HANDSON - 1

Question :

Objectives

- Define SPA and its benefits Define React and identify its working Identify the di erences between SPA and MPA
- Explain Pros & Cons of Single-Page Application
- Explain about React
- Define virtual DOM
- Explain Features of React

In this hands-on lab, you will learn how to:

- Set up a react environment
- Use create-react-app

Prerequisites

The following is required to complete this hands-on lab:

- Node.js
- NPM
- Visual Studio Code

Notes

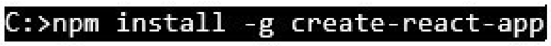
Estimated time to complete this lab: 30 minutes.

Create a new React Application with the name “myfirstreact”, Run the application to print “welcome to the first session of React” as heading of that page.

1. To create a new React app, Install Nodejs and Npm from the following link:

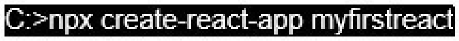
   https://nodejs.org/en/download/

2. Install Create-react-app by running the following command in the command prompt:



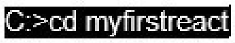
```
C:>npm install -g create-react-app
```

3. To create a React Application with the name of “myfirstreact”, type the following command:



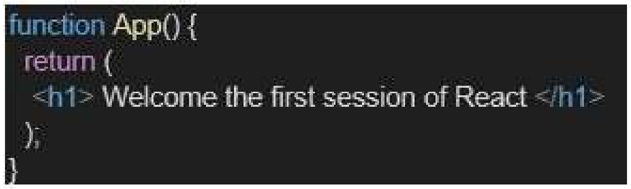
```
C:>npx create-react-app myfirstreact
```

4. Once the App is created, navigate into the folder of myfirstreact by typing the following command:



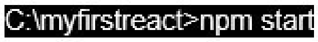
```
C:>cd myfirstreact
```

5. Open the folder of myfirstreact in Visual Studio Code
6. Open the App.js file in Src Folder of myfirstreact
7. Remove the current content of “App.js”
8. Replace it with the following:



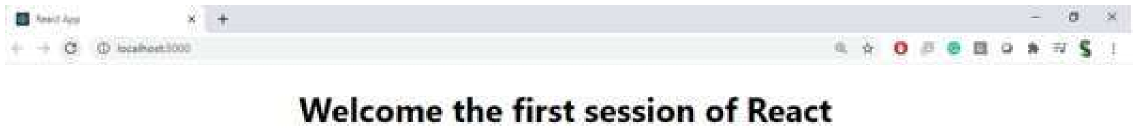
```
function App() {
  return (
   <h1> Welcome the first session of React </h1>
  );
}
```

9. Run the following command to execute the React application:



```
C:\myfirstreact>npm start
```

10. Open a new browser window and type “localhost:3000” in the address bar

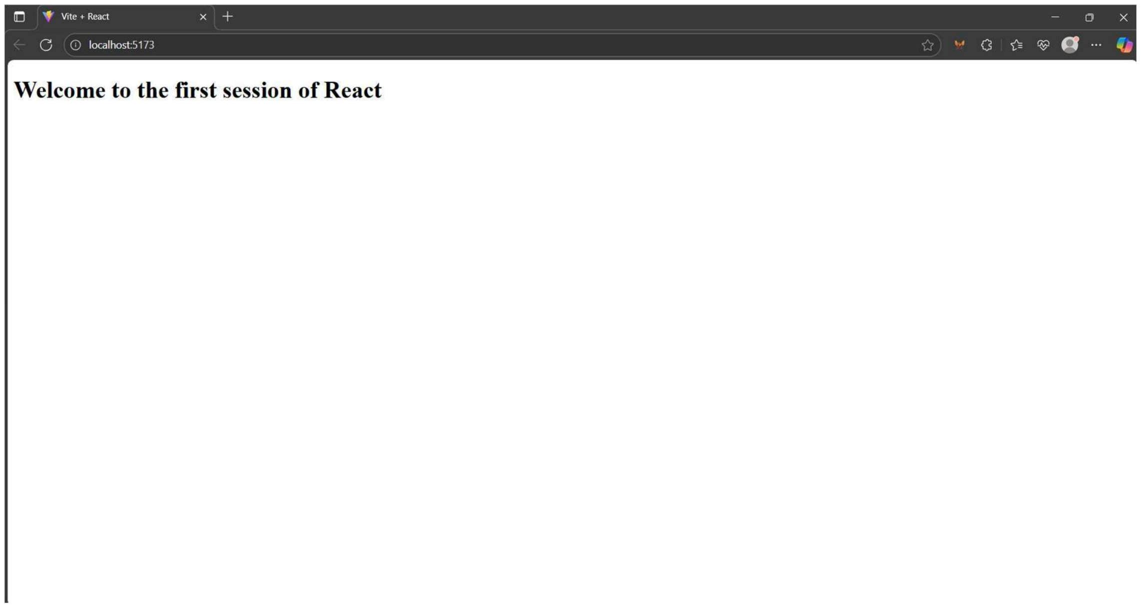


Welcome the first session of React

Code :

```
import React from 'react';

function App() {
return (
   <div>
     <h1>Welcome to the first session of React</h1>
   </div>
  );
}

export default App;
```

Output :



Welcome to the first session of React

HANDSON - 2

Objec ves

- 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 applica ons
- Use inline CSS in JSX

Prerequisites

The following is required to complete this hands-on lab:

- Node.js
- NPM
- Visual Studio Code

Notes

Es mated me to complete this lab: 60 minutes.

