**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:

App.js

import React from 'react';

import ListofPlayers from './ListofPlayers';

import IndianPlayers from './IndianPlayers';

function App() {

  const flag = true;

  return (

    <div className="App">

      <h1>Cricket App</h1>

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

    </div>

  );

}

export default App;

IndianPlayers.js

import React from 'react';

const IndianPlayers = () => {

  const players = ["Virat", "Rohit", "Dhoni", "Jadeja", "Shami", "Ashwin", "Rahul", "Pant", "Hardik", "Bumrah", "Surya"];

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

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

  const [odd1, odd2, ...restOdd] = oddTeam;

  const [even1, even2, ...restEven] = evenTeam;

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

  const RanjiTrophyPlayers = ["Pujara", "Iyer", "Thakur"];

  const allPlayers = [...T20players, ...RanjiTrophyPlayers];

  return (

    <div>

      <h2>Odd Team Players:</h2>

      <ul>

        {[odd1, odd2, ...restOdd].map((p, i) => <li key={i}>{p}</li>)}

      </ul>

      <h2>Even Team Players:</h2>

      <ul>

        {[even1, even2, ...restEven].map((p, i) => <li key={i}>{p}</li>)}

      </ul>

      <h2>All Merged Players (T20 + Ranji Trophy):</h2>

      <ul>

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

      </ul>

    </div>

  );

};

export default IndianPlayers;

ListofPlayers.js

import React from 'react';

const ListofPlayers = () => {

  const players = [

    { name: "Virat", score: 85 },

    { name: "Rohit", score: 75 },

    { name: "Dhoni", score: 65 },

    { name: "Jadeja", score: 55 },

    { name: "Shami", score: 90 },

    { name: "Ashwin", score: 72 },

    { name: "Rahul", score: 68 },

    { name: "Pant", score: 77 },

    { name: "Hardik", score: 40 },

    { name: "Bumrah", score: 92 },

    { name: "Surya", score: 88 }

  ];

  const below70 = 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>

      <h2>Players with Score Below 70:</h2>

      <ul>

        {below70.map((player, index) => (

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

        ))}

      </ul>

    </div>

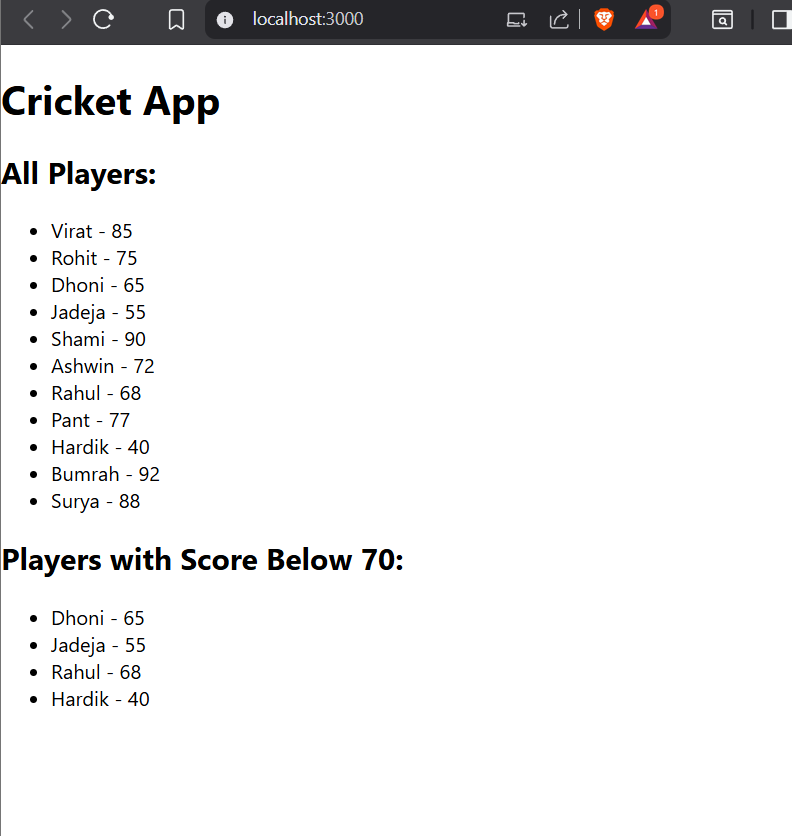
  );

