**Week 7 - React - Hands-on**

DEBANJAN KAURI(Superset ID-6362388)

**Mandatory Hands-on**

**Question 1: Create a React Application named “cricketapp”**

Scenario:

Create cricketapp React app using ES6 features.

To achieve this:

1. Add ListofPlayers component:

* Use map() for 11 players with name & score.
* Use arrow function to filter score < 70.

2. Add IndianPlayers component:

* Use destructuring for odd/even teams.
* Merge T20players & RanjiTrophyPlayers using spread.

3. Use if-else with flag to show both components on homepage.

### **Code:**

*ListofPlayers.js*

import React from 'react';

function ListofPlayers() {

const players = [

{ name: "Rohit", score: 89 },

{ name: "Kohli", score: 45 },

{ name: "Gill", score: 76 },

{ name: "SKY", score: 50 },

{ name: "Hardik", score: 30 },

{ name: "Rahul", score: 81 },

{ name: "Iyer", score: 66 },

{ name: "Samson", score: 25 },

{ name: "Pant", score: 71 },

{ name: "Bumrah", score: 55 },

{ name: "Shami", score: 90 }

];

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

return (

<div>

<h2>List of Players</h2>

<ul>

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

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

))}

</ul>

<h2>List of Players (score ≥ 70)</h2>

<ul>

{filteredPlayers.map((p, index) => (

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

))}

</ul>

</div>

);

}

export default ListofPlayers;

*IndianPlayers.js*

import React from 'react';

function IndianPlayers() {

const team = ["Rohit", "Kohli", "Gill", "SKY", "Hardik", "Rahul"];

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

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

const T20players = ["Bumrah", "Shami", "Chahal"];

const RanjiTrophy = ["Pujara", "Rahane"];

const merged = [...T20players, ...RanjiTrophy];

return (

<div>

<h2>Indian Players</h2>

<h3>Odd Team:</h3>

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

<h3>Even Team:</h3>

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

<h3>Merged Team (T20 + Ranji Trophy):</h3>

<ul>{merged.map((name, index) => <li key={index}>{name}</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>Cricket App</h1>

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

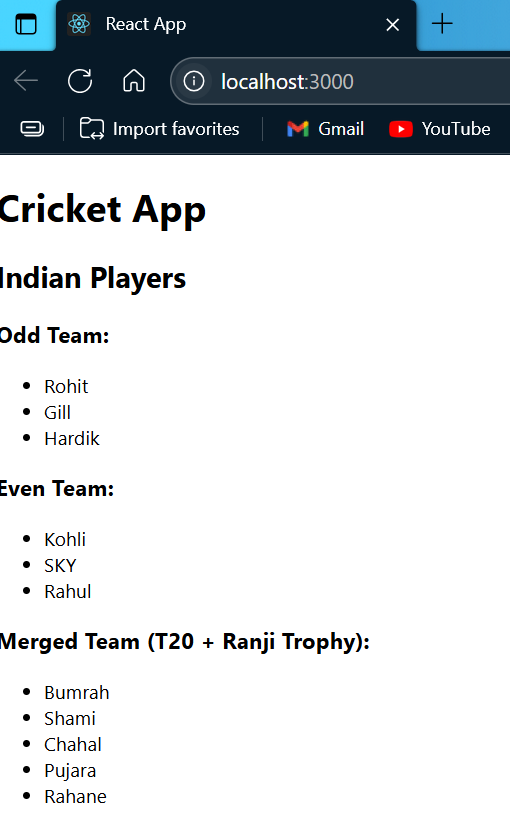
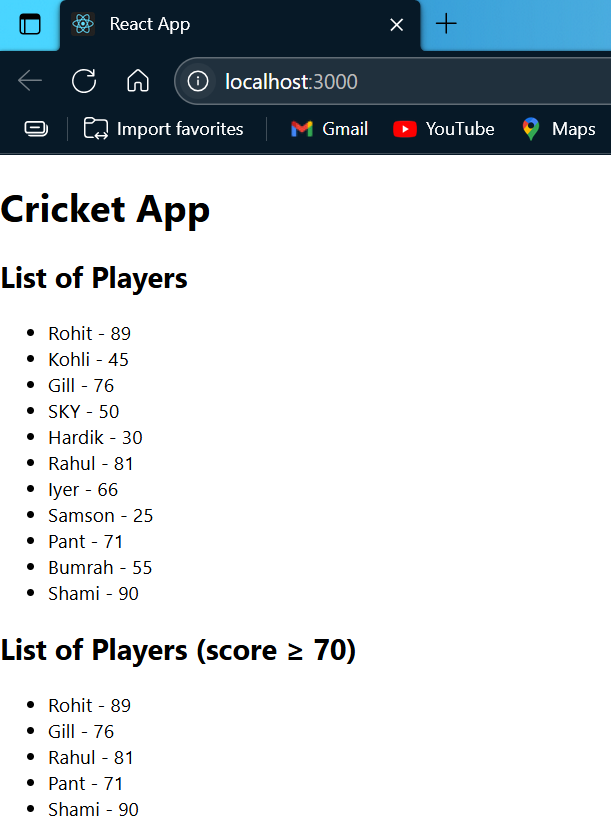
</div>

