**DN 4.0 Dotnet FSE**

**Name: Sparsh Guha**

**Superset ID:6361106**

**Week 7**

**QUESTION-1 (REACT-9)**

**Create a React Application named “cricketapp”**

**INPUT AND CODE:**

1. Create the React App  
      
   npx create-react-app cricketapp  
   cd cricketapp  
     
   2. Create the ListofPlayers Component  
   **File: src/ListofPlayers.js**  
     
   import React from 'react';  
     
   const players = [  
   { name: "Rohit Sharma", score: 85 },  
   { name: "Shubman Gill", score: 45 },  
   { name: "Virat Kohli", score: 90 },  
   { name: "Shreyas Iyer", score: 60 },  
   { name: "KL Rahul", score: 75 },  
   { name: "Hardik Pandya", score: 55 },  
   { name: "Ravindra Jadeja", score: 80 },  
   { name: "Kuldeep Yadav", score: 35 },  
   { name: "Mohammed Shami", score: 25 },  
   { name: "Jasprit Bumrah", score: 70 },  
   { name: "Mohammed Siraj", score: 65 }  
   ];  
     
   const ListofPlayers = () => {  
   const allPlayers = players.map((p, idx) => (  
   <li key={idx}>{p.name} - {p.score}</li>  
   ));  
     
     
   const below70 = players.filter(p => p.score < 70);  
     
   return (  
   <div>  
   <h2>All Players</h2>  
   <ul>{allPlayers}</ul>  
   <h2>Players with Score Below 70</h2>  
   <ul>  
   {below70.map((p, idx) => (  
   <li key={idx}>{p.name} - {p.score}</li>  
   ))}  
   </ul>  
   </div>  
   );  
   };  
     
   export default ListofPlayers;  
     
   3. Create the IndianPlayers Component  
   **File: src/IndianPlayers.js**  
     
      
   import React from 'react';  
     
   const T20players = ["Rohit Sharma", "Virat Kohli", "Suryakumar Yadav", "Rishabh Pant"];  
   const RanjiTrophyPlayers = ["Cheteshwar Pujara", "Ajinkya Rahane", "Prithvi Shaw", "Hanuma Vihari"];  
     
     
   const mergedPlayers = [...T20players, ...RanjiTrophyPlayers];  
     
   const IndianPlayers = () => {  
     
   const oddTeam = mergedPlayers.filter((\_, idx) => idx % 2 === 0);  
   const evenTeam = mergedPlayers.filter((\_, idx) => idx % 2 !== 0);  
     
   return (  
   <div>  
   <h2>Merged Players</h2>  
   <ul>  
   {mergedPlayers.map((name, idx) => <li key={idx}>{name}</li>)}  
   </ul>  
   <h3>Odd Team Players (Destructured)</h3>  
   <ul>  
   {oddTeam.map((name, idx) => <li key={idx}>{name}</li>)}  
   </ul>  
   <h3>Even Team Players (Destructured)</h3>  
   <ul>  
   {evenTeam.map((name, idx) => <li key={idx}>{name}</li>)}  
   </ul>  
   </div>  
   );  
   };  
     
   export default IndianPlayers;  
     
   4. Edit App.js to Toggle Components with a Flag  
   **File: src/App.js**  
     
      
   import React, { useState } from 'react';  
   import ListofPlayers from './ListofPlayers';  
   import IndianPlayers from './IndianPlayers';  
     
   function App() {  
   const [flag, setFlag] = useState(true);  
     
   return (  
   <div>  
   <button onClick={() => setFlag(f => !f)}>  
   Toggle Component (Flag is {flag ? "true" : "false"})  
   </button>  
   {flag ? <ListofPlayers /> : <IndianPlayers />}  
   </div>  
   );  
   }  
     
   export default App;  
     