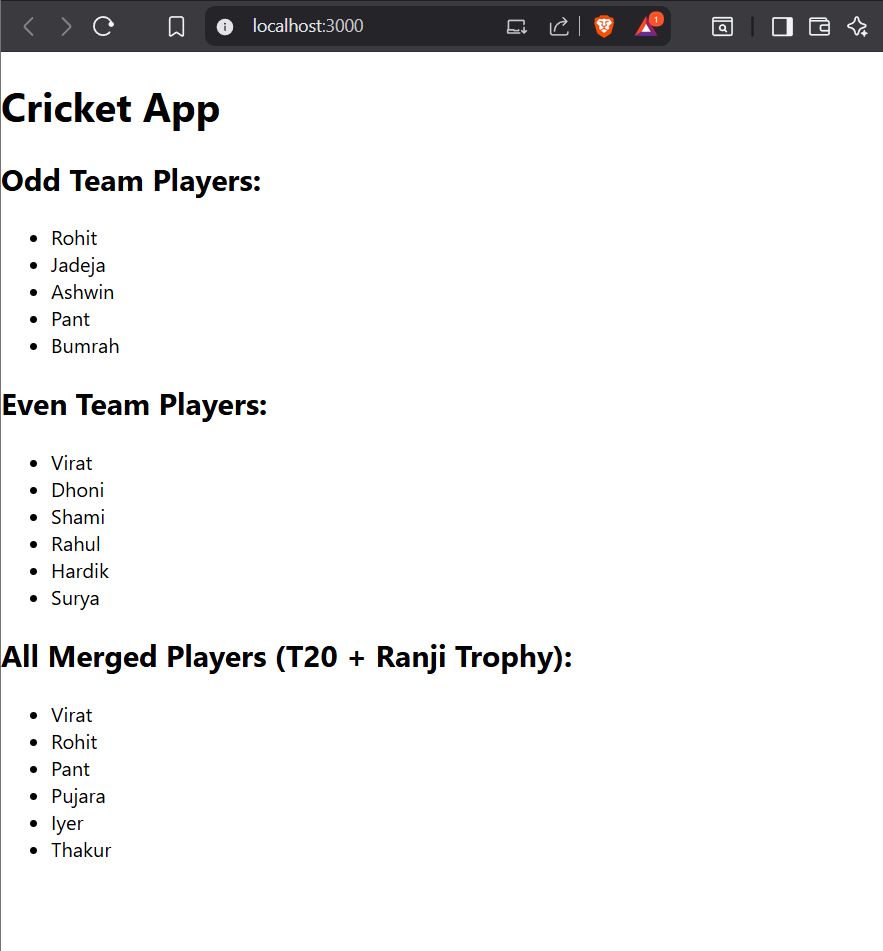
};

export default ListofPlayers;

output:

flag = true;



Flag = false;

**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:

App.js

import React from 'react';

import OfficeSpace from './OfficeSpace';

function App() {

  return (

    <div className="App">

      <OfficeSpace />

    </div>

  );

}

export default App;

OfficeSpace.js

import React from 'react';

import './OfficeSpace.css'; // CSS styling file

const OfficeSpace = () => {

  const officeList = [

    {

      name: "Tech Park",

      rent: 75000,

      address: "Whitefield, Bangalore",

      image: "https://via.placeholder.com/300x150?text=Tech+Park"

    },

    {

      name: "Startup Hub",

      rent: 55000,

      address: "Hitech City, Hyderabad",

      image: "https://via.placeholder.com/300x150?text=Startup+Hub"

    },

    {

      name: "Corporate Tower",

      rent: 90000,

      address: "Gurgaon, Haryana",

      image: "https://via.placeholder.com/300x150?text=Corporate+Tower"

