**WEEK- 7**

**9.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**
   2. **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.**

**Ans:**

**1. List the features of ES6:**ES6 introduced let, const, arrow functions, classes, template literals, promises, destructuring, and modules.

**2. Explain JavaScript let:**let declares a block-scoped variable that can be reassigned.

**3. Identify the differences between var and let:**var is function-scoped and hoisted, while let is block-scoped and not hoisted the same way.

**4. Explain JavaScript const:**const declares a block-scoped constant that cannot be reassigned.

**5. Explain ES6 class fundamentals:**ES6 classes provide a clearer syntax for creating objects and handling inheritance in JavaScript.

**6. Explain ES6 class inheritance:**Classes can inherit from other classes using the extends keyword and call parent constructors using super().

**7. Define ES6 arrow functions:**Arrow functions are a shorter syntax for functions and do not bind their own this.

**8. Identify set(), map():**Set stores unique values, while Map stores key-value pairs with any type of key.

**App.js**

import React from "react";

import ListofPlayers from "./components/ListofPlayers";

import Scorebelow70 from "./components/Scorebelow70";

import OddPlayers from "./components/OddPlayers";

import EvenPlayers from "./components/EvenPlayers";

import ListofIndianPlayers from "./components/ListofIndianPlayers";

export default function App() {

const flag = false; // change to false to see other view

const players = [

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

{ name: "Rohit Sharma", score: 65 },

{ name: "Shubman Gill", score: 90 },

{ name: "KL Rahul", score: 45 },

{ name: "Rishabh Pant", score: 75 },

{ name: "Hardik Pandya", score: 55 },

{ name: "Ravindra Jadeja", score: 88 },

{ name: "Mohammed Shami", score: 60 },

{ name: "Jasprit Bumrah", score: 95 },

{ name: "Bhuvneshwar Kumar", score: 40 },

{ name: "Yuzvendra Chahal", score: 72 }

];

const IndianTeam = ["Virat", "Rohit", "Gill", "Rahul", "Pant", "Hardik"];

const IndianPlayers = ["Kohli", "Rohit", "Pant"];

if (flag === true) {

return (

<div>

<h1>List of Players</h1>

<ListofPlayers players={players} />

<hr />

<h1>List of Players having Scores Less than 70</h1>

<Scorebelow70 players={players} />

</div>

);

} else {

return (

<div>

<h1>Indian Team</h1>

<h2>Odd Players</h2>

<OddPlayers team={IndianTeam} />

<hr />

<h2>Even Players</h2>

<EvenPlayers team={IndianTeam} />

<hr />

<h2>List of Indian Players Merged:</h2>

<ListofIndianPlayers IndianPlayers={IndianPlayers} />

</div>

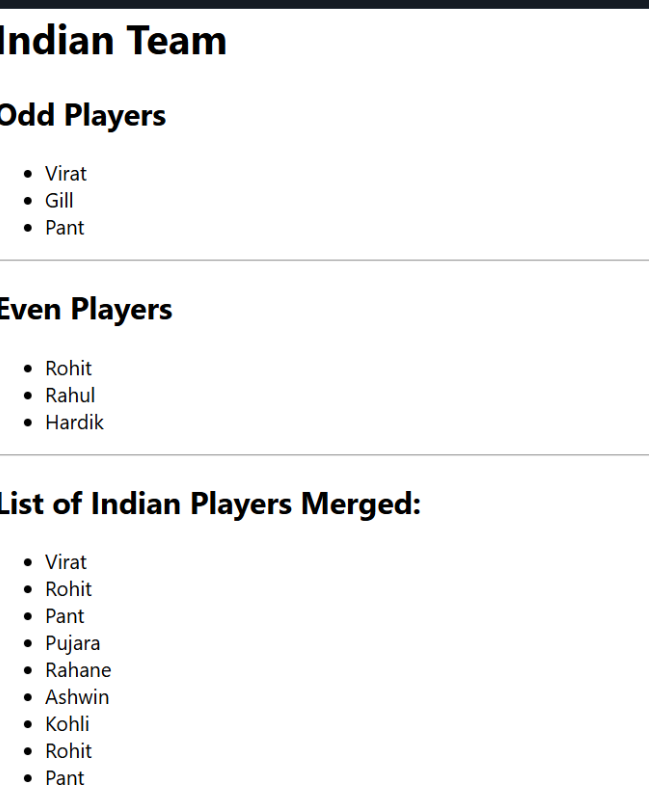
);

}

}

**Output**

**Flag==false**



Flag==true



**10.**

**Create a React Application named “officespacerentalapp” which uses React JSX to create elements, attributes and renders DOM to display the page.**

**1. Define JSX:**JSX is a syntax extension that lets you write HTML-like code inside JavaScript for React.

**2. Explain about ECMA Script:**ECMAScript is the standardized specification that defines JavaScript's syntax and features.