   5. Run the App  
      
   npm start  
     
   Open <http://localhost:3000>

**OUTPUT:**

****

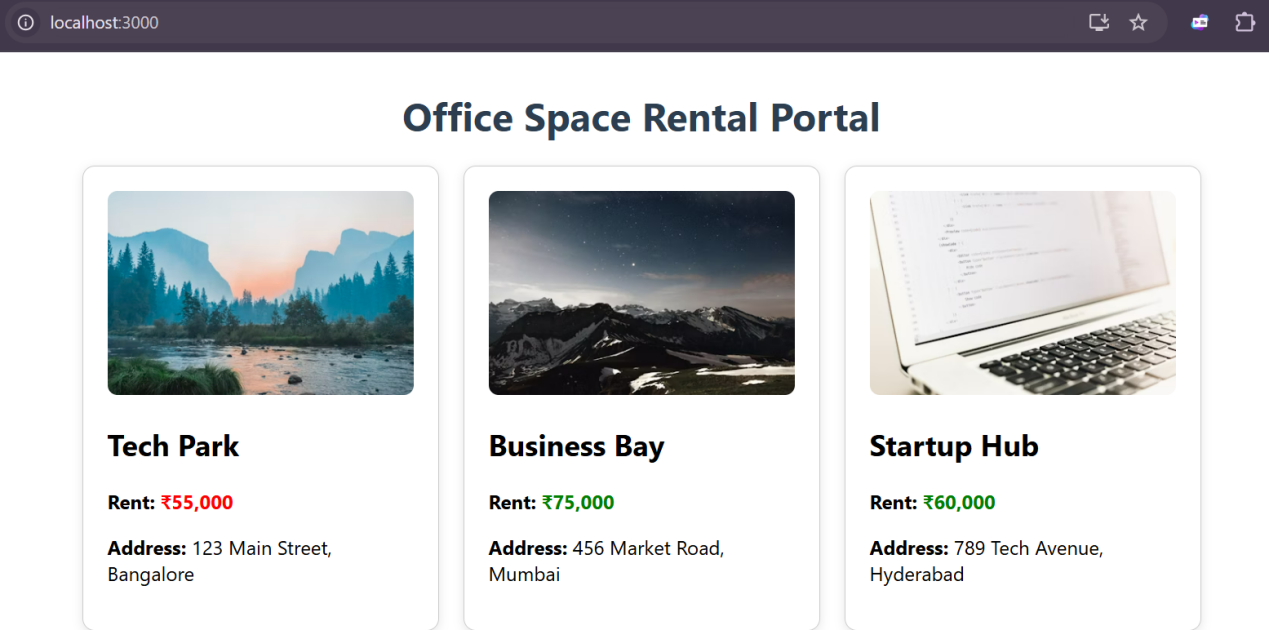
**QUESTION-2 (REACT-10)**

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

**INPUT AND CODE:**

1. Create a New React App  
      
   npx create-react-app officespacerentalapp  
   cd officespacerentalapp  
     
   **2. Edit src/App.js**  
   import React from 'react';  
     
     
   const offices = [  
   {  
   id: 1,  
   name: "Tech Park",  
   rent: 55000,  
   address: "123 Main Street, Bangalore",  
   image: "https://images.unsplash.com/photo-1506744038136-46273834b3fb?auto=format&fit=crop&w=400&q=80"  
   },  
   {  
   id: 2,  
   name: "Business Bay",  
   rent: 75000,  
   address: "456 Market Road, Mumbai",  
   image: "https://images.unsplash.com/photo-1464983953574-0892a716854b?auto=format&fit=crop&w=400&q=80"  
   },  
   {  
   id: 3,  
   name: "Startup Hub",  
   rent: 60000,  
   address: "789 Tech Avenue, Hyderabad",  
   image: "https://images.unsplash.com/photo-1482062364825-616fd23b8fc1?auto=format&fit=crop&w=400&q=80"  
   }  
   ];  
     
     
   const headingStyle = {  
   color: "#2c3e50",  
   textAlign: "center",  
   margin: "30px 0 20px 0"  
   };  
     
   function App() {  
   return (  
   <div>  
   {}  
   <h1 style={headingStyle}>Office Space Rental Portal</h1>  
     