);

}

export default App;

### **Output:**



Flag=true

Flag=false

**Question 2: Create a React Application named “officespacerentalapp”**

Scenario:

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

To achieve this:

1. Create an element to display the heading of the page.

2. Attribute to display the image of the office space.

3. Create an object of office to display the details like Name, Rent and Address.

4. Create a list of Object and loop through the office space item to display more data.

5. To apply CSS, Display the color of the Rent in Red if it’s below 60000 and in Green if it’s above 60000.

### **Code:** *App.js*

import React from 'react';

const offices = [

{

name: "Corporate Hub",

rent: 45000,

address: "Sector 21, New Delhi",

image: "https://images.unsplash.com/39/lIZrwvbeRuuzqOoWJUEn\_Photoaday\_CSD%20%281%20of%201%29-5.jpg?q=80&w=2070&auto=format&fit=crop&ixlib=rb-4.1.0&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D"

},

{

name: "Urban Workspace",

rent: 75000,

address: "Bandra East, Mumbai",

image: "https://images.unsplash.com/photo-1751151015781-87f5c01e6f1c?q=80&w=1074&auto=format&fit=crop&ixlib=rb-4.1.0&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D"

},

{

name: "Tech Park",

rent: 60000,

address: "Electronic City, Bangalore",

image: "https://images.unsplash.com/photo-1733664602980-5052fc24f9a4?q=80&w=1074&auto=format&fit=crop&ixlib=rb-4.1.0&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D"

}

];

function getRentStyle(rent) {

return {

color: rent >= 60000 ? "green" : "red",

fontWeight: "bold"

};

}

function App() {

return (

<div style={{ padding: '30px', fontFamily: 'Arial' }}>

<h1>Office Space Rentals</h1>

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

<div key={index} style={{

border: "1px solid #ccc",

padding: "15px",

margin: "10px 0",

borderRadius: "10px",

backgroundColor: "#f9f9f9"

}}>

<img src={office.image} alt={office.name} width="200" />

<h2>{office.name}</h2>

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

<p>

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

</p>

</div>

))}

</div>

);

}

export default App;

### **Output:**

**Question 3: Create a React Application named “eventexamplesapp”**

Scenario:

Create a React Application “eventexamplesapp” to handle various events of the form elements in HTML.

To achieve this:

1. Add Increment and Decrement buttons:

* Increment calls multiple methods: increment counter + show “Hello” message.
* Decrement decreases counter value.

2. Add Say Welcome button that takes "welcome" as an argument in function

3. Add button to handle synthetic event onClick → show “I was clicked”.

4. Create CurrencyConvertor component:

* Input in INR, on Convert button click, convert to Euro.
* Use handleSubmit to perform conversion.

