent] = useState('books'); // Conditional rendering flag

const books = [

{ title: 'Atomic Habits', author: 'James Clear' },

{ title: 'The Alchemist', author: 'Paulo Coelho' }

];

const blogs = [

{ title: 'React Tips', content: 'Use hooks and keep components small.' },

{ title: 'State vs Props', content: 'Props are immutable, state is local.' }

];

const courses = [

{ name: 'React Fundamentals', duration: '3 Weeks' },

{ name: 'Advanced JavaScript', duration: '4 Weeks' }

];

return (

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

<h1>Blogger App Dashboard</h1>

{/\* Conditional Buttons \*/}

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

<button onClick={() => setShowComponent('books')}>Show Book Details</button>

<button onClick={() => setShowComponent('blogs')}>Show Blog Details</button>

<button onClick={() => setShowComponent('courses')}>Show Course Details</button>

</div>

{/\* Conditional Rendering Approaches \*/}

{/\* 1. if-else rendering \*/}

{showComponent === 'books' && <BookDetails books={books} />}

{/\* 2. Ternary operator \*/}

{showComponent === 'blogs'

? <BlogDetails blogs={blogs} />

: showComponent === 'courses' && <CourseDetails courses={courses} />

}

{/\* 3. Switch-like rendering with element variable (alternative way) \*/}

{/\* This method isn't needed if using above methods but shown here for demo \*/}

{/\* let component; if (showComponent === 'books') component = <BookDetails ... /> \*/}

</div>

);

}

export default App;

**CourseDetails.js**

import React from 'react';

const CourseDetails = ({ courses }) => {

return (

<div>

<h2>Course Details</h2>

{courses?.length ? (

<ul>

{courses.map((course, index) => (

<li key={index}>

<strong>{course.name}</strong> - {course.duration}

</li>

))}

</ul>

) : (

<p>No courses listed.</p>

)}

</div>

);

};

export default CourseDetails;

**BlogDetails.js:**

import React from 'react';

const BlogDetails = ({ blogs }) => {

return (

<div>

<h2>📝 Blog Posts</h2>

{blogs.length > 0 && (

<div>

{blogs.map((blog, index) => (

<div key={index}>

<h4>{blog.title}</h4>

<p>{blog.content}</p>

</div>

))}

</div>

)}

{blogs.length === 0 && <p>No blog posts found.</p>}

</div>

);

};

export default BlogDetails;

**BookDetails.js**

import React from 'react';

const BookDetails = ({ books }) => {

return (

<div>

<h2> Book Details</h2>

{books.length === 0 ? (

<p>No books available</p>

) : (

<ul>

{books.map((book, index) => (

<li key={index}>

<strong>{book.title}</strong> by {book.author}

</li>

))}

</ul>

)}

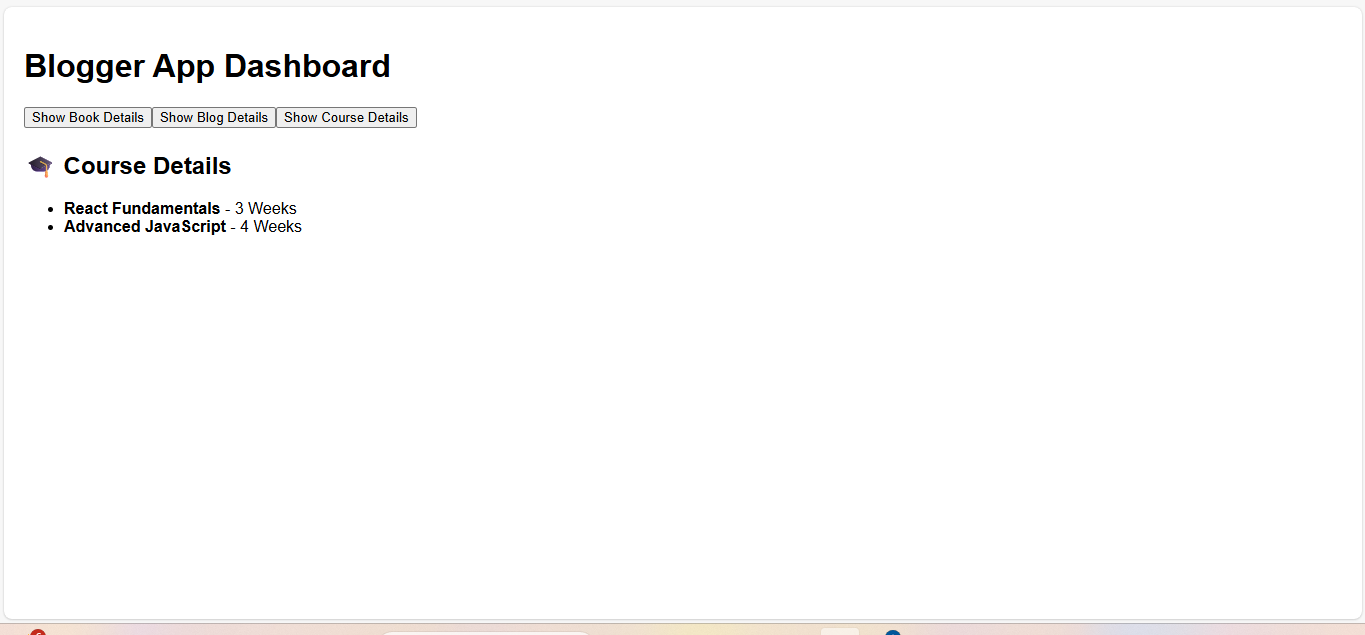
</div>

);