   {}  
   <div style={{ display: "flex", justifyContent: "center", gap: "20px" }}>  
   {offices.map((office) => (  
   <div  
   key={office.id}  
   style={{  
   border: "1px solid #ccc",  
   borderRadius: "10px",  
   padding: "20px",  
   width: "250px",  
   boxShadow: "0 2px 8px rgba(0,0,0,0.1)"  
   }}  
   >  
   {}  
   <img  
   src={office.image}  
   alt={office.name}  
   style={{ width: "100%", borderRadius: "8px" }}  
   />  
     
   {}  
   <h2>{office.name}</h2>  
   <p>  
   <strong>Rent:</strong>{" "}  
   <span  
   style={{  
   color: office.rent < 60000 ? "red" : "green",  
   fontWeight: "bold"  
   }}  
   >  
   ₹{office.rent.toLocaleString()}  
   </span>  
   </p>  
   <p>  
   <strong>Address:</strong> {office.address}  
   </p>  
   </div>  
   ))}  
   </div>  
   </div>  
   );  
   }  
     
   export default App;  
     
   3. Run the App  
      
   npm start  
   Open <http://localhost:3000>

**OUTPUT:**

****

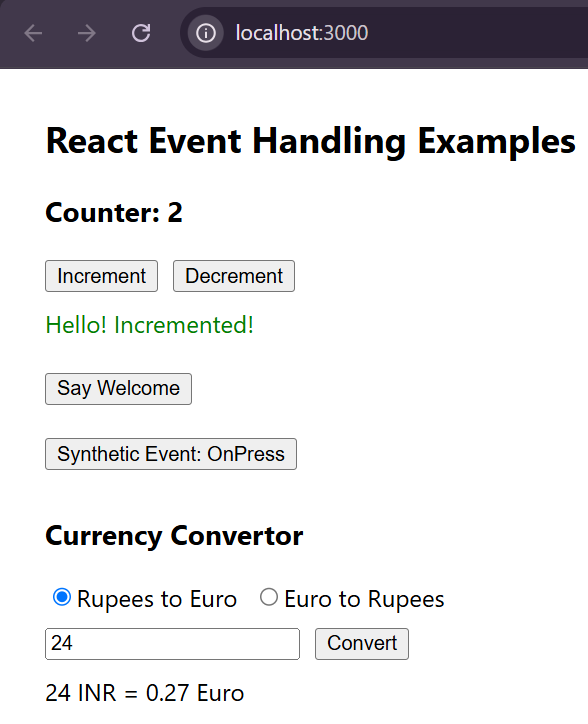
**QUESTION-3 (REACT-11)**

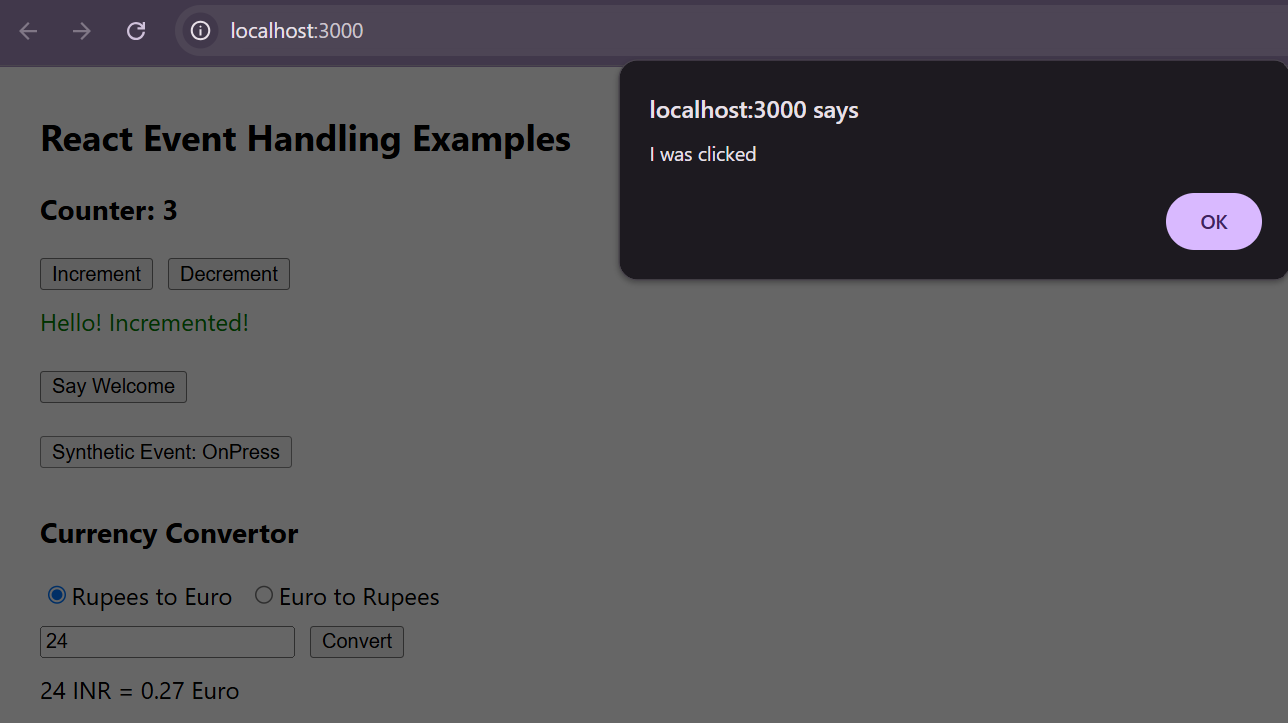
**Create a React Application “eventexamplesapp” to handle various events of the form elements in HTML**.