### **Code:**

*Counter.js*

import React, { Component } from "react";

class Counter extends Component {

constructor(props) {

super(props);

this.state = {

count: 0

};

}

increment = () => {

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

this.sayHello();

};

decrement = () => {

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

};

sayHello = () => {

alert("Hello from Cognizant React Lab!");

};

render() {

return (

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

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

<button onClick={this.increment}>Increment</button>{" "}

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

</div>

);

}

}

export default Counter;

*Welcome.js*

import React from 'react';

function Welcome() {

const sayWelcome = (msg) => {

alert("Message: " + msg);

};

return (

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

<button onClick={() => sayWelcome("Welcome to React!")}>

Say Welcome

</button>

</div>

);

}

export default Welcome;

*ClickMessage.js*

import React from 'react';

function ClickMessage() {

const handleClick = (e) => {

e.preventDefault();

alert("I was clicked");

};

return (

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

<button onClick={handleClick}>Click Me</button>

</div>

);

}

export default ClickMessage;

*CurrencyConverter.js*

import React, { useState } from "react";

function CurrencyConverter() {

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

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

const handleSubmit = (e) => {

e.preventDefault();

const euro = parseFloat(rupees) \* 0.011; // sample conversion rate

setEuros(euro.toFixed(2));

};

return (

<div>

<h2 style={{ padding: '30px', fontFamily: 'Arial' , color:'green'}}>Currency Converter</h2>

<form onSubmit={handleSubmit}>

<label>Rupees: </label>

<input

type="number"

value={rupees}

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

required

/><br /><br />

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

</form>

{euros && (

<p>

<strong>Euros:</strong> € {euros}

</p>

)}

</div>

);

}

export default CurrencyConverter;

*App.js*

import React from "react";

import Counter from "./Counter";

import Welcome from "./Welcome";

import ClickMessage from "./ClickMessage";

import CurrencyConverter from "./CurrencyConverter";