**3. Explain React.createElement():**React.createElement() is a method to create virtual DOM elements in React.

**4. Explain how to create React nodes with JSX:**React nodes can be created using HTML-like tags directly in JSX.

**5. Define how to render JSX to DOM:**JSX is rendered to the DOM using ReactDOM.render() method.

**6. Explain how to use JavaScript expressions in JSX:**JavaScript expressions are embedded in JSX using curly braces {}.

**7. Explain how to use inline CSS in JSX:**Inline CSS in JSX is applied using a JavaScript object with camelCase style properties.

**APP.js**

import React from "react";

import "./App.css";

import imgDBS from "./images/dbs.jpg";

import imgRegus from "./images/regus.jpg";

import imgWeWork from "./images/wework.jpg";

function App() {

const element = "Office Space";

const officeList = [

{ Name: "DBS", Rent: 50000, Address: "Chennai", img: imgDBS },

{ Name: "Regus", Rent: 75000, Address: "Bangalore", img: imgRegus },

{ Name: "WeWork", Rent: 60000, Address: "Mumbai", img: imgWeWork }

];

return (

<div>

<h1>{element}, at Affordable Range</h1>

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

// Rent color logic

let colors = [];

if (office.Rent <= 60000) {

colors.push("textRed");

} else {

colors.push("textGreen");

}

return (

<div key={index}>

<img src={office.img} width="25%" height="25%" alt={office.Name} />

<h1>Name: {office.Name}</h1>

<h3 className={colors.join(" ")}>Rent: Rs. {office.Rent}</h3>

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

<hr />

</div>

);

})}

</div>

);

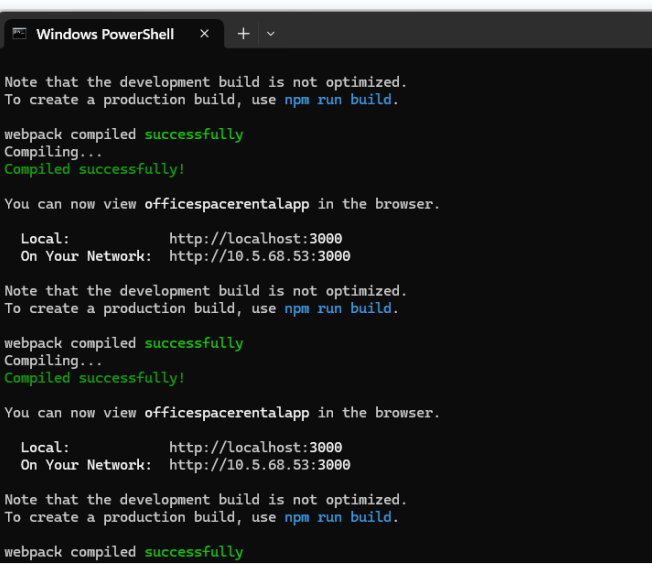
}

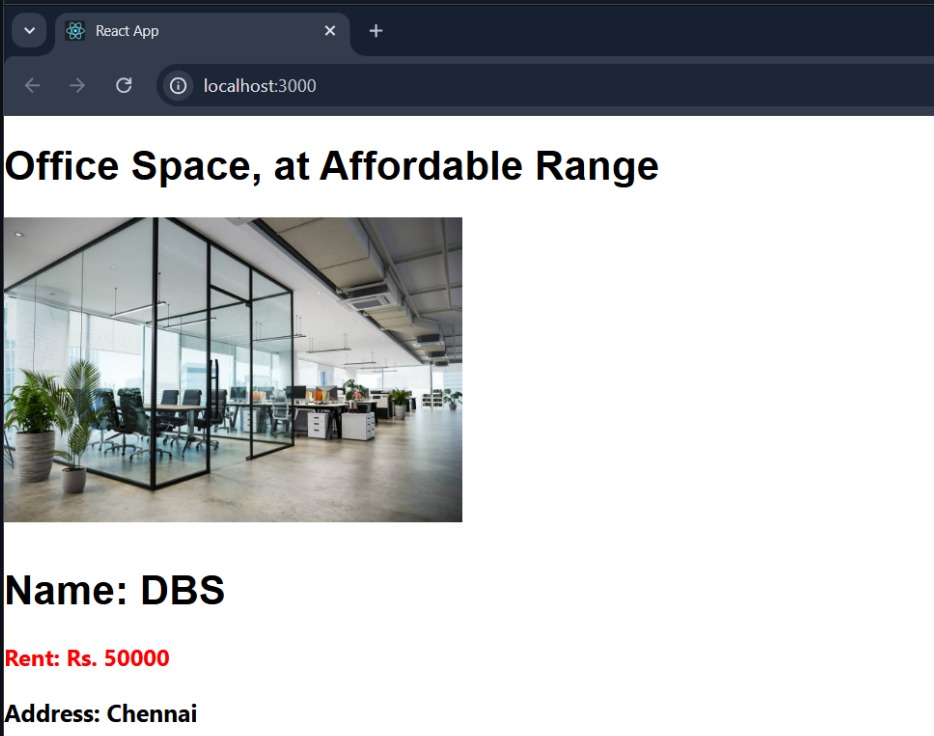
export default App;