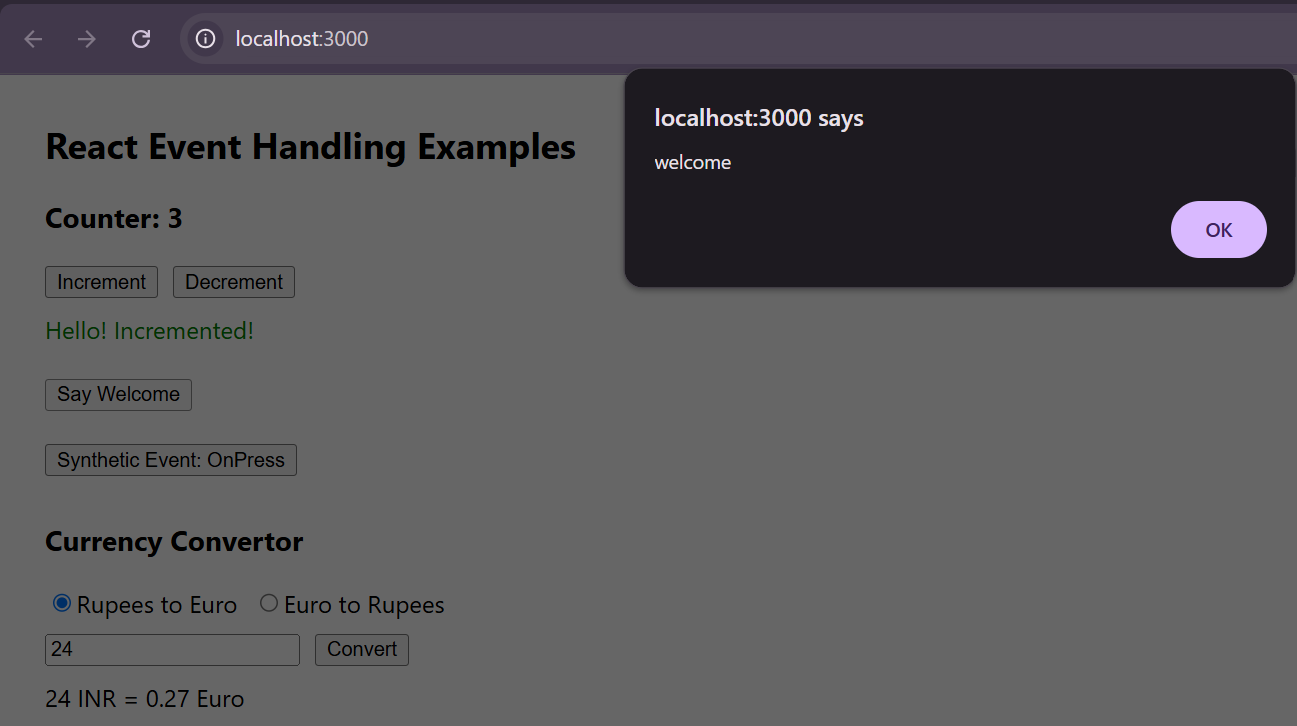
**INPUT AND CODE:**

1. Create a New React App  
      
   npx create-react-app eventexamplesapp  
   cd eventexamplesapp  
     
   **2. App.js — Integrated Code**  
     
   import React, { Component } from 'react';  
   import CurrencyConvertor from './CurrencyConvertor';  
     
   class App extends Component {  
   constructor(props) {  
   super(props);  
   this.state = {  
   counter: 0,  
   message: ''  
   };  
     
   this.handleWelcome = this.handleWelcome.bind(this);  
   }  
     
   increment = () => {  
   this.setState(prevState => ({ counter: prevState.counter + 1 }));  
   this.sayHello();  
   this.sayMessage("Incremented!");  
   };  
     
   decrement = () => {  
   this.setState(prevState => ({ counter: prevState.counter - 1 }));  
   };  
     
   sayHello = () => {  
   this.setState({ message: "Hello! " });  
   };  
     
   sayMessage = (msg) => {  
   this.setState(prev => ({ message: (prev.message || "") + msg }));  
   };  
     
   handleWelcome(msg) {  
   alert(msg);  
   }  
     
   handleSyntheticEvent = (e) => {  
     
   alert("I was clicked");  
   };  
     
   render() {  
   return (  
   <div style={{ margin: 30 }}>  
   <h2>React Event Handling Examples</h2>  
   <div style={{ marginBottom: 20 }}>  
   <h3>Counter: {this.state.counter}</h3>  
   <button onClick={this.increment}>Increment</button>  
   <button onClick={this.decrement} style={{ marginLeft: 10 }}>Decrement</button>  
