**digital nurture Deep skilling-Week7**

## **REACTJS-HOL**

## **exercise 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:**



**CODE:  
ListofPlayer.js**

import React from 'react';

function ListofPlayers() {

  const players = [

    { name: 'Virat Kohli', score: 85 },

    { name: 'Rohit Sharma', score: 75 },

    { name: 'KL Rahul', score: 65 },

    { name: 'Shubman Gill', score: 50 },

    { name: 'Hardik Pandya', score: 90 },

    { name: 'Jasprit Bumrah', score: 88 },

    { name: 'Mohammed Shami', score: 68 },

    { name: 'Suryakumar Yadav', score: 55 },

    { name: 'Ishan Kishan', score: 76 },

    { name: 'Axar Patel', score: 49 },

    { name: 'Kuldeep Yadav', score: 92 }

  ];

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

  return (

    <div style={{ padding: '20px' }}>

      <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';

function IndianPlayers() {

  const team = ['Virat', 'Rohit', 'Rahul', 'Bumrah', 'Hardik', 'Jadeja'];

  const [odd1, even1, odd2, even2, odd3, even3] = team;

  const T20Players = ['Kohli', 'Rohit', 'Bumrah'];

  const RanjiPlayers = ['Pujara', 'Iyer', 'Shaw'];

  const mergedPlayers = [...T20Players, ...RanjiPlayers];

  return (

    <div style={{ padding: '20px' }}>

      <h2>Team Split using Destructuring</h2>

      <p>Odd Team Players: {odd1}, {odd2}, {odd3}</p>

      <p>Even Team Players: {even1}, {even2}, {even3}</p>

      <h3>Merged T20 & Ranji Players:</h3>

      <ul>

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

          <li key={index}>{player}</li>

        ))}

      </ul>

    </div>

  );

}

export default IndianPlayers;

**App.js**

import ListofPlayers from './Cricketapp/ListofPlayers';

import IndianPlayers from './Cricketapp/IndianPlayers';

function App() {

  const flag = false; // Change to false to see the other component

  return (

    <div className="App">

      <h1>Cricket App</h1>

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

    </div>

  );

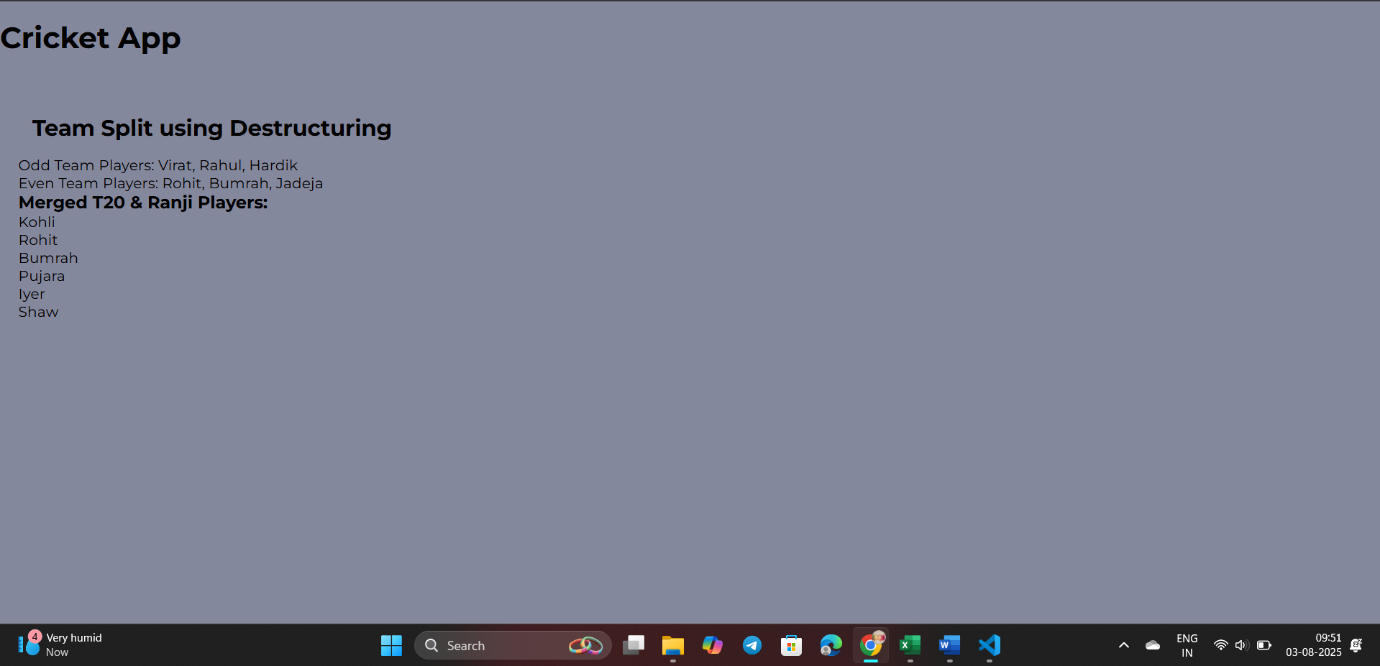
}

export default App;