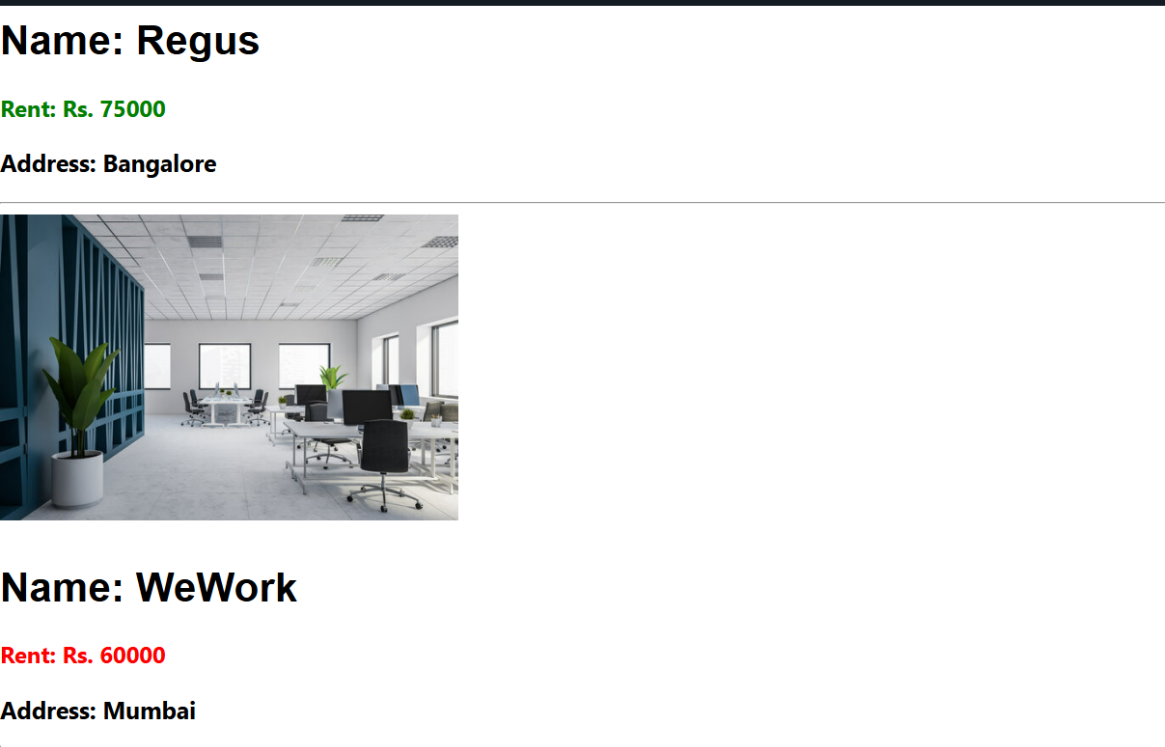
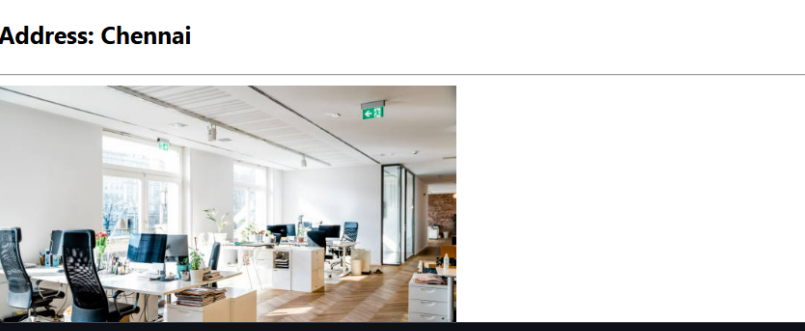
**App.test.js**



OUTPUT







**11.**

**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.**
2. **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.**

**1. Explain React events:**  
React events are objects that handle user interactions like clicks or form inputs in a React component.