function App() {

return (

<div style={{ padding: "30px", fontFamily: "Arial" }}>

<h1>React Events & Synthetic Event Handling</h1>

<Counter />

<Welcome />

<ClickMessage />

<CurrencyConverter />

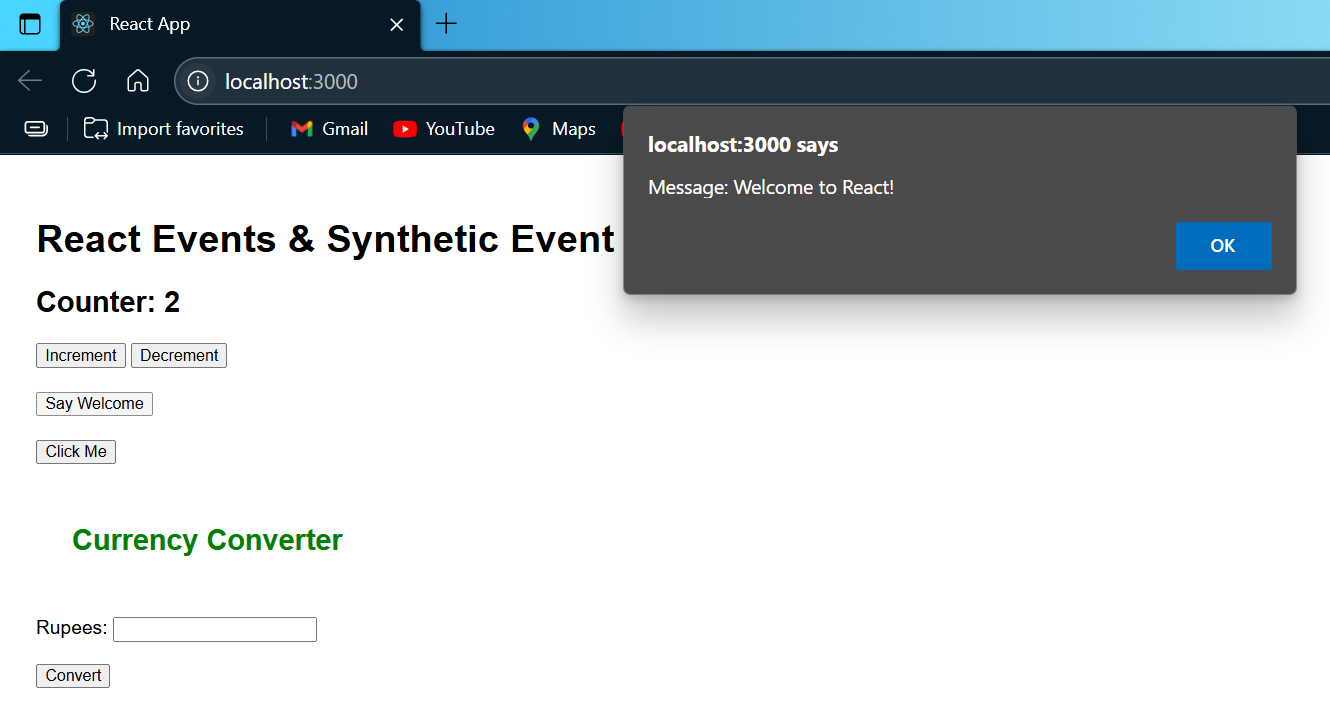
</div>

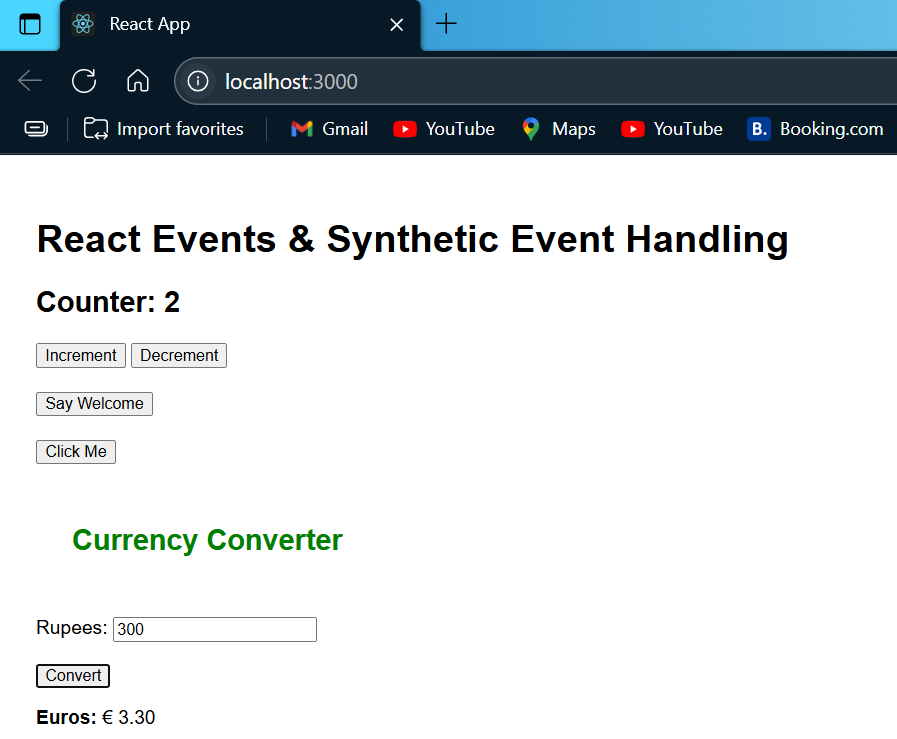
);

}

export default App;

### **Output:**





**Question 4: Create a React Application named “ticketbookingapp”**

Scenario:

Create React app ticketbookingapp with conditional views for guest and logged-in users.

To achieve this:

1. Show Guest Page by default with flight details (no booking access).

2. Add Login and Logout buttons.

3. On Login, show User Page with ticket booking access.

4. On Logout, switch back to Guest Page.

5. Use a flag/state to toggle between guest and user views.

**Code:**

*Guest.js*

import React from "react";

function Guest() {

return (

<div>

<h2>Welcome Guest!</h2>

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

</div>

);

}

export default Guest;

*User.js*

import React from "react";

function User() {

return (

<div>

<h2>Welcome Back!</h2>

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

<button>Book Now</button>

</div>

);

}

export default User;

*LoginControl.js*

import React, { useState } from "react";

import Guest from "./Guest";

import User from "./User";

function LoginControl() {

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

const handleLogin = () => {

setIsLoggedIn(true);

};

const handleLogout = () => {

setIsLoggedIn(false);

};

let content;

if (isLoggedIn) {

content = <User />;

} else {

content = <Guest />;

}

return (

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

<h1>Ticket Booking App</h1>

{isLoggedIn ? (

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

) : (

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

)}

<hr />

{content}

</div>

);

}

export default LoginControl;