**OUTPUT:**

**flag==true  
**

**flag==false**

****

**EXERCISE 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:**





**CODE:  
App.js**

import React from 'react';

function App() {

  const office1 = {

    name: "CoWork Central",

    rent: 55000,

    address: "123 Main St, Bangalore",

    image: "https://images.unsplash.com/photo-1570129477492-45c003edd2be?ixlib=rb-4.0.3&auto=format&fit=crop&w=800&q=60"

  };

  const officeSpaces = [

    {

      name: "Regus Office",

      rent: 45000,

      address: "45 MG Road, Chennai",

      image: "https://images.unsplash.com/photo-1573164574572-cb89e39749b4?auto=format&fit=crop&w=800&q=60"

    },

    {

      name: "WeWork Prestige",

      rent: 78000,

      address: "Indiranagar, Bangalore",

      image: "https://images.unsplash.com/photo-1588702547923-7093a6c3ba33?auto=format&fit=crop&w=800&q=60"

    },

    {

      name: "Innov8 Hub",

      rent: 60000,

      address: "Hitech City, Hyderabad",

      image: "https://images.unsplash.com/photo-1573164574572-cb89e39749b4?ixlib=rb-4.0.3&auto=format&fit=crop&w=800&q=60"

    }

  ];

  return (

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

      <h1>Office Space Rental</h1>

      <div style={{ border: '1px solid gray', padding: '10px', marginBottom: '20px' }}>

        <img src={office1.image} alt="Office" style={{ width: '100%', maxWidth: '300px' }} />

        <h2>{office1.name}</h2>

        <p style={{ color: office1.rent < 60000 ? 'red' : 'green' }}>

          Rent: ₹{office1.rent}

        </p>

        <p>Address: {office1.address}</p>

      </div>

      <h2>More Office Spaces</h2>

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

        <div key={index} style={{ border: '1px solid #ccc', marginBottom: '15px', padding: '10px' }}>

          <img src={office.image} alt={`Office ${index}`} style={{ width: '100%', maxWidth: '300px' }} />