**2. Explain about event handlers:**  
Event handlers are functions defined in React to respond to specific events like onClick or onChange.

**3. Define Synthetic event:**  
A SyntheticEvent is a cross-browser wrapper around the native browser event in React.

**4. Identify React event naming convention:**  
React event names use camelCase (e.g., onClick, onMouseOver) instead of lowercase.

**App.js**

import React, { useState } from "react";

import CurrencyConvertor from "./CurrencyConvertor";

export default function App() {

const [count, setCount] = useState(1);

const increment = () => {

setCount(count + 1);

};

const sayHello = () => {

alert("Hello! Member1");

};

const handleIncrementClick = () => {

increment();

sayHello();

};

const decrement = () => {

setCount(count - 1);

};

const sayWelcome = (msg) => {

alert(msg);

};

const handleClickEvent = () => {

alert("I was clicked");

};

return (

<div>

<p>$</p>

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

<br />

<button onClick={decrement}>Decrement</button>

<br />

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

<br />

<button onClick={handleClickEvent}>Click on me</button>

<br />

<CurrencyConvertor />

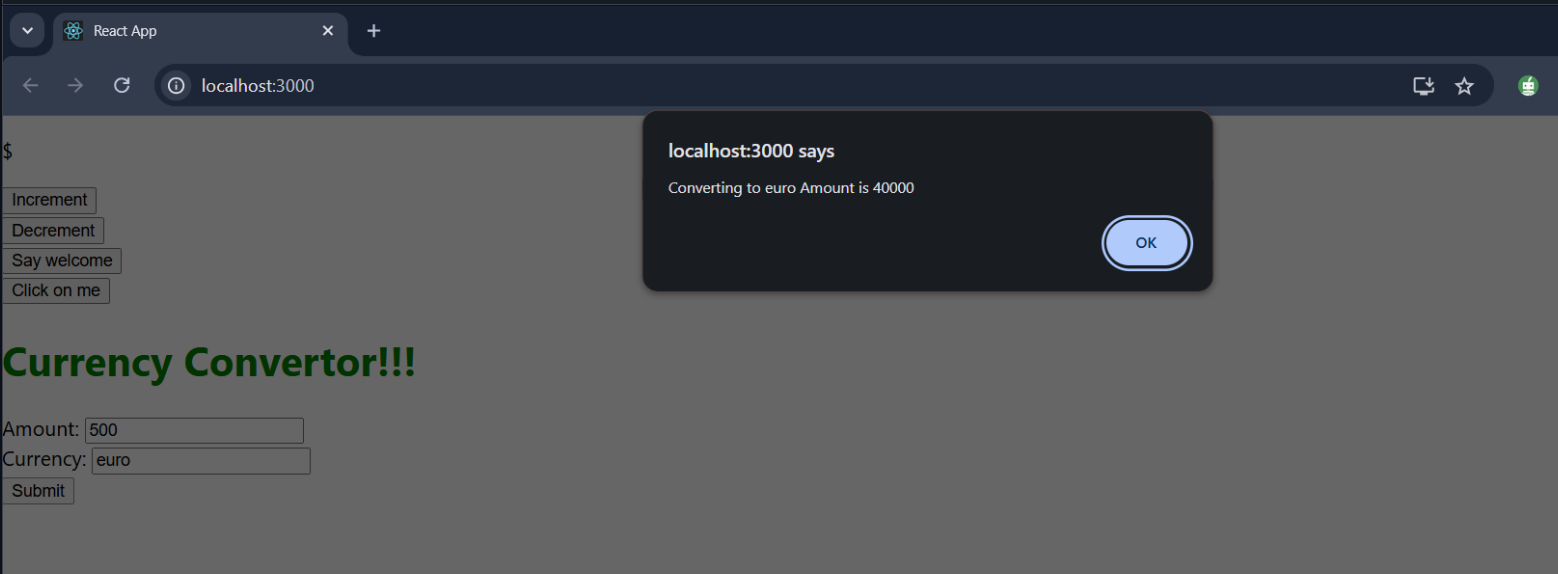
</div>

);