Create a React Applica on named “officespacerentalapp” which uses React JSX to create elements, a ributes and renders DOM to display the page.

Create an element to display the heading of the page.

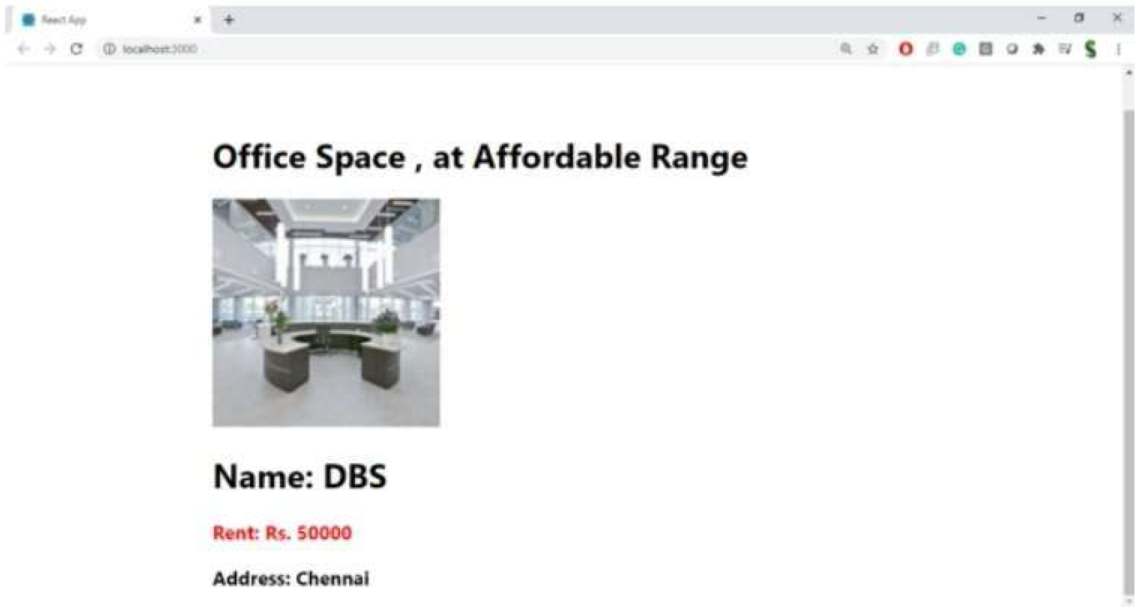
A ribute to display the image of the office space

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:



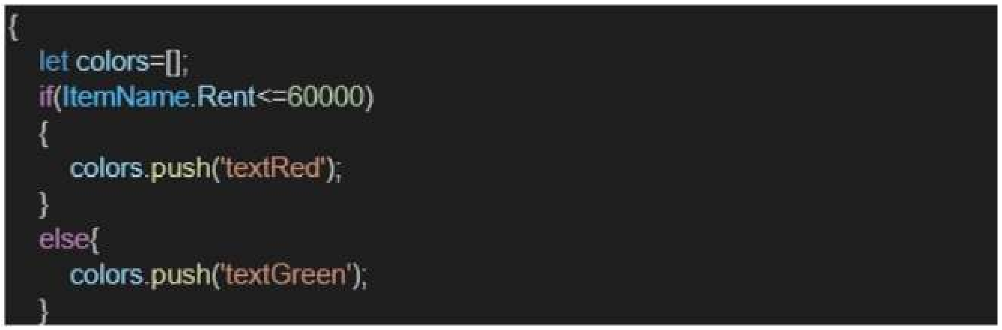
Office Space , at Affordable Range

Name: DBS

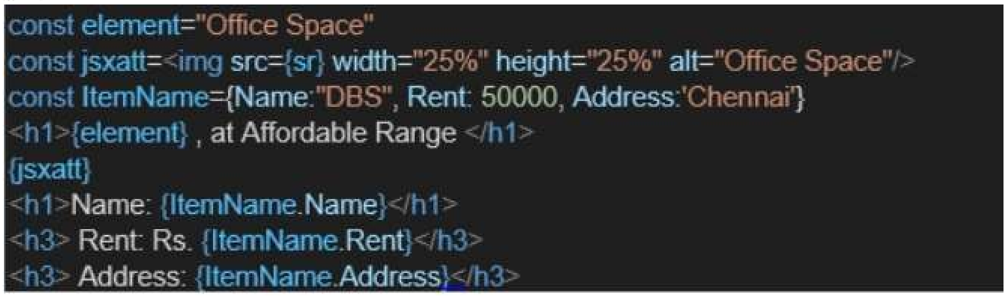
Rent: Rs. 50000

Address: Chennai

Hint:



```
{
    let colors=[];
    if(ItemName.Rent<=60000)
    {
        colors.push('textRed');
    }
    else{
        colors.push('textGreen');
    }
```



```
const element="Office Space"
const jsxatt=<img src={sr} width="25%" height="25%" alt="Office Space"/>
const ItemName={Name:"DBS", Rent: 50000, Address:'Chennai'}
<h1>{element} , at Affordable Range </h1>
{jsxatt}
<h1>Name: {ItemName.Name}</h1>
<h3> Rent: Rs. {ItemName.Rent}</h3>
<h3> Address: {ItemName.Address}</h3>
```

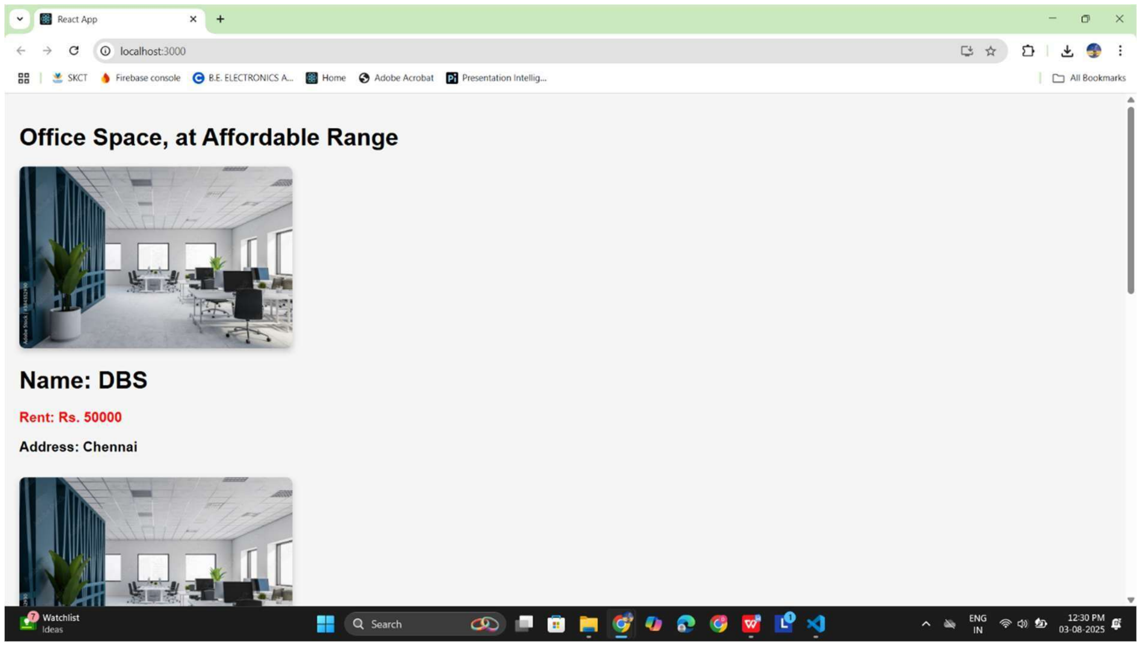
Answer import React from "react"; import "./App.css"; import officeImg from "./office.jpg"; // Make sure this image exists

```
func on App() {   const heading
= "Office Space";   const offices
= [
   { Name: "DBS", Rent: 50000, Address: "Chennai" },
   { Name: "Regus", Rent: 70000, Address: "Bangalore" },
   { Name: "SmartWorks", Rent: 45000, Address: "Pune" },
   { Name: "WeWork", Rent: 90000, Address: "Hyderabad" },
 ];

 return (
   <div style={{ padding: "20px", fontFamily: "Arial" }}>
     <h1>{heading}, at Affordable Range</h1>

     {offices.map((office, index) => {         const
rentStyle = {           color: office.Rent <= 60000 ?
"red" : "green",
       };

       return (
         <div key={index} style={{ marginBo om: "30px" }}>
           <img
src={officeImg}              alt="Office
Space"              width="25%"
height="25%"              style={{
borderRadius: "8px" }}
           />
           <h1>Name: {office.Name}</h1>
           <h3 style={rentStyle}>Rent: Rs. {office.Rent}</h3>
           <h3>Address: {office.Address}</h3>
         </div>
       );
```

```
      })}
    </div>
  );
}

export default App;
```

```
//css body {    background-
color: #f5f5f5;
}

img {   box-shadow: 0 4px 8px rgba(0, 0,
0, 0.2);
}
```

Output:



Office Space, at Affordable Range

Name: DBS

Rent: Rs. 50000

Address: Chennai

HANDSON - 3

Code : App.js import React, { useState } from "react";

import CurrencyConvertor from "./CurrencyConvertor";

function App() {   const [count,

setCount] = useState(0);

  const increment = () => {

setCount(count + 1);

  };

  const sayHello = () => {   alert("Hello!