          <h3>{office.name}</h3>

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

            Rent: ₹{office.rent}

          </p>

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

        </div>

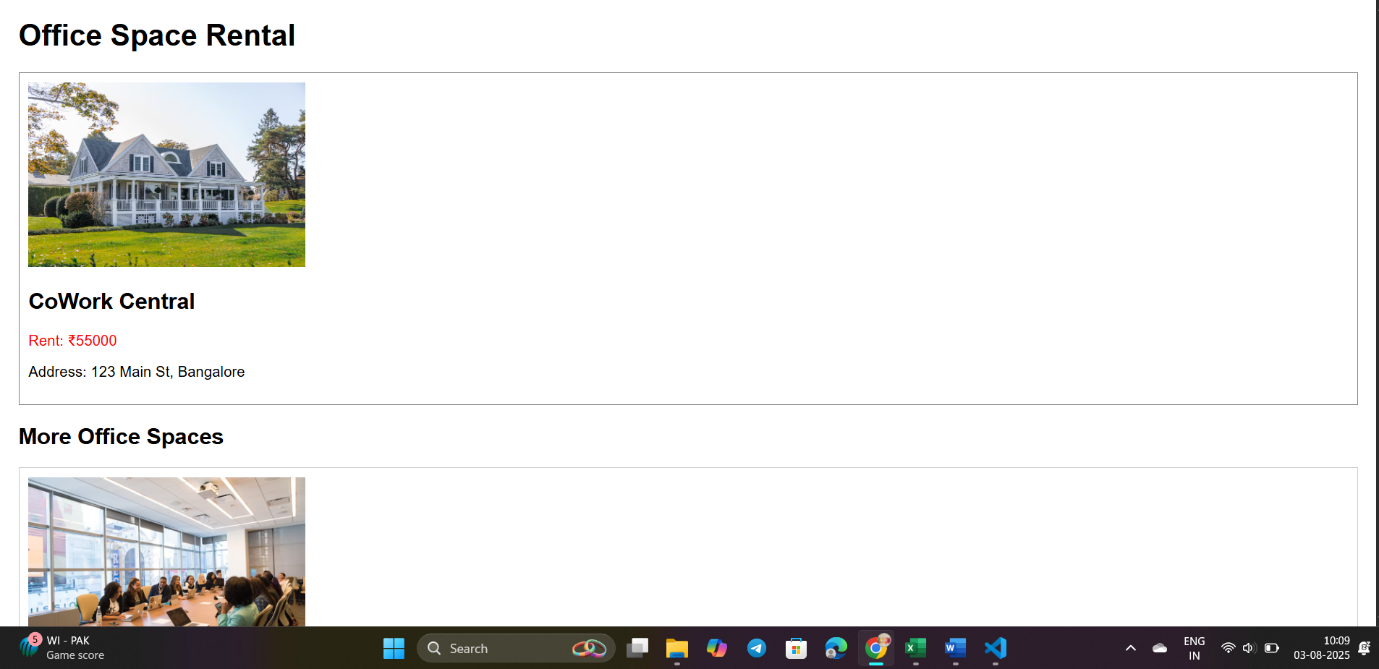
      ))}

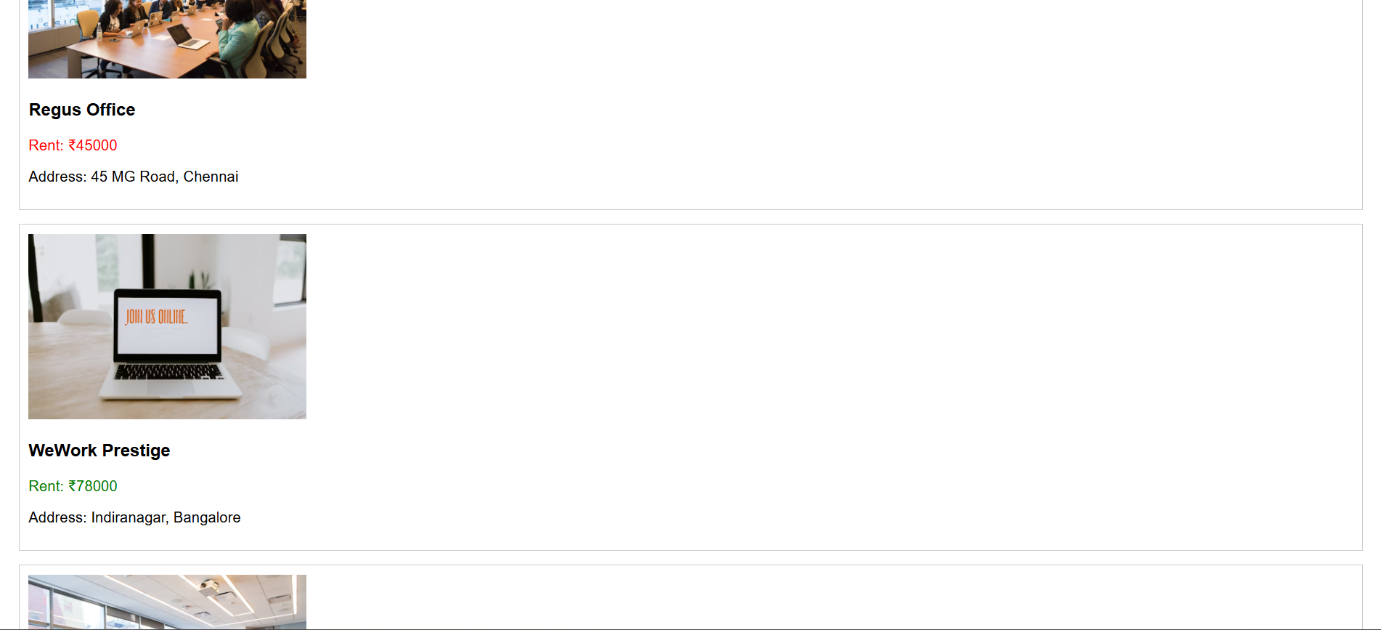
    </div>

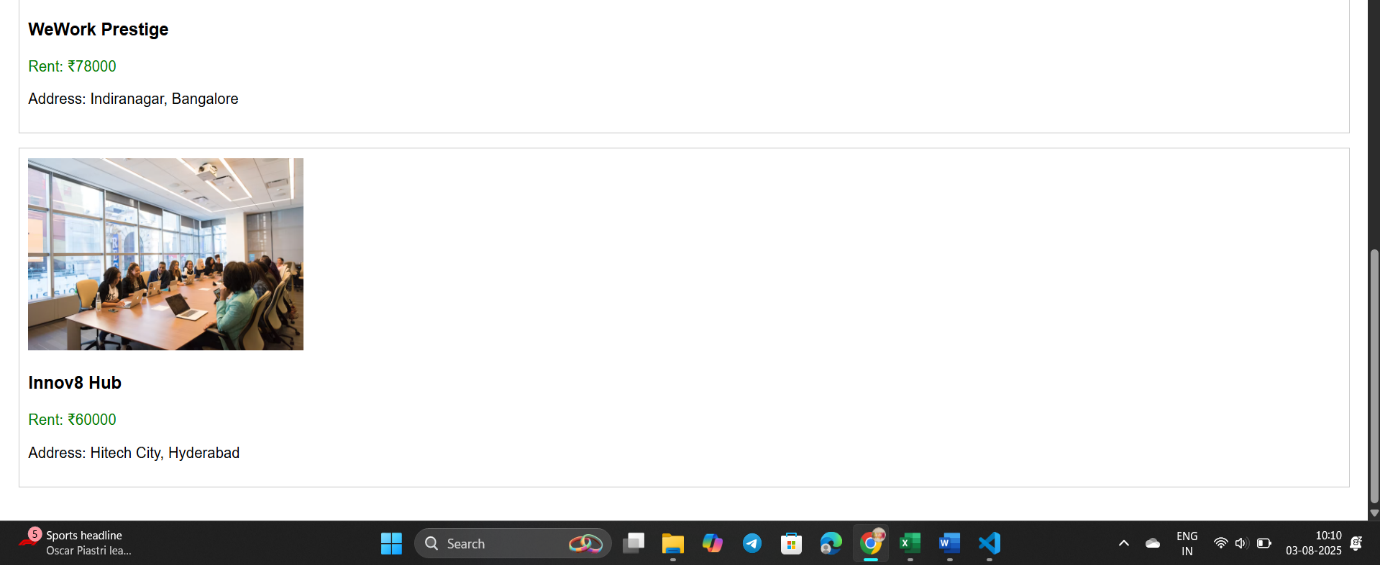
  );