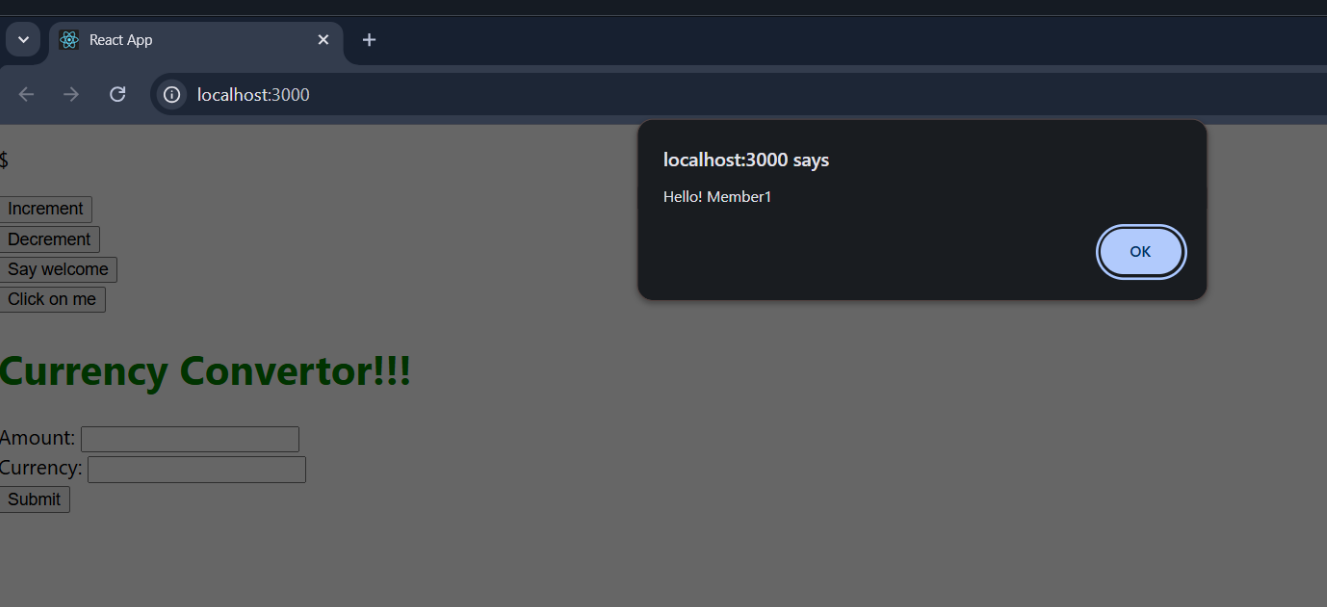
}

**OUTPUT**

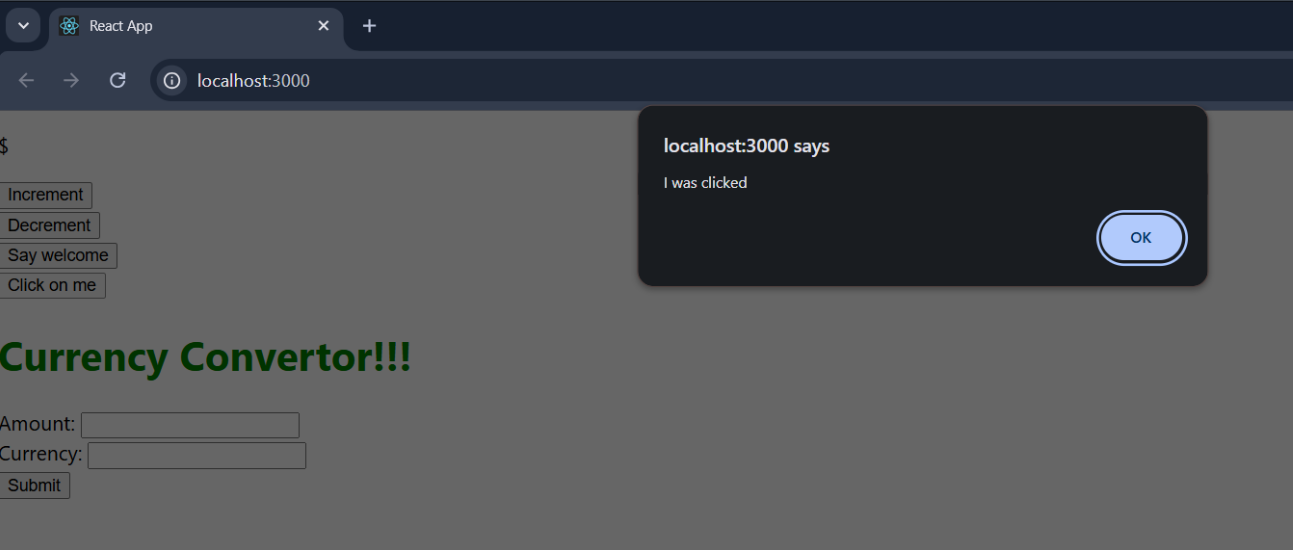
Curreny converter



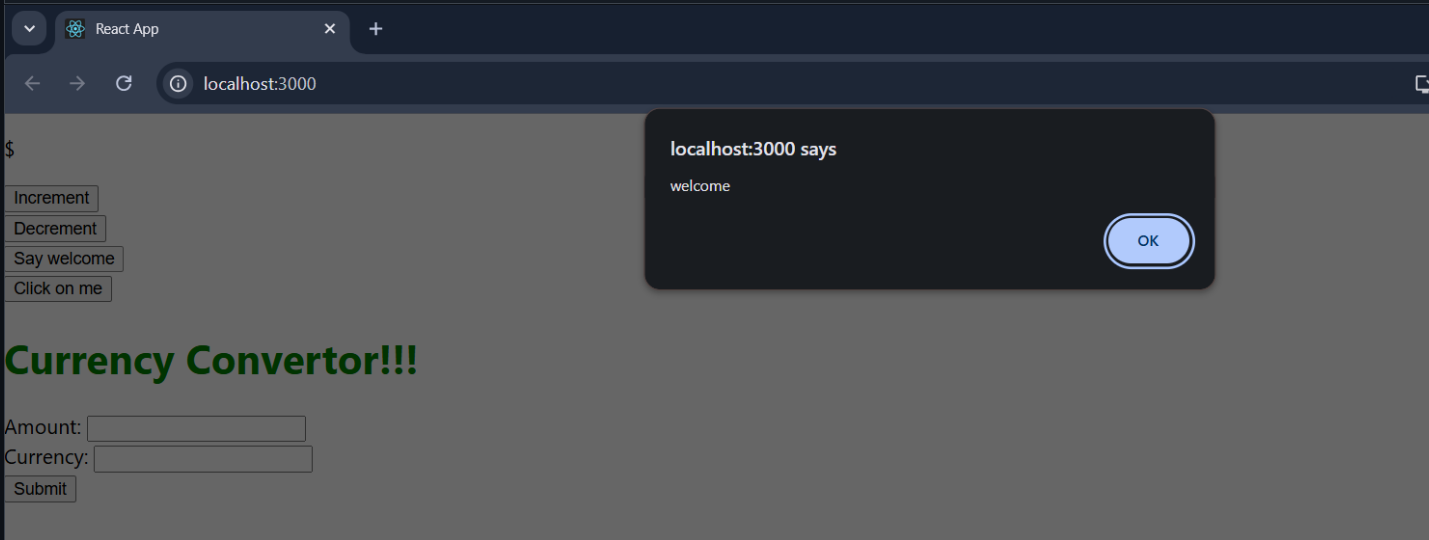
Increment-decrement



On-press



welcome



**12.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.**

**Ans:**

**Explain about conditional rendering in React:**  
Conditional rendering in React means showing components or elements based on certain conditions using JavaScript logic.

**Define element variables:**  
Element variables are used to store JSX elements in a variable to render them conditionally or dynamically.

**Explain how to prevent components from rendering:**  
Components can be prevented from rendering by returning null or using conditional statements.

**App.js**

import React, { useState } from "react";

import LoginButton from "./LoginButton";

import LogoutButton from "./LogoutButton";

import Greeting from "./Greeting";

export default function App() {

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

const handleLoginClick = () => {

setIsLoggedIn(true);

};

const handleLogoutClick = () => {

setIsLoggedIn(false);

};

let button;