*App.js*

import React from "react";

import LoginControl from "./LoginControl";

function App() {

return (

<div className="App">

<LoginControl />

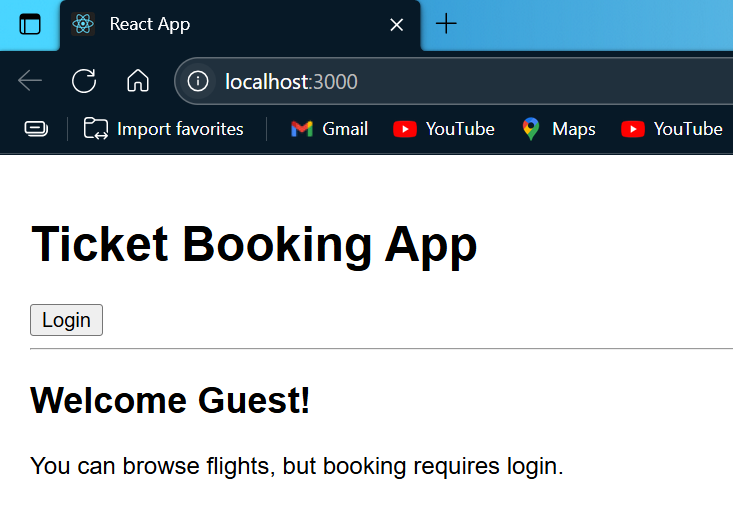
</div>

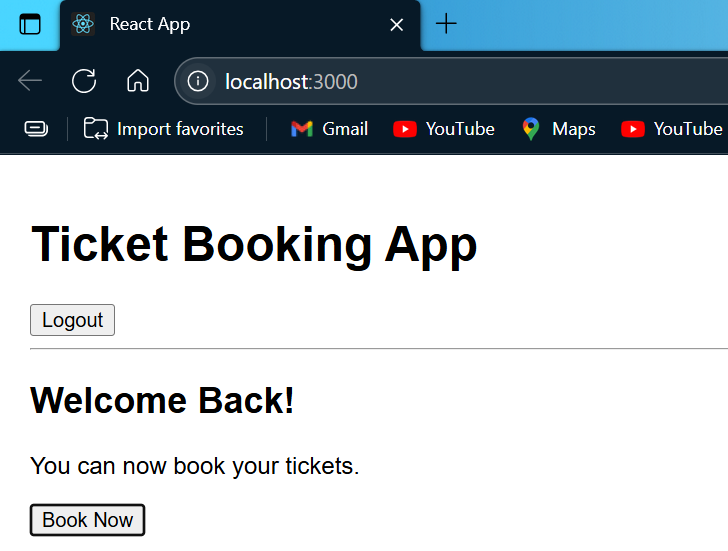
);

}

export default App;

### **Output:**





**Question 5: Create a React Application named “bloggerapp”**

Scenario:

Create a React app bloggerapp with 3 components using different conditional rendering methods.

To achieve this:

1. Create the 3 components in src.

2. In App.js, implement multiple conditional rendering techniques.

3. Use flags or state variables to switch between components.

4. Render the components based on conditions.

### **Code:**

*BlogDetails.js*

import React from "react";

function BlogDetails({ blogs }) {

return (

<div>

<h2>Blog List</h2>

{blogs.length > 0 ? (

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

<div key={index}>

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

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

<hr />

</div>

))

) : (

<p>No blogs found.</p>

)}

</div>

);

}

export default BlogDetails;

*BookDetails.js*

import React from "react";

function BookDetails({ books }) {

return (

<div>

<h2>Book List</h2>

<ul>

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

<li key={index}>

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

</li>

))}

</ul>

</div>

);

}

export default BookDetails;

*CourseDetails.js*

import React from "react";

function CourseDetails({ courses }) {

return (

<div>

<h2>Courses Offered</h2>

<ul>

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

<li key={index}>{course.name} – {course.duration}</li>

))}

</ul>

</div>

);

}

export default CourseDetails;