}

export default App;

**OUTPUT:**





## **EXERCISE 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.



**CODE:  
CurrencyConvertor.js**

// CurrencyConvertor.js

import React, { Component } from 'react';

class CurrencyConvertor extends Component {

  constructor() {

    super();

    this.state = {

      rupees: '',

      euro: null

    };

  }

  handleChange = (e) => {

    this.setState({ rupees: e.target.value });

  };

  handleSubmit = () => {

    const conversionRate = 0.011; // 1 INR ≈ 0.011 EUR

    const euroValue = this.state.rupees \* conversionRate;

    this.setState({ euro: euroValue.toFixed(2) });

  };

  render() {

    return (

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

        <h2>Currency Convertor</h2>

        <input

          type="number"

          placeholder="Enter INR"

          value={this.state.rupees}

          onChange={this.handleChange}

        />

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

        {this.state.euro !== null && (

          <p>{this.state.rupees} INR = €{this.state.euro}</p>

        )}

      </div>

    );

  }

}

export default CurrencyConvertor;

**App.js**

import React, { Component } from 'react';

import CurrencyConvertor from './Eventexamplesapp/CurrencyConvertor';

class App extends Component {

  constructor() {

    super();

    this.state = {

      counter: 0

    };

  }

  increment = () => {

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

  };

  sayHello = () => {

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

  };

  increaseHandler = () => {

    this.increment();

    this.sayHello();

  };

  decrement = () => {

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

  };

  sayWelcome = (message) => {

    alert(message);

  };

  handleClick = (e) => {

    e.preventDefault();

    alert("I was clicked");

  };

  render() {

    return (

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

        <h1>React Event Handling Examples</h1>

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

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

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

        <br /><br />

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

        <br /><br />

        <button onClick={this.handleClick}>Synthetic Event Button</button>

        <br /><br />

        <CurrencyConvertor />

      </div>

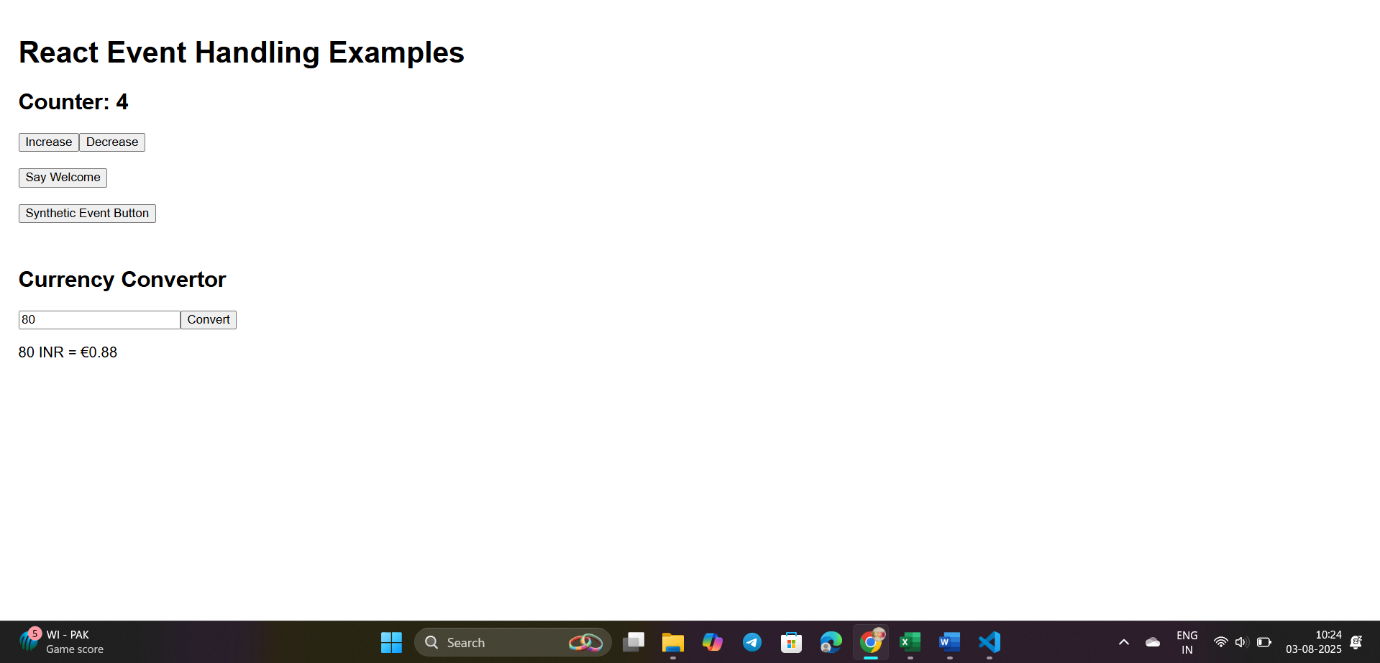
    );