if (isLoggedIn) {

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

} else {

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

}

return (

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

<Greeting isLoggedIn={isLoggedIn} />

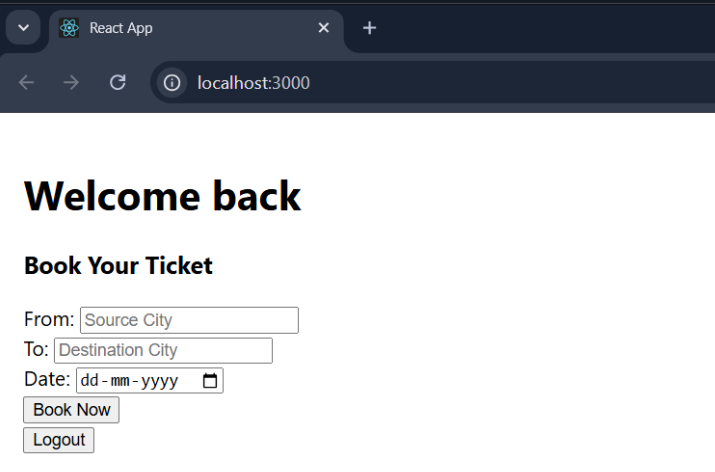
{button}

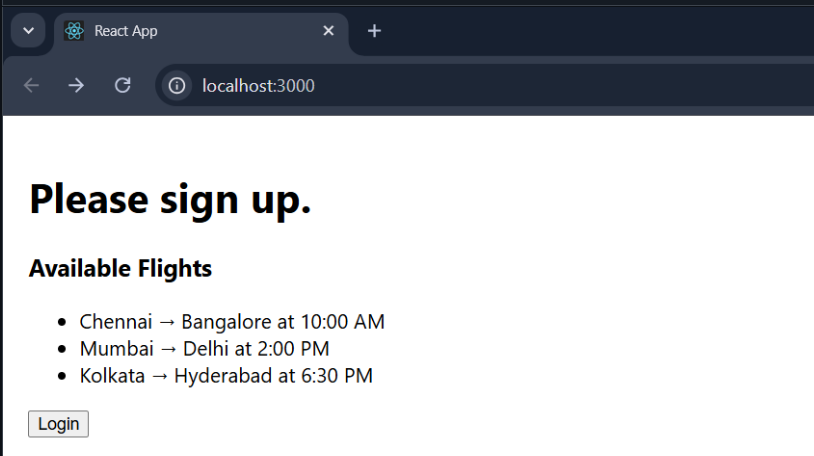
</div>

);

}

**OUTPUT**





**13.**

**Create a React App named “bloggerapp” in with 3 components.**

1. **Book Details**
2. **Blog Details**
3. **Course Details**

**Ans:**

**Explain various ways of conditional rendering:**  
Conditional rendering can be done using if-else, ternary operator, logical &&, or element variables.

**Explain how to render multiple components:**  
Multiple components can be rendered by including them inside a parent component or using fragments (<> </>).

**Define list component:**  
A list component in React renders a collection of similar elements using arrays and the map() function.

**Explain about keys in React applications:**  
Keys are unique identifiers used to help React efficiently update and manage list items.

**Explain how to extract components with keys:**  
You can extract list items into separate components and pass a unique key prop to each instance.

**Explain React Map,** map() **function:**  
The map() function in React is used to iterate over arrays and return a list of elements for rendering.

**App.js**

import React from "react";

import "./App.css";

import BookDetails from "./BookDetails";

import BlogDetails from "./BlogDetails";

import CourseDetails from "./CourseDetails";

export default function App() {

const books = [

{ id: 1, bname: "Master React", price: 670 },

{ id: 2, bname: "Deep Dive into Angular 11", price: 800 },

{ id: 3, bname: "Mongo Essentials", price: 450 }

];

const blogs = [

{

id: 1,

title: "React Learning",

author: "Stephen Biz",

content: "Welcome to learning React!"

},

{

id: 2,

title: "Installation",

author: "Schwezdneier",

content: "You can install React from npm."

}

];

const courses = [

{ id: 1, name: "Angular", date: "4/5/2021" },

{ id: 2, name: "React", date: "6/3/2021" }

];

return (

<div style={{ display: "flex", justifyContent: "space-around" }}>

<BookDetails books={books} />

<BlogDetails blogs={blogs} />

<CourseDetails courses={courses} />

</div>

);

}

**BlogDetails.js**

import React from "react";

export default function BlogDetails(props) {

// Conditional Rendering (Ternary Operator)

return (

<div className="v1">

<h1>Blog Details</h1>

{props.blogs.length > 0 ? (

props.blogs.map((blog) => (

<div key={blog.id}>

<h2>{blog.title}</h2>

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

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

</div>

))