*App.js*

import React, { useState } from "react";

import BookDetails from "./BookDetails";

import BlogDetails from "./BlogDetails";

import CourseDetails from "./CourseDetails";

function App() {

const [view, setView] = useState("books");

const books = [

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

{ title: "Deep Work", author: "Cal Newport" }

];

const blogs = [

{ title: "React Tips", description: "Learn about React best practices" },

{ title: "Frontend Trends", description: "What's new in frontend?" }

];

const courses = [

{ name: "ReactJS", duration: "30 hours" },

{ name: "NodeJS", duration: "25 hours" }

];

return (

<div style={{ padding: "30px", fontFamily: "Arial" }}>

<h1>BloggerApp - Conditional Rendering</h1>

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

<button onClick={() => setView("books")}>Show Books</button>{" "}

<button onClick={() => setView("blogs")}>Show Blogs</button>{" "}

<button onClick={() => setView("courses")}>Show Courses</button>

</div>

{view === "books" && <BookDetails books={books} />}

{view === "blogs" && <BlogDetails blogs={blogs} />}

{view === "courses" && <CourseDetails courses={courses} />}

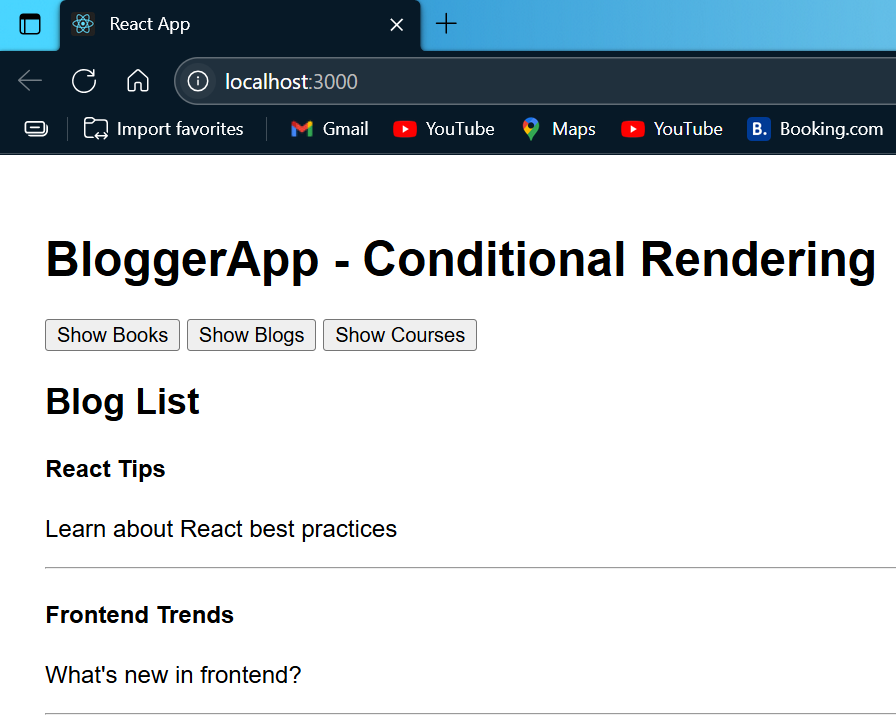
</div>

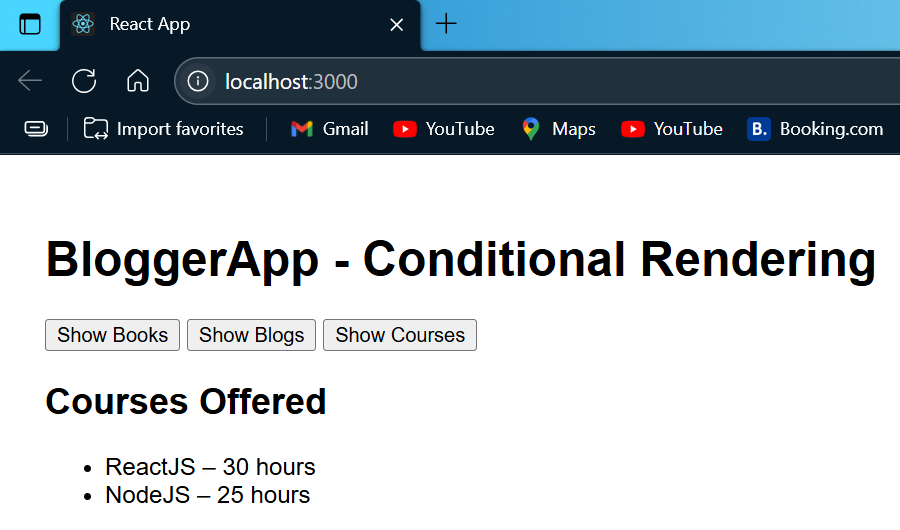
);

}

export default App;

### **Output:**





**Additional Hands-on**

**Question 6: Create a new React Employee Management App**

Scenario:

Create a React app bloggerapp with 3 components using different conditional rendering methods.