Welcome to React Events");

  };

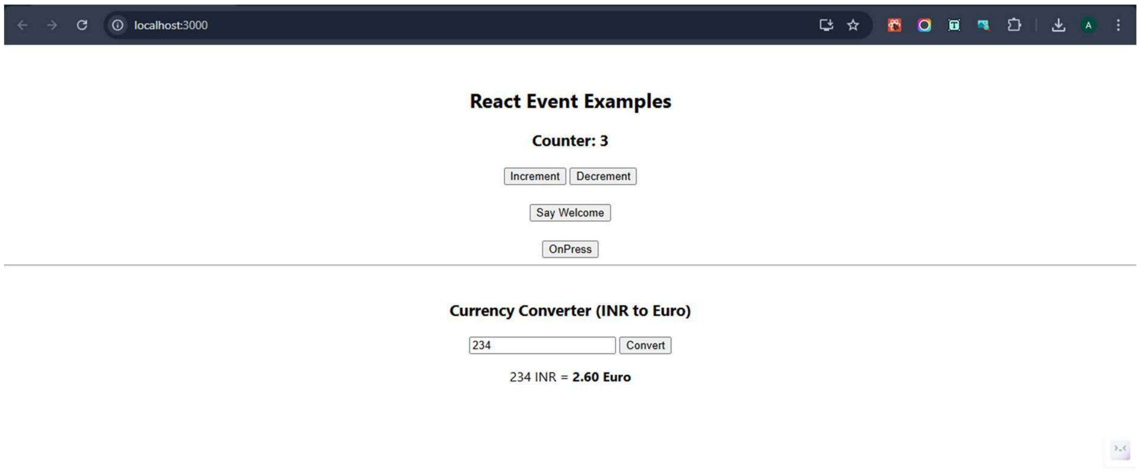
  const handleWelcome = (msg) => {

alert(msg);

  };

```
  const handleSyntheticEvent = (event) => {
alert("I was clicked");     console.log("Synthetic
Event Object:", event);
  };

  const handleIncrementClick = () => {
increment();
    sayHello();
  };

  return (
    <div style={{ textAlign: "center", marginTop: "50px" }}>
      <h2>React Event Examples</h2>



      <h3>Counter: {count}</h3>
      <button onClick={handleIncrementClick}>Increment</button>
      &nbsp;
      <button onClick={() => setCount(count - 1)}>Decrement</button>
      <br /><br />

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



      <button onClick={handleSyntheticEvent}>OnPress</button>
      <hr />



      <CurrencyConvertor />
    </div>
  );
}
export default App;
CurrencyConvertor.js import React, {
useState } from "react";

function CurrencyConvertor() {   const
[rupees, setRupees] = useState("");   const
[euro, setEuro] = useState(null);

  // 1 Euro = 90 INR (example rate)
const conversionRate = 90;

  // Handle form submit   const
handleSubmit = (e) => {
    e.preventDefault(); // prevent page refresh     if
(rupees !== "") {       setEuro((rupees /
conversionRate).toFixed(2));
    } else {
      alert("Please enter a valid amount in Rupees");
    }
  };
```

```
 return (
   <div style={{ marginTop: "40px" }}>
     <h3>Currency Converter (INR to Euro)</h3>
     <form onSubmit={handleSubmit}>
       <input          type="number"
placeholder="Enter amount in INR"
value={rupees}

         onChange={(e) => setRupees(e.target.value)}
       />
       &nbsp;
       <button type="submit">Convert</button>
     </form>


     {euro !== null && (
       <p>
         {rupees} INR = <strong>{euro} Euro</strong>
       </p>
     )}
   </div>
 );
}


export default CurrencyConvertor;
```