) : (

<p>No blogs available.</p>

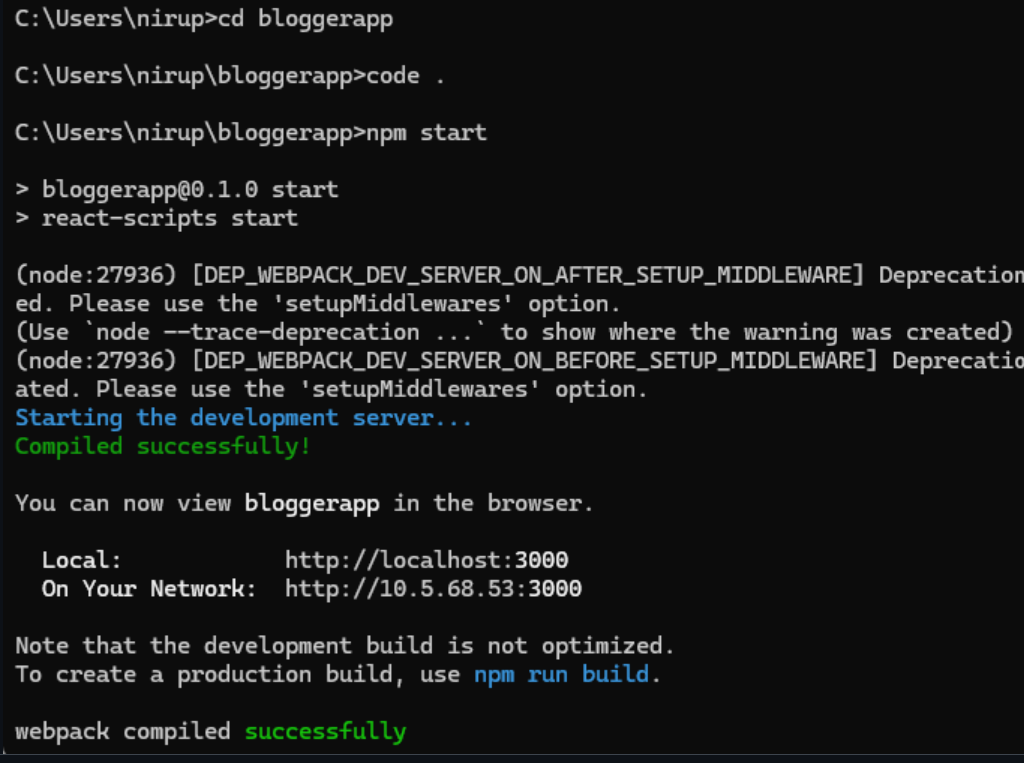
)}

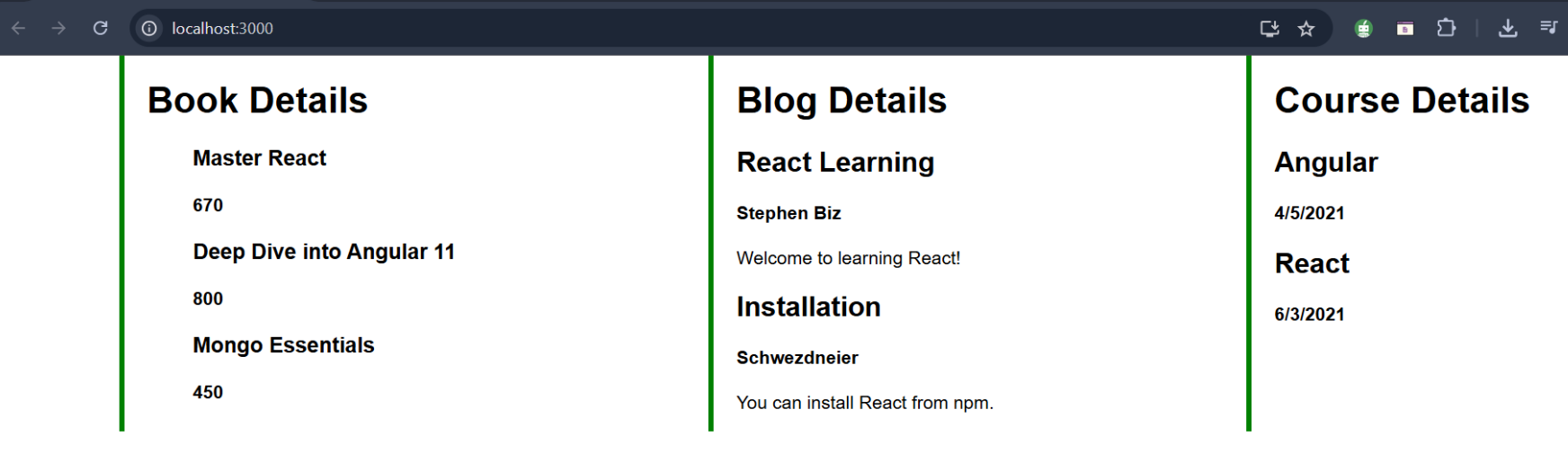
</div>

);

}

**OUTPUT**





**App.js**



**OUTPUT**