To achieve this:

1. Unzip employeesapp.zip

2. Review App.js, EmployeesList.js, and EmployeeCard.js.

3. Create ThemeContext.js:

* Define and export a context with default value "light"

4. In App.js:

* Import ThemeContext, wrap JSX with ThemeContext.Provider
* Use component state to provide theme value & Remove theme prop from EmployeeList.

5. In EmployeesList.js:

* Remove theme prop passing to EmployeeCard.

6. In EmployeeCard.js:

* Import ThemeContext and use useContext() to get theme.
* Apply theme as className for button styling.

### **Code:**

*ThemeContext.js*

import { createContext } from 'react';

const ThemeContext = createContext('light');

export default ThemeContext;

*EmployeesList.js*

import React from "react";

import EmployeeCard from "./EmployeeCard";

function EmployeesList() {

const employees = [

{ id: 1, name: "Alice", designation: "Engineer" },

{ id: 2, name: "Bob", designation: "Tester" },

{ id: 3, name: "Charlie", designation: "Manager" }

];

return (

<div>

<h2>Employee List</h2>

<div style={{

display: 'flex',

flexWrap: 'wrap',

gap: '20px',

justifyContent: 'center'

}}>

{employees.map(emp => (

<EmployeeCard key={emp.id} employee={emp} />

))}

</div>

</div>

);

}

export default EmployeesList;

*EmployeeCard.js*

import React, { useContext } from "react";

import ThemeContext from "./ThemeContext";

function EmployeeCard({ employee }) {

const theme = useContext(ThemeContext);

const buttonStyle = {

backgroundColor: theme === "light" ? "#e0e0e0" : "#333",

color: theme === "light" ? "#000" : "#fff",

padding: "8px 16px",

border: "none",

borderRadius: "5px",

marginTop: "5px"

};

return (

<div style={{

border: "1px solid #aaa",

padding: "10px",

borderRadius: "8px",

width: "250px",

boxShadow: "2px 2px 10px rgba(0,0,0,0.1)"

}}>

<h3>{employee.name}</h3>

<p>Designation: {employee.designation}</p>

<button style={buttonStyle}>View Profile</button>

</div>

);

}

export default EmployeeCard;

*App.js*

import React, { useState } from "react";

import EmployeesList from "./EmployeesList";

import ThemeContext from "./ThemeContext";

function App() {

const [theme, setTheme] = useState("light");

const toggleTheme = () => {

setTheme(prev => (prev === "light" ? "dark" : "light"));

};