localhost:3000

React Event Examples

Counter: 3

Increment Decrement

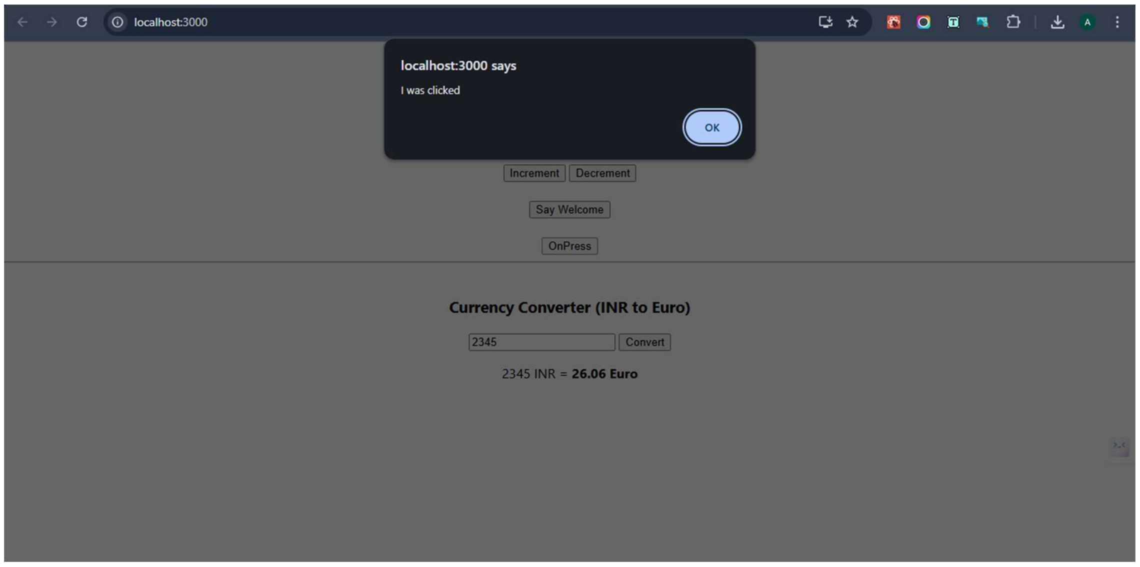
Say Welcome

OnPress

Currency Converter (INR to Euro)

234 Convert

234 INR = **2.60 Euro**



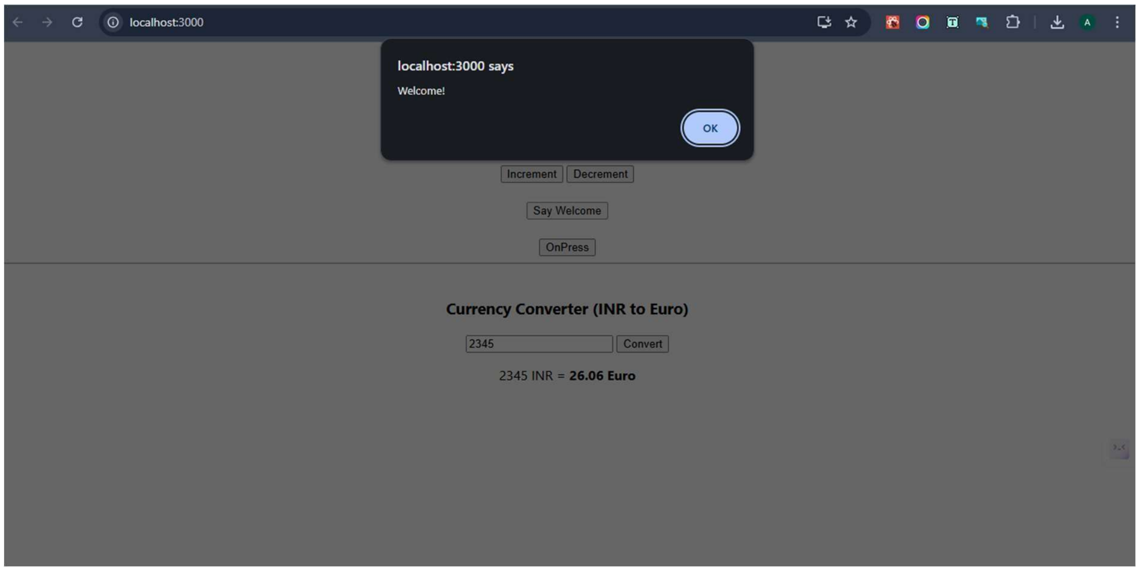
localhost:3000 says

I was clicked

Currency Converter (INR to Euro)