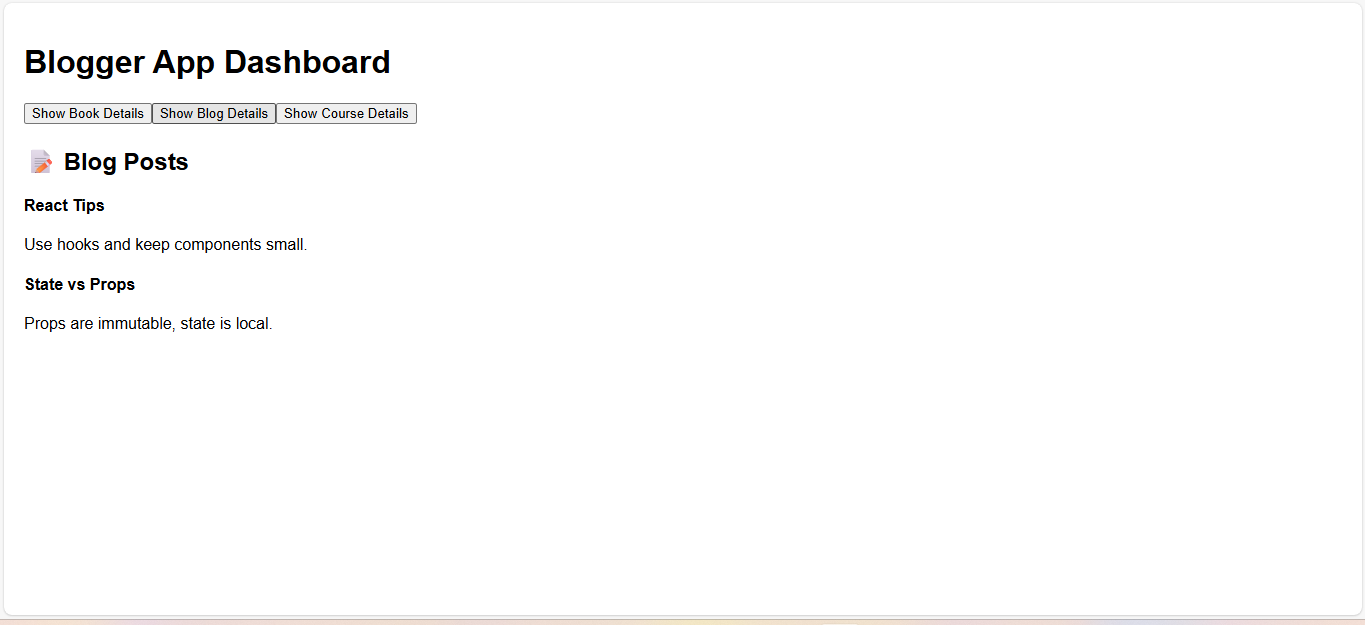
};

export default BookDetails**;**

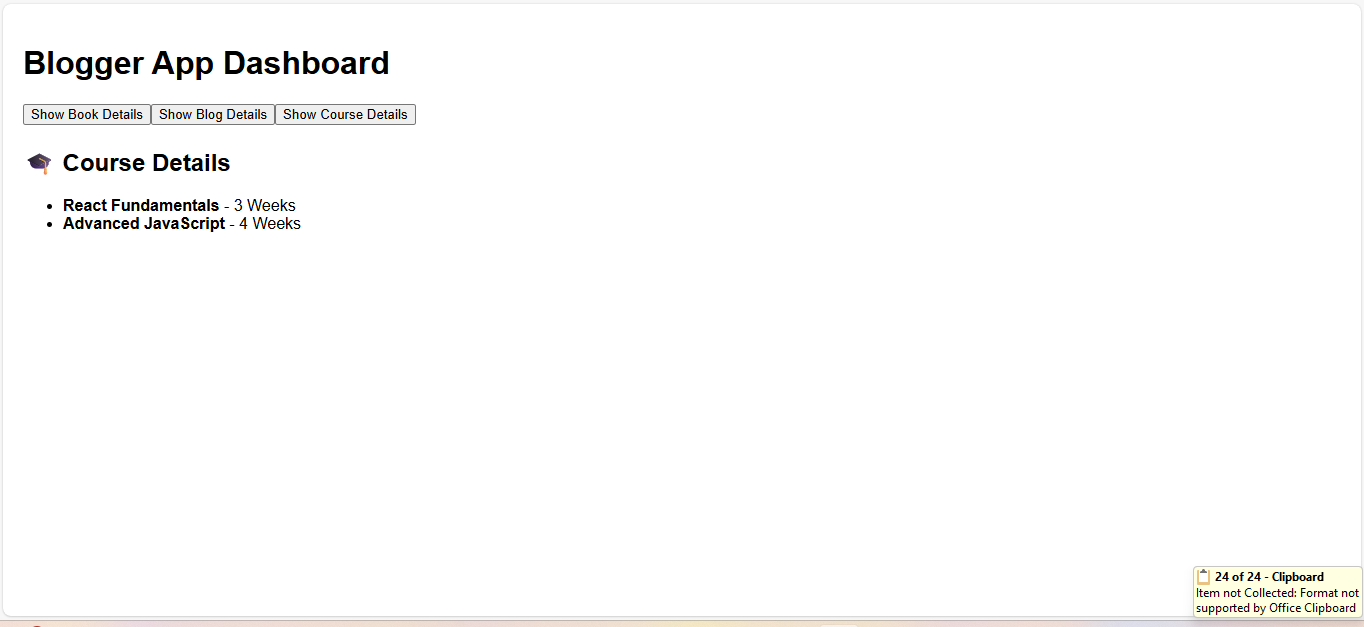
**Output:**

****

**BlogDetails:**

****

**CourseDetails:**

****