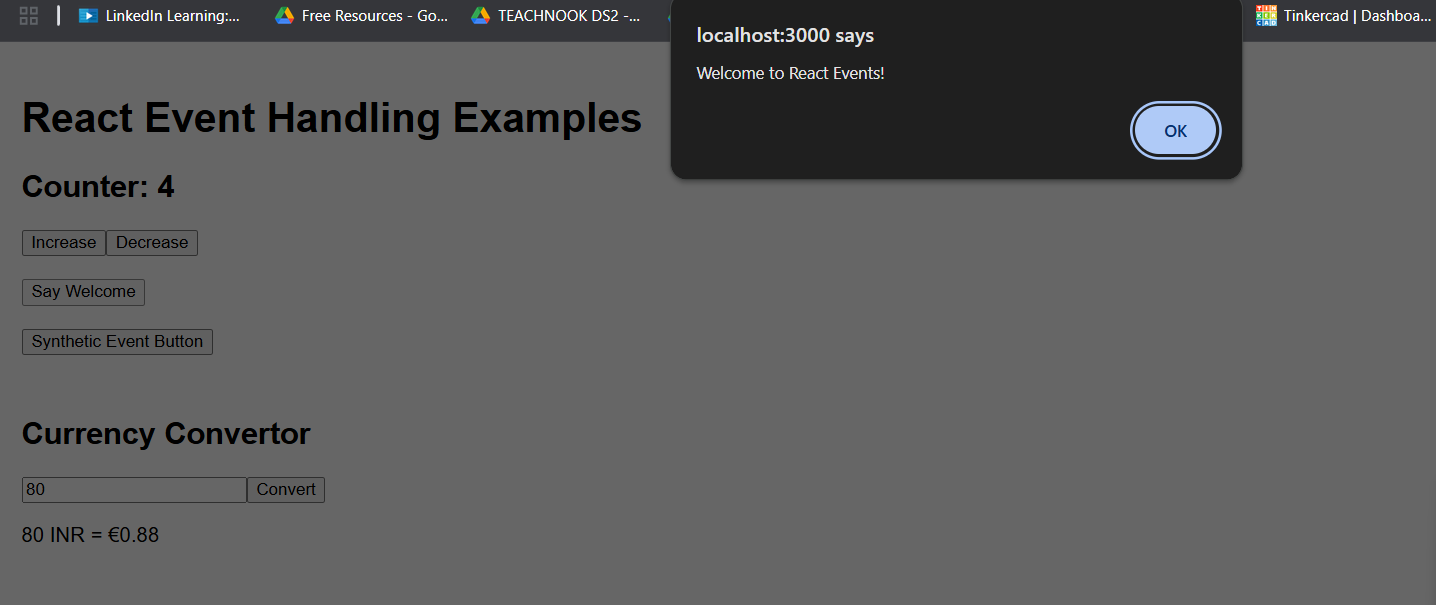
  }

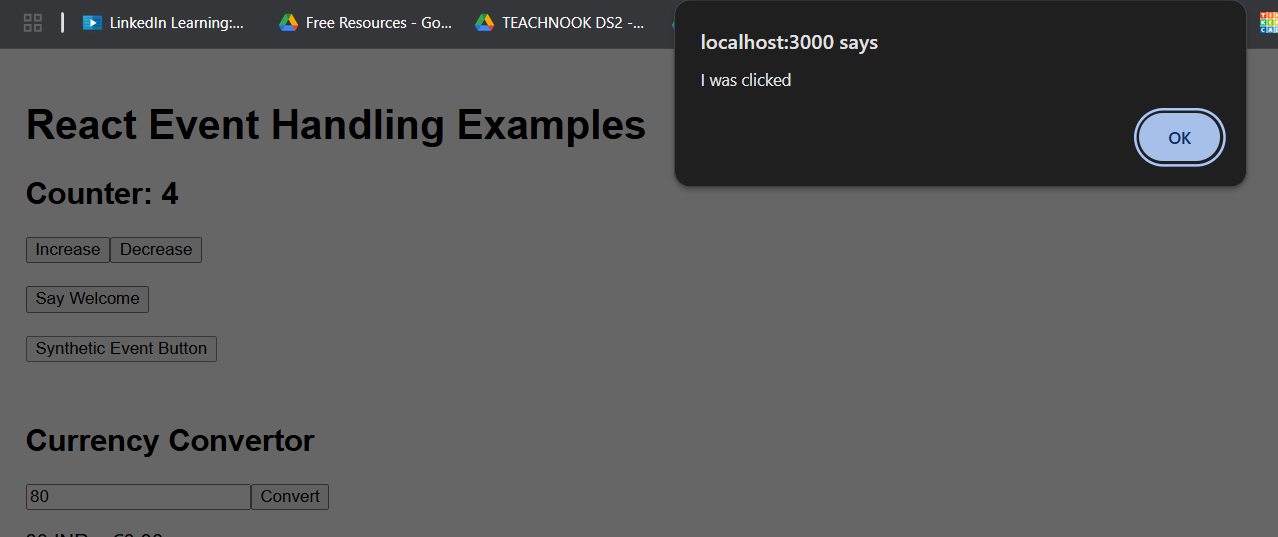
}

export default App;

**OUTPUT:**

****

****

****

**EXERCISE 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:**







**CODE:  
GuestPage.js**

import React from 'react';

function GuestPage({ onLogin }) {

  return (

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

      <h2>Welcome, Guest!</h2>

      <p>You can browse flight details, but booking requires login.</p>

      <button onClick={onLogin}>Login</button>

    </div>

  );

}

export default GuestPage;

**UserPage.js**

import React from 'react';

function UserPage({ onLogout }) {

  return (

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

      <h2>Welcome, User!</h2>

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

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

    </div>

  );

}

export default UserPage;

**App.js**

//TicketBookingApp

// App.js

import React, { Component } from 'react';

import GuestPage from './TicketBookingApp/GuestPage';

import UserPage from './TicketBookingApp/UserPage';

class App extends Component {

  constructor() {

    super();

    this.state = {

      isLoggedIn: false

    };

  }

  handleLogin = () => {

    this.setState({ isLoggedIn: true });

  };

  handleLogout = () => {

    this.setState({ isLoggedIn: false });

  };

  render() {

    let page;

    if (this.state.isLoggedIn) {

      page = <UserPage onLogout={this.handleLogout} />;

    } else {

      page = <GuestPage onLogin={this.handleLogin} />;