    }

  ];

  return (

    <div>

      <h1>Office Spaces Available</h1>

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

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

          <img src={office.image} alt={office.name} className="office-image" />

          <h2>{office.name}</h2>

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

          <p>

            <strong>Rent:</strong>{' '}

            <span className={office.rent < 60000 ? 'red' : 'green'}>

              ₹{office.rent.toLocaleString()}

            </span>

          </p>

        </div>

      ))}

    </div>

  );

};

export default OfficeSpace;

OfficeSpace.css

.office-card {

  border: 1px solid #ccc;

  padding: 16px;

  margin: 16px auto;

  width: 320px;

  border-radius: 10px;

  box-shadow: 2px 2px 12px #eee;

  text-align: center;

  font-family: Arial, sans-serif;

}

.office-image {

  width: 100%;

  height: auto;

  border-radius: 5px;

}

.red {

  color: red;

  font-weight: bold;

}

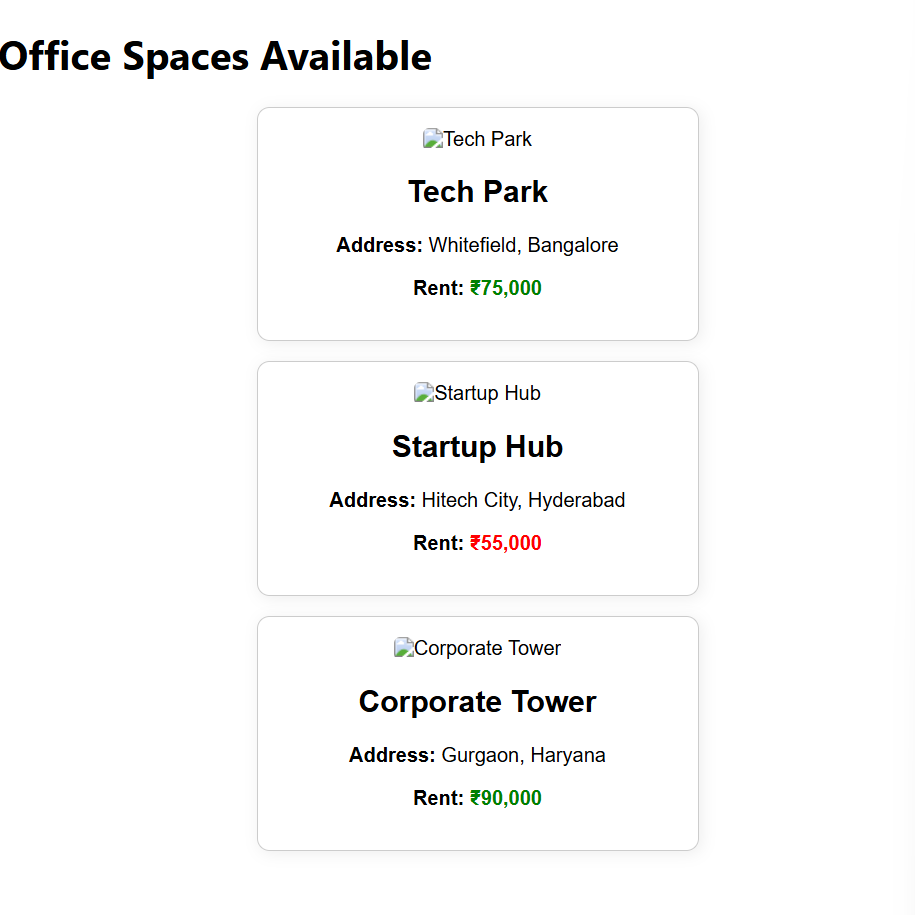
.green {

  color: green;

  font-weight: bold;

}

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.