   <div style={{ marginTop: 10, color: 'green' }}>{this.state.message}</div>  
   </div>  
     
   <div style={{ marginBottom: 20 }}>  
   <button onClick={() => this.handleWelcome("welcome")}>Say Welcome</button>  
   </div>  
     
   <div style={{ marginBottom: 20 }}>  
   <button onClick={this.handleSyntheticEvent}>Synthetic Event: OnPress</button>  
   </div>  
     
   <CurrencyConvertor />  
   </div>  
   );  
   }  
   }  
     
   export default App;  
     
   3. CurrencyConvertor.js  
   **Create a new file src/CurrencyConvertor.js:**  
     
   import React, { Component } from 'react';  
     
   class CurrencyConvertor extends Component {  
   constructor(props) {  
   super(props);  
   this.state = {  
   rupees: '',  
   euro: '',  
   direction: 'INRtoEURO'  
   };  
   }  
     
   handleInputChange = (e) => {  
   this.setState({ [e.target.name]: e.target.value });  
   };  
     
   handleDirectionChange = (e) => {  
   this.setState({ direction: e.target.value, rupees: '', euro: '' });  
   };  
     
   handleSubmit = (e) => {  
   e.preventDefault(); // Synthetic event  
   const { rupees, euro, direction } = this.state;  
   if (direction === 'INRtoEURO' && rupees) {  
   // Example: 1 Euro = 90 INR  
   this.setState({ euro: (parseFloat(rupees) / 90).toFixed(2) });  
   } else if (direction === 'EUROtoINR' && euro) {  
   this.setState({ rupees: (parseFloat(euro) \* 90).toFixed(2) });  
   }  
   };  
     