return (

<ThemeContext.Provider value={theme}>

<div>

<h1>Employee Management App</h1>

<button onClick={toggleTheme}>Toggle Theme</button>

<EmployeesList />

</div>

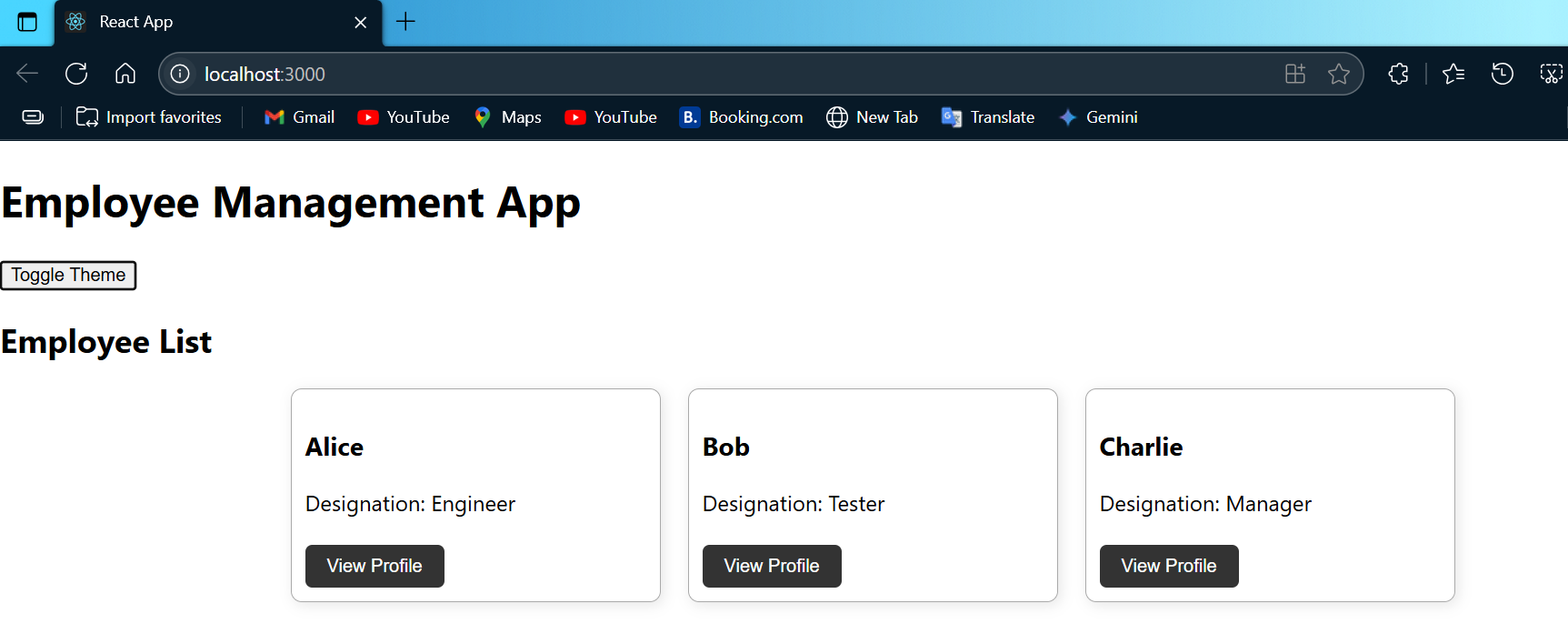
</ThemeContext.Provider>

);

}

export default App;

### **Output:**



**Question 7: Create a new React Application named “ticketraisingapp”**

Scenario:

Create a React app ticketraisingapp to register and submit complaints.

To achieve this:

4. Create ComplaintRegister component with:

* Textbox for employee name.
* Textarea for complaint.
* Submit button with handleSubmit.

5. On submit, generate a reference number and show it in an alert.

### **Code:**

*ComplaintRegister.js*

import React, { useState } from 'react';

function ComplaintRegister() {

const [name, setName] = useState('');

const [complaint, setComplaint] = useState('');

const handleSubmit = (e) => {

e.preventDefault();

if (name.trim() === '' || complaint.trim() === '') {

alert('Please fill all fields!');

return;

}

const referenceNumber = 'REF' + Math.floor(1000 + Math.random() \* 9000);

alert(`Complaint submitted!\nReference No: ${referenceNumber}`);

setName('');

setComplaint('');

};

return (

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

<h2>Complaint Register</h2>

<form onSubmit={handleSubmit}>

<div style={{ marginBottom: '10px' }}>

<label>Employee Name: </label><br />

<input

type="text"

value={name}

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

style={{ width: '300px', padding: '8px' }}

/>

</div>

<div style={{ marginBottom: '10px' }}>

<label>Complaint: </label><br />

<textarea

value={complaint}

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

rows="4"

style={{ width: '300px', padding: '8px' }}

></textarea>

</div>

<button type="submit" style={{ padding: '8px 16px' }}>Submit</button>

</form>

</div>

);

}

export default ComplaintRegister;

*App.js*

import React from 'react';

import ComplaintRegister from './ComplaintRegister';

function App() {

return (

<div className="App">

<h1>Ticket Raising App</h1>

<ComplaintRegister />

</div>

);

}

export default App;

### **Output:**