Solution:

App.js

import React from 'react';

import CounterAndConverter from './CounterAndConverter';

function App() {

  return (

    <div className="App">

      <CounterAndConverter />

    </div>

  );

}

export default App;

CounterAndConverter.js

import React, { useState } from 'react';

const CounterAndConverter = () => {

  const [count, setCount] = useState(0);

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

  const [euros, setEuros] = useState('');

  const exchangeRate = 0.011;

  const increment = () => {

    setCount(prev => prev + 1);

    sayHello();

  };

  const decrement = () => {

    setCount(prev => prev - 1);

  };

  const sayHello = () => {

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

  };

  const sayWelcome = (msg) => {

    alert(msg);

  };

  const handleClick = () => {

    alert("I was clicked");

  };

  const handleSubmit = () => {

    const euroValue = (parseFloat(rupees) \* exchangeRate).toFixed(2);

    setEuros(euroValue);

  };

  const handleReverseSubmit = () => {

    const inrValue = (parseFloat(euros) / exchangeRate).toFixed(2);

    setRupees(inrValue);

  };

  return (

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

      <h2>Counter: {count}</h2>

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

      <button onClick={decrement} style={{ marginLeft: "10px" }}>Decrement</button>

      <br /><br />

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

      <br /><br />

      <button onClick={handleClick}>Synthetic OnClick</button>

      <hr />

      <h2>Currency Converter</h2>

      <div>

        <label>Rupees (INR): </label>

        <input

          type="number"

          value={rupees}

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

        />

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

      </div>

      <br />

      <div>

        <label>Euro (EUR): </label>

        <input

          type="number"

          value={euros}

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

        />

        <button onClick={handleReverseSubmit}>Convert to Rupees</button>

      </div>

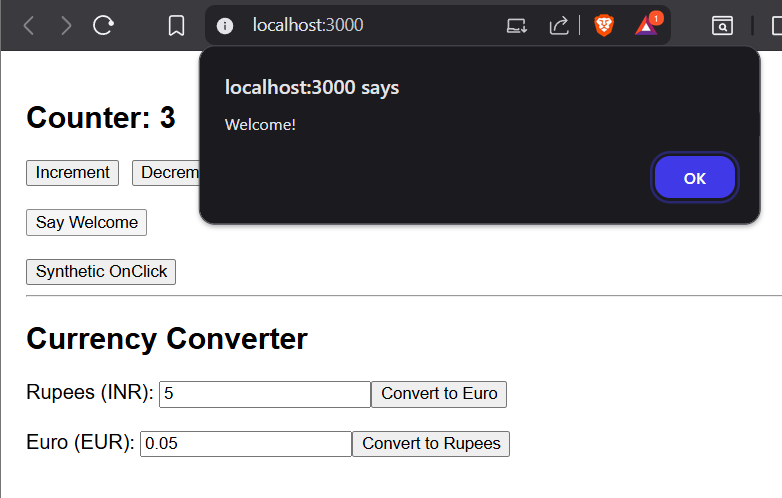
    </div>

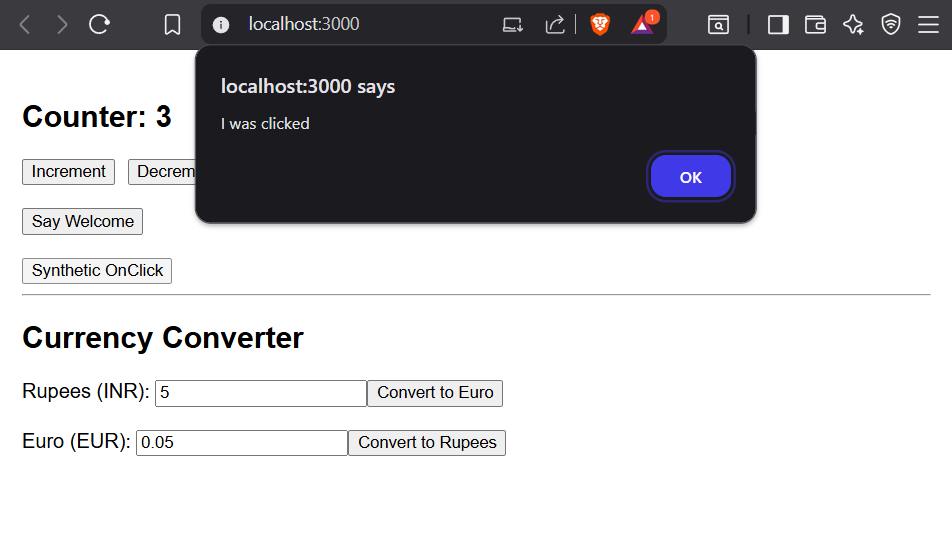
  );