   render() {  
   const { rupees, euro, direction } = this.state;  
   return (  
   <div style={{ marginTop: 30 }}>  
   <h3>Currency Convertor</h3>  
   <form onSubmit={this.handleSubmit}>  
   <div>  
   <label>  
   <input  
   type="radio"  
   value="INRtoEURO"  
   checked={direction === "INRtoEURO"}  
   onChange={this.handleDirectionChange}  
   />  
   Rupees to Euro  
   </label>  
   <label style={{ marginLeft: 10 }}>  
   <input  
   type="radio"  
   value="EUROtoINR"  
   checked={direction === "EUROtoINR"}  
   onChange={this.handleDirectionChange}  
   />  
   Euro to Rupees  
   </label>  
   </div>  
   {direction === "INRtoEURO" ? (  
   <input  
   type="number"  
   name="rupees"  
   placeholder="Enter Rupees"  
   value={rupees}  
   onChange={this.handleInputChange}  
   required  
   style={{ marginTop: 10 }}  
   />  
   ) : (  
   <input  
   type="number"  
   name="euro"  
   placeholder="Enter Euro"  
   value={euro}  
   onChange={this.handleInputChange}  
   required  
   style={{ marginTop: 10 }}  
   />  
   )}  
   <button type="submit" style={{ marginLeft: 10 }}>Convert</button>  
   </form>  
   <div style={{ marginTop: 10 }}>  
   {direction === "INRtoEURO" && euro && (  
   <span>{rupees} INR = {euro} Euro</span>  
   )}  
   {direction === "EUROtoINR" && rupees && (  
   <span>{euro} Euro = {rupees} INR</span>  
   )}  
   </div>  
   </div>  
   );  
   }  
   }  
   export default CurrencyConvertor;

**OUTPUT:**







**QUESTION - 4 (REACT 12)**

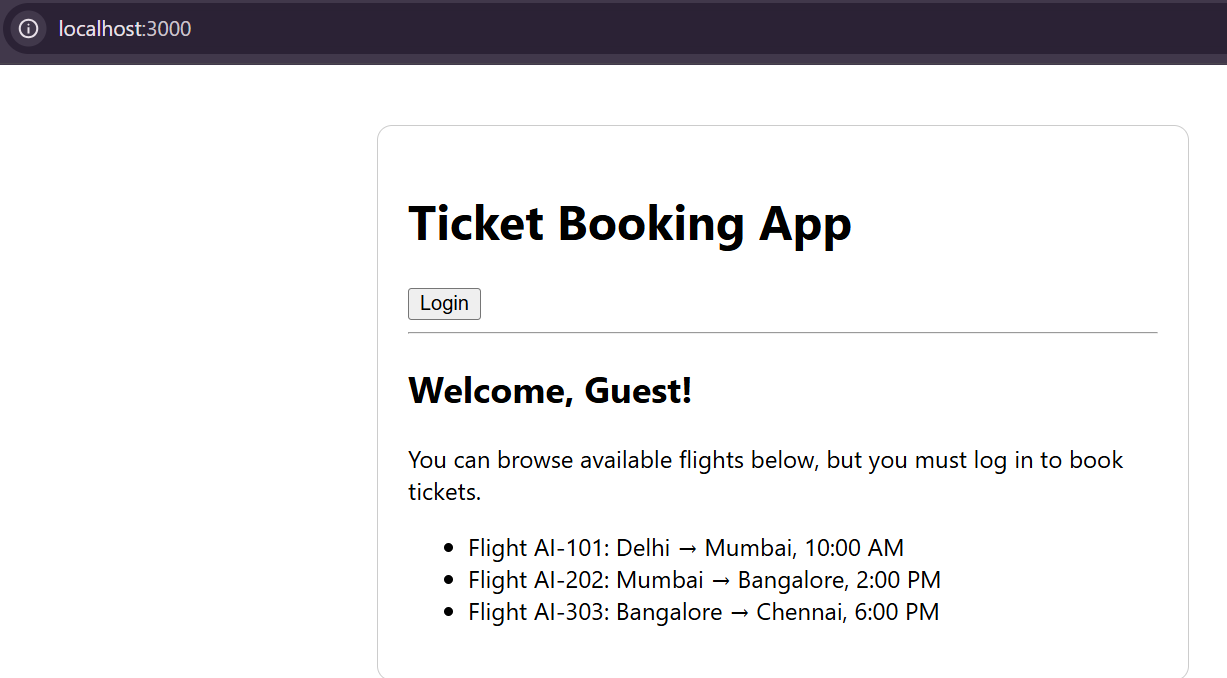
**Create a React Application named “ticketbookingapp” where the guest user can browse the page where the flight details are displayed whereas the logged in user only can book tickets.**