    }

    return (

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

        <h1>Ticket Booking Application</h1>

        {page}

      </div>

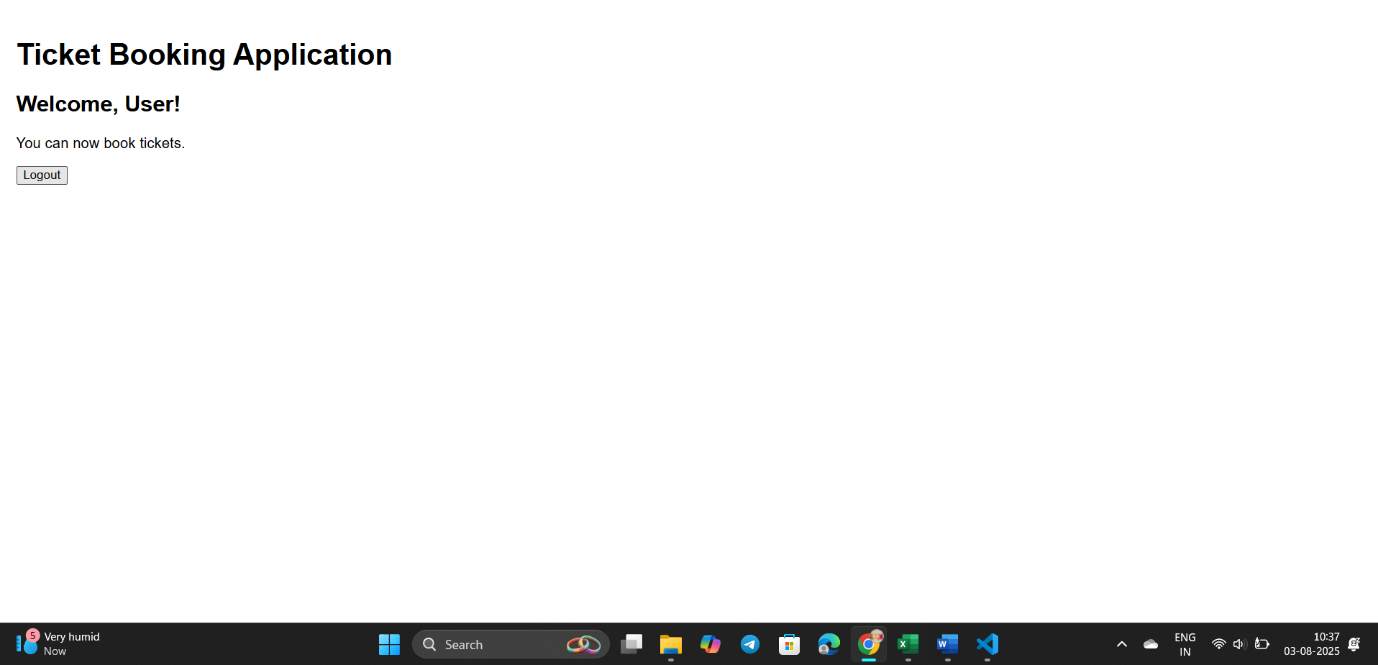
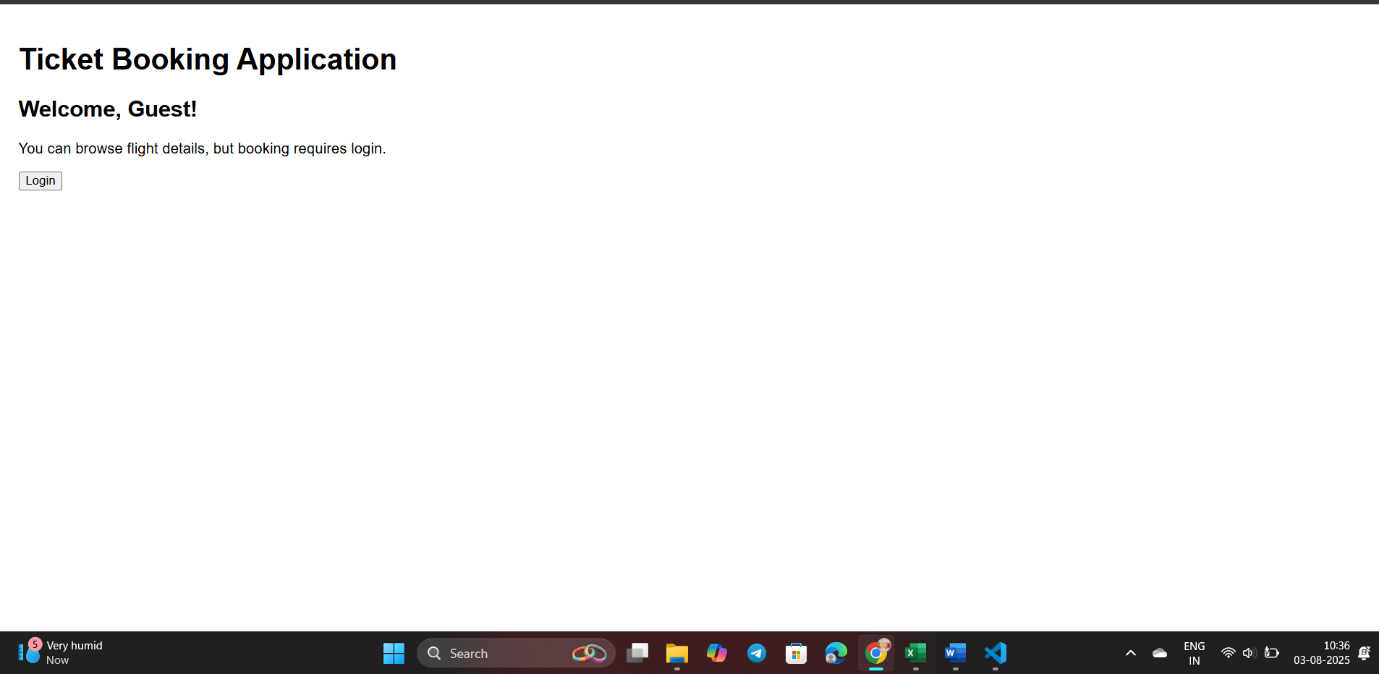
    );

  }

}

export default App;

**OUTPUT:**

****

**EXERCISE 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:**







**CODE:**

**App.js:**

import './App.css';

import BookDetails from './Bloggerapp/BlogDetails';

import BlogDetails from './Bloggerapp/BookDetails';

import CourseDetails from './Bloggerapp/CourseDetails';

function App() {

  return (

    <div className="container">

      <CourseDetails />

      <BookDetails />

      <BlogDetails />

    </div>

  );

}

export default App;