2345 Convert

2345 INR = **26.06 Euro**



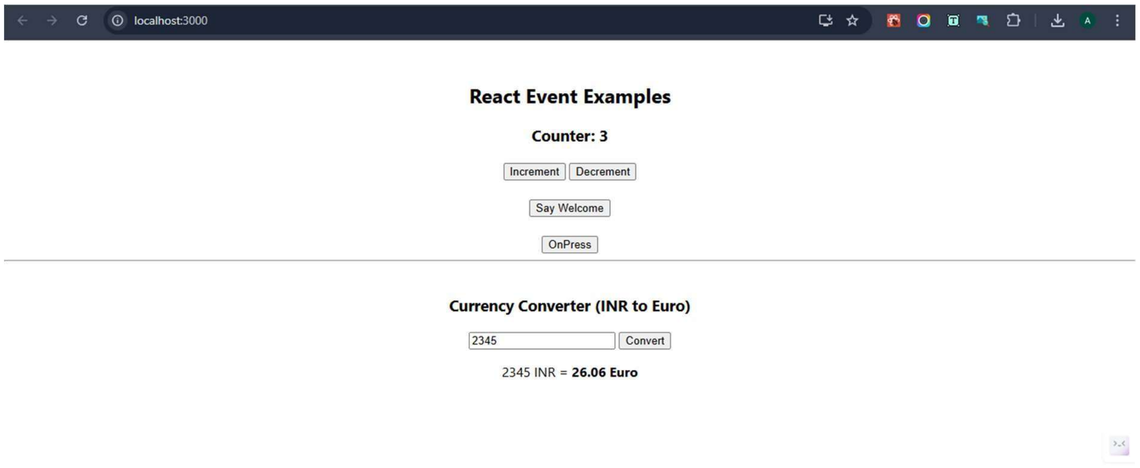
localhost:3000 says

Welcome!

Currency Converter (INR to Euro)

2345 Convert

2345 INR = **26.06 Euro**



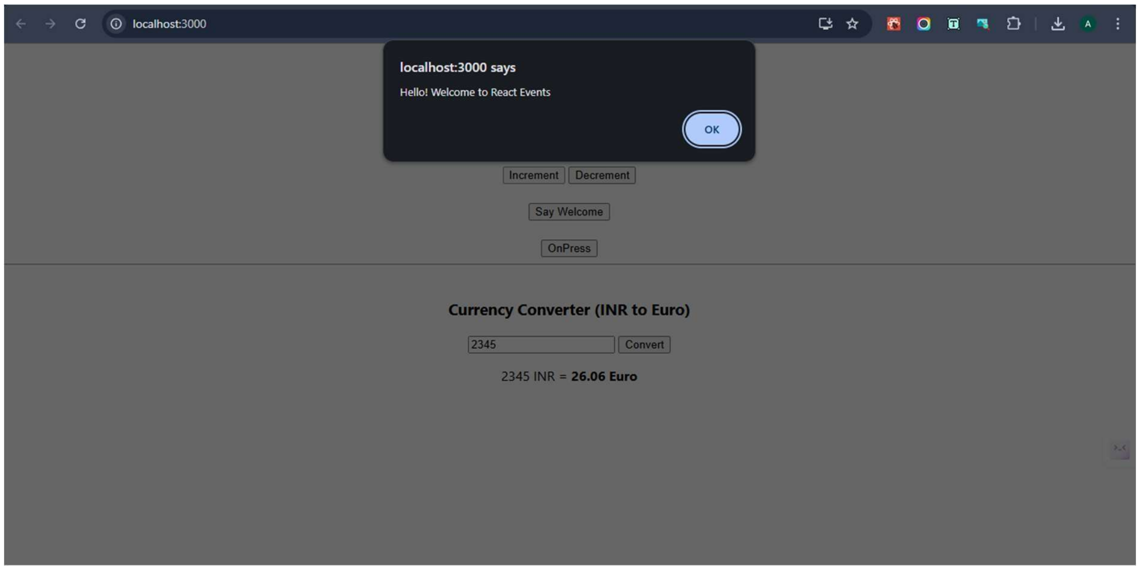
React Event Examples

Counter: 3

Currency Converter (INR to Euro)

2345 Convert

2345 INR = **26.06 Euro**



localhost:3000

localhost:3000 says

Hello! Welcome to React Events

Currency Converter (INR to Euro)

2345

2345 INR = **26.06 Euro**

HANDSON - 4

Ques on:

Create a React Applica on named “ cketbookingapp” where:

- Guest users can browse the page to view flight details.
- Only logged-in users can book ckets.
- Login and Logout bu ons should toggle between:

  o Guest page (for browsing flights) o

  User page (for booking ckets)

Objec ves:

- Explain about condi onal rendering in React
- Define element variables
- Explain how to prevent components from rendering

In this hands-on lab, you will learn how to:

- Implement condi onal rendering in React applica ons

Prerequisites:

The following is required to complete this hands-on lab:

- Node.js
- NPM
- Visual Studio Code

Notes:

Es mated me to complete this lab: 60 minutes

Steps:

1. Install Node.js and NPM from h ps://nodejs.org/en/download/
2. Install create-react-app globally:

bash

CopyEdit

npm install -g create-react-app

3. Create a new React applica on:

bash CopyEdit npx create-react-app

cketbookingapp

4. Navigate into the app folder:

bash CopyEdit cd

cketbookingapp

5. Open the folder in Visual Studio Code:

bash

CopyEdit

code .