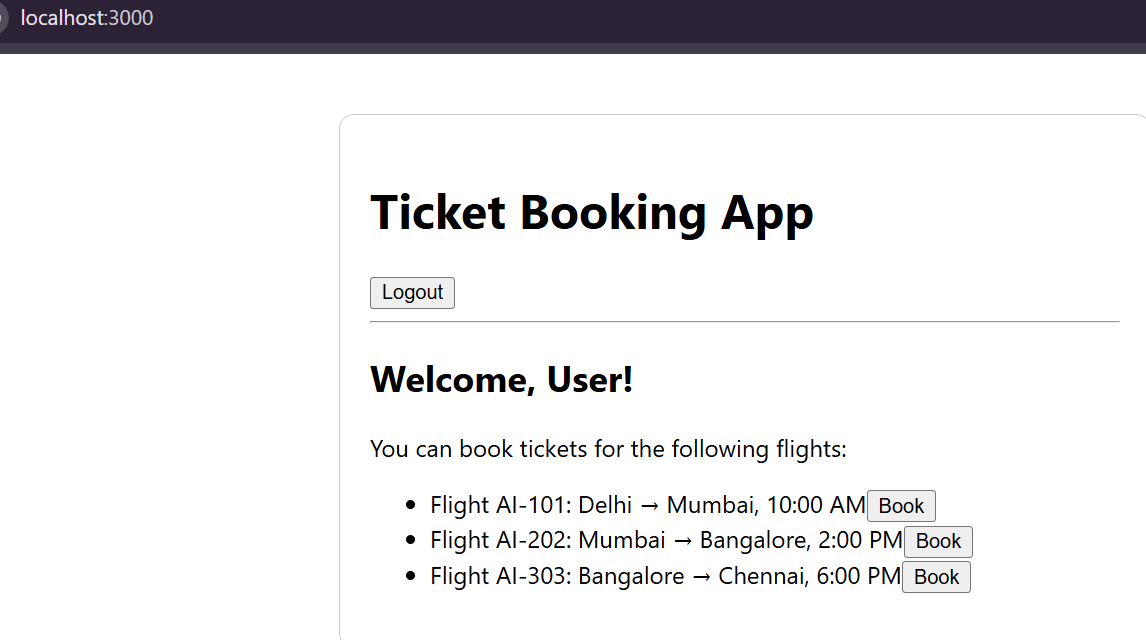
**The Login and Logout buttons should accordingly display different pages. Once the user is logged in the User page should be displayed. When the user clicks on Logout, the Guest page should be displayed.**

**INPUT AND CODE:**

Step 1: Create the React App  
   
npx create-react-app ticketbookingapp  
cd ticketbookingapp  
  
Step 2: Create Component Files  
**A. GuestPage.js**  
  
function GuestPage() {  
return (  
<div>  
<h2>Welcome, Guest!</h2>  
<p>You can browse available flights below, but you must log in to book tickets.</p>  
<ul>  
<li>Flight AI-101: Delhi → Mumbai, 10:00 AM</li>  
<li>Flight AI-202: Mumbai → Bangalore, 2:00 PM</li>  
<li>Flight AI-303: Bangalore → Chennai, 6:00 PM</li>  
</ul>  
</div>  
);  
}  
  
export default GuestPage;  
  