};

export default CounterAndConverter;

Output:





**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:

App.js

import React, { useState } from 'react';

import GuestPage from './GuestPage';

import UserPage from './UserPage';

const App = () => {

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

  const handleLogin = () => setIsLoggedIn(true);

  const handleLogout = () => setIsLoggedIn(false);

  return (

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

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

        <button onClick={isLoggedIn ? handleLogout : handleLogin}>

          {isLoggedIn ? "Logout" : "Login"}

        </button>

      </header>

      {isLoggedIn ? <UserPage /> : <GuestPage />}

    </div>

  );

};

export default App;

GuestPage.js

import React from 'react';

import FlightList from './FlightList';

const GuestPage = () => {

  return (

    <div>

      <h1>Welcome Guest</h1>

      <p>Please log in to book tickets.</p>

      <FlightList />

    </div>

  );

};

export default GuestPage;

UserPage.js

import React from 'react';

import FlightList from './FlightList';

const UserPage = () => {

  const handleBook = () => {

    alert("Your ticket has been booked!");

  };

  return (

    <div>

      <h1>Welcome Back, User!</h1>

      <FlightList />

      <button onClick={handleBook} style={{ marginTop: '20px' }}>

        Book Selected Flight

      </button>

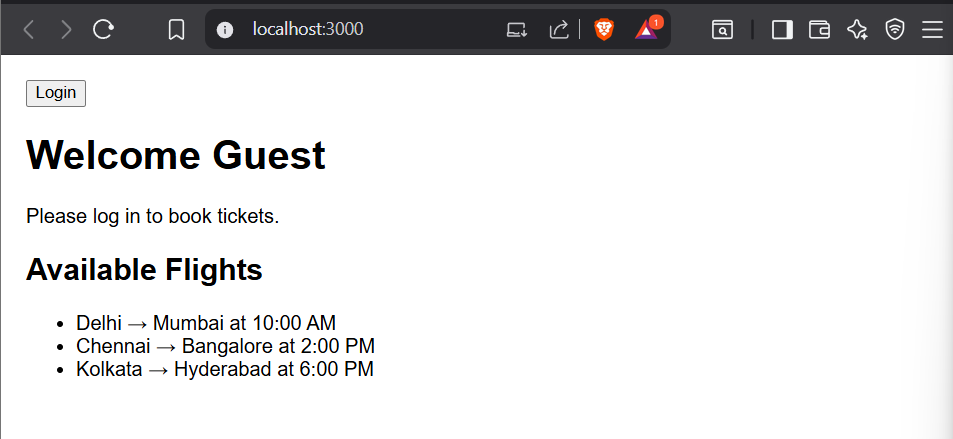
    </div>

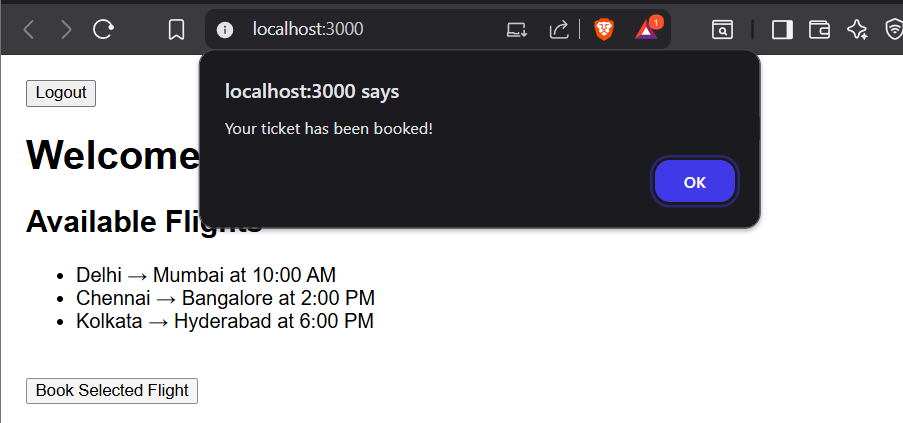
  );

};

export default UserPage;

Output:





**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:**