6. Go to src/App.js, remove the exis ng content, and replace with the following code:

Code:

jsx

```
CopyEdit import React, { useState }
from 'react';

func on App() {   const [isLoggedIn,
setIsLoggedIn] = useState(false);

  const login = () => setIsLoggedIn(true);
const logout = () => setIsLoggedIn(false);

  const GuestPage = () => (
    <div>
      <h2>Welcome Guest!</h2>
      <p>Here are some flight details:</p>
      <ul>
```

```
        <li>Flight A - 10:00 AM</li>
        <li>Flight B - 02:00 PM</li>
      </ul>
      <bu on onClick={login}>Login</bu on>
    </div>
  );

  const UserPage = () => (
    <div>
      <h2>Welcome User!</h2>
      <p>You can now book ckets.</p>
      <bu on onClick={logout}>Logout</bu on>
    </div>
  );

  return (
    <div>
      <h1>Ticket Booking App</h1>
      {isLoggedIn ? <UserPage /> : <GuestPage />}
    </div>
  );
}

export default App;
```

Run the Applica on:

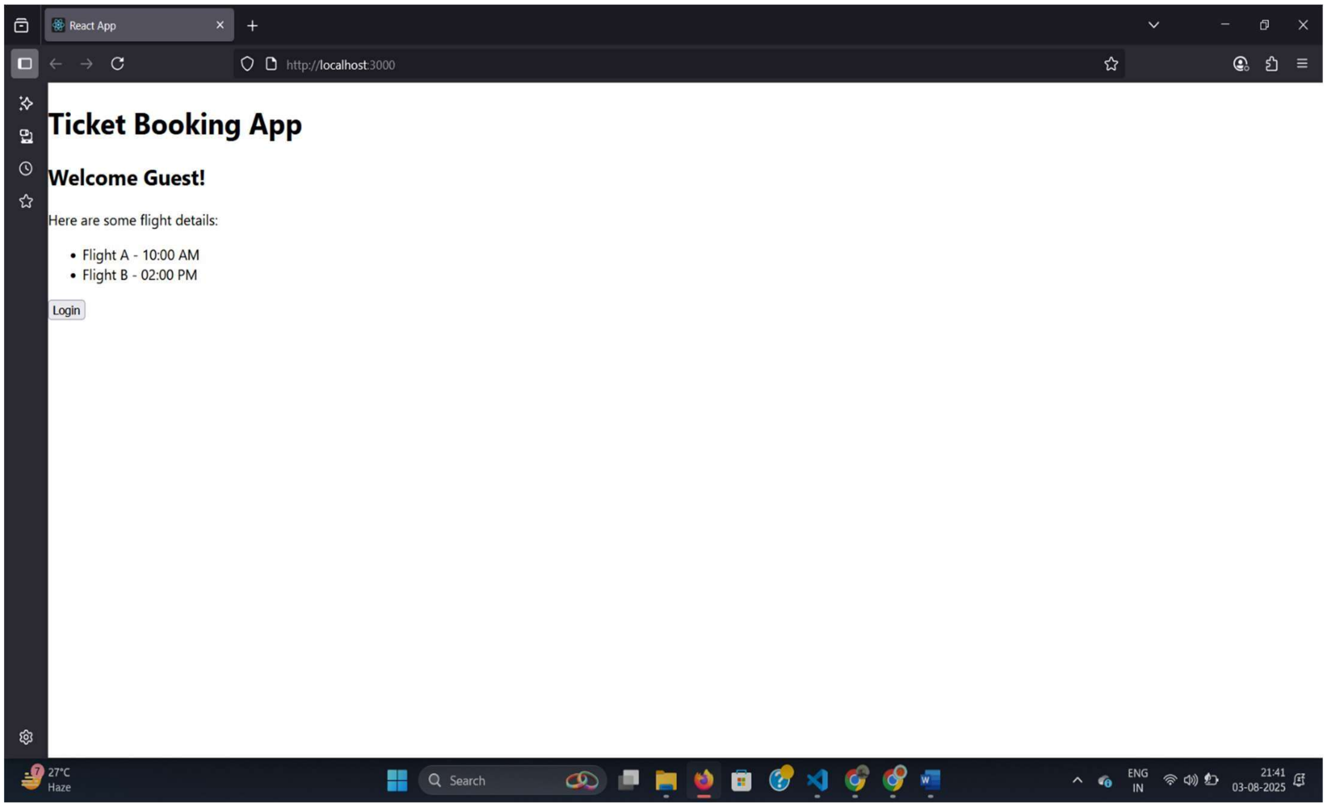
bash

CopyEdit

```
npm start
```

Output:

- When the app loads, GuestPage is shown by default.
- Clicking Login switches to UserPage.
- Clicking Logout returns to GuestPage.



Ticket Booking App

Welcome Guest!

Here are some flight details:

- Flight A - 10:00 AM
- Flight B - 02:00 PM

Login

HANDSON - 5

Ques on:

Create a React App named “bloggerapp” with 3 components:

- BookDetails
- BlogDetails
- CourseDetails

Implement the components using as many ways of Condi onal Rendering as possible.