**App.css**

.container {

  display: flex;

  justify-content: space-around;

  margin: 40px;

  font-family: Arial, sans-serif;

}

.section {

  border-left: 4px solid green;

  padding-left: 20px;

  width: 25%;

}

**BookDetails.js**

import React from 'react';

const books = [

  { title: 'Master React', price: 670 },

  { title: 'Deep Dive into Angular 11', price: 800 },

  { title: 'Mongo Essentials', price: 450 }

];

const BookDetails = () => {

  return (

    <div className="section">

      <h2>Book Details</h2>

      {books.map((book, index) => (

        <div key={index}>

          <h3>{book.title}</h3>

          <p>{book.price}</p>

        </div>

      ))}

    </div>

  );

};

export default BookDetails;

**BlogDetails.js**

import React from 'react';

const blogs = [

  {

    title: 'React Learning',

    author: 'Stephen Biz',

    content: 'Welcome to learning React!'

  },

  {

    title: 'Installation',

    author: 'Schwezdenier',

    content: 'You can install React from npm.'

  }

];

const BlogDetails = () => {

  return (

    <div className="section">

      <h2>Blog Details</h2>

      {blogs.map((blog, index) => (

        <div key={index}>

          <h3>{blog.title}</h3>

          <strong>{blog.author}</strong>

          <p>{blog.content}</p>

        </div>

      ))}

    </div>

  );

};

export default BlogDetails;

**CourseDetails.js**

import React from 'react';

const CourseDetails = () => {

  const courses = [

    { name: 'Angular', date: '4/5/2021' },

    { name: 'React', date: '6/3/20201' }

  ];

  return (

    <div className="section">

      <h2>Course Details</h2>

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

        <div key={index}>

          <h3>{course.name}</h3>

          <p>{course.date}</p>

        </div>

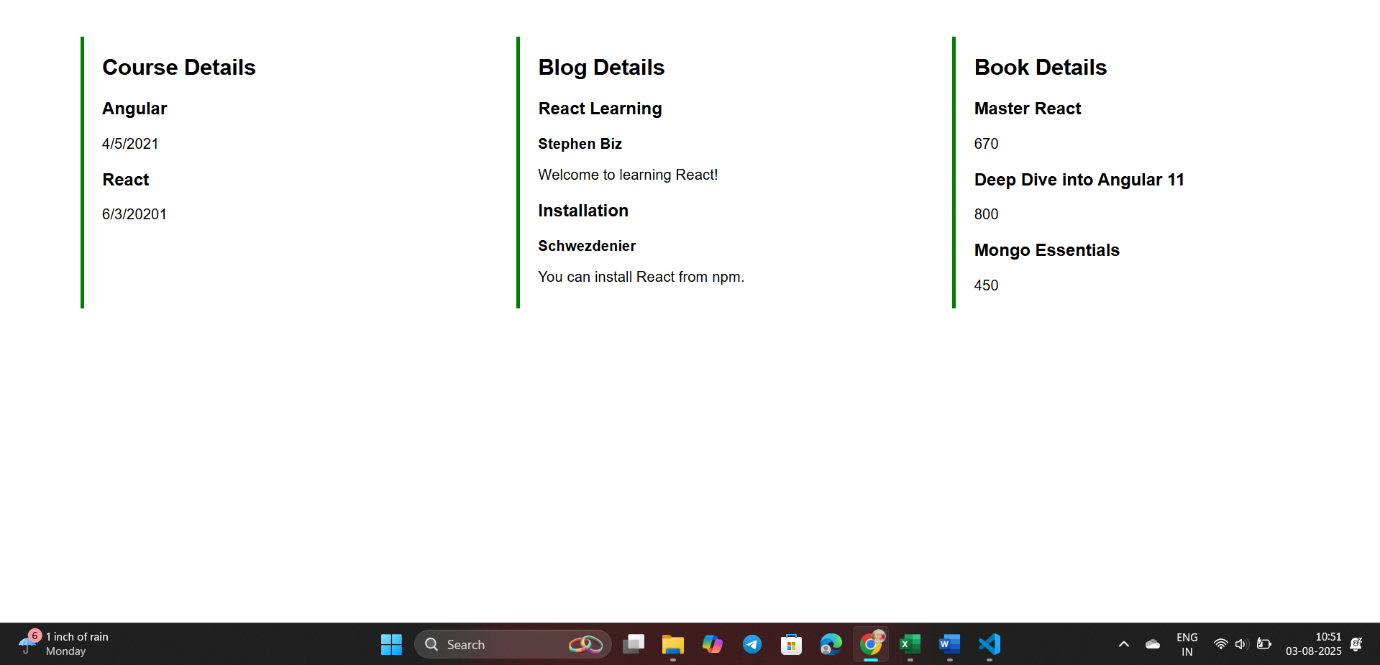
      ))}

    </div>

  );

};

export default CourseDetails;

**OUTPUT:  
**