Objec ves:

- Explain various ways of condi onal rendering

- Explain how to render mul ple components
- Define list component
- Explain about keys in React applica ons
- Explain how to extract components with keys
- Explain React Map, map() func on

In this hands-on lab, you will learn how to:

- Implement condi onal rendering in React applica ons

Prerequisites:

The following is required to complete this hands-on lab:

- Node.js
- NPM
- Visual Studio Code

Notes:

Es mated me to complete this lab: 60 minutes

胦 Step-by-Step Setup

Step 1: Open a New Folder and Terminal

- Create and open a folder named handson13 in VS Code
- Open terminal: Terminal > New Terminal

Step 2: Create React App in Current

Folder bash CopyEdit npx create-react-

app .

This installs React in your current folder (handson13)

Step 3: Run the Applica

on bash CopyEdit

npm start

This opens the default app at h p://localhost:3000

Step 4: Open and Replace App.js

Go to src/App.js and replace its content with:

jsx

CopyEdit import React, { useState }

from 'react';

func on BookDetails() {   return <p>This is the

Book Details component.</p>;

}

func on BlogDetails() {   return <p>This is the Blog

Details component.</p>;

}

func on CourseDetails() {   return <p>This is the

Course Details component.</p>; }

func on App() {   const [ac ve, setAc ve] =

useState("book");

// Way 1: If-Else   let

componentToRender;   if (ac ve ===

"book") {   componentToRender =

<BookDetails />;

```
  } else if (ac ve === "blog") {
componentToRender = <BlogDetails />;
  } else {
    componentToRender = <CourseDetails />;
  }

  // Way 2: Ternary
const heading =
ac ve === "book"
? "Book Sec on"
      : ac ve === "blog"
      ? "Blog Sec on"
      : "Course Sec on";

  // Way 3: Logical AND   const showFooter = ac ve === "course" &&
<p>Thanks for viewing courses!</p>;

  return (
    <div>
      <h1>{heading}</h1>

      <div>
        <bu on onClick={() => setAc ve("book")}>Book</bu on>
        <bu on onClick={() => setAc ve("blog")}>Blog</bu on>
                on  onClick={()  =>  setAc  ve("course")}>Course</bu
        <bu on>
      </div>

      {componentToRender}
      {showFooter}
    </div>
  );
}

export default App;
```

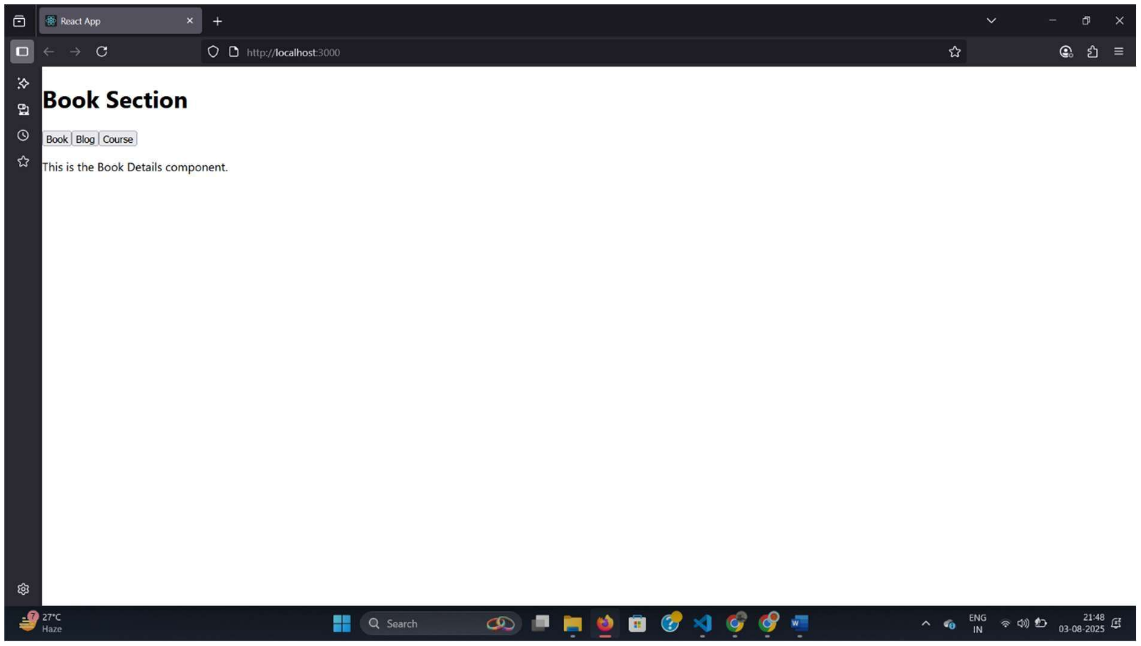
Output:

- Ini ally shows BookDetails with tle “Book Sec on”
- Clicking Blog switches to BlogDetails
- Clicking Course shows CourseDetails and a thank-you footer
- All 3 methods of condi onal rendering are used:

  o if-else o ternary

  operator o logical

  AND (&&)



Book Section

Book | Blog | Course

This is the Book Details component.