**B. UserPage.js**  
   
function UserPage({ onBook }) {  
return (  
<div>  
<h2>Welcome, User!</h2>  
<p>You can book tickets for the following flights:</p>  
<ul>  
<li>  
Flight AI-101: Delhi → Mumbai, 10:00 AM  
<button onClick={() => onBook("AI-101")}>Book</button>  
</li>  
<li>  
Flight AI-202: Mumbai → Bangalore, 2:00 PM  
<button onClick={() => onBook("AI-202")}>Book</button>  
</li>  
<li>  
Flight AI-303: Bangalore → Chennai, 6:00 PM  
<button onClick={() => onBook("AI-303")}>Book</button>  
</li>  
</ul>  
</div>  
);  
}  
  
export default UserPage;  
  
Step 3: Edit App.js for Conditional Rendering  
**File: src/App.js**  
  
   
import React, { useState } from 'react';  
import GuestPage from './GuestPage';  
import UserPage from './UserPage';  
  
function App() {  
const [isLoggedIn, setIsLoggedIn] = useState(false);  
const [bookingMsg, setBookingMsg] = useState('');  
  
  
let AuthButton;  
if (isLoggedIn) {  
AuthButton = <button onClick={() => setIsLoggedIn(false)}>Logout</button>;  
} else {  
AuthButton = <button onClick={() => setIsLoggedIn(true)}>Login</button>;  
}  
  
  
const handleBook = (flight) => {  
setBookingMsg(`Ticket booked for flight ${flight}!`);  
setTimeout(() => setBookingMsg(''), 2000);  
};  
  
return (  
<div style={{ maxWidth: 500, margin: "40px auto", padding: 20, border: "1px solid #ccc", borderRadius: 10 }}>  
<h1>Ticket Booking App</h1>  
{AuthButton}  
<hr />  
{isLoggedIn ? <UserPage onBook={handleBook} /> : <GuestPage />}  
{bookingMsg && <div style={{ color: "green", marginTop: 20 }}>{bookingMsg}</div>}  
</div>  
);  
}  
  
export default App;  
  
Step 4: Run the App  
   
npm start  
  
Open <http://localhost:3000>

**OUTPUT:**

****

****

**QUESTION-5 (REACT-13)**

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

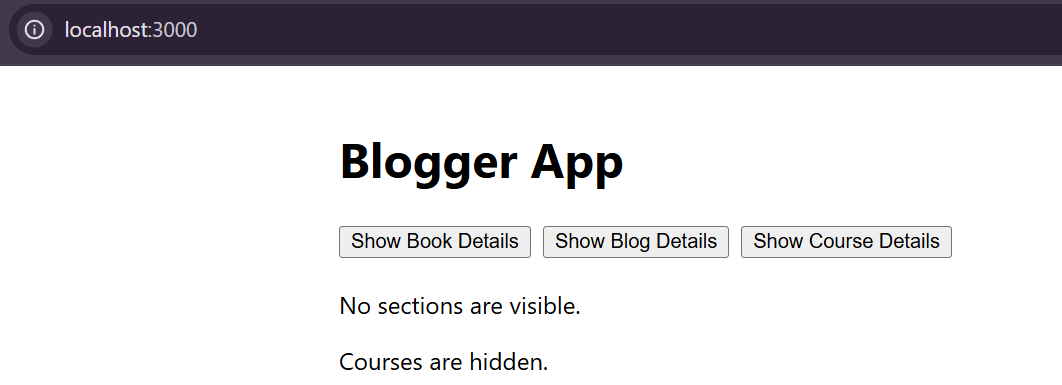
1. **Book Details**
2. **Blog Details**
3. **Course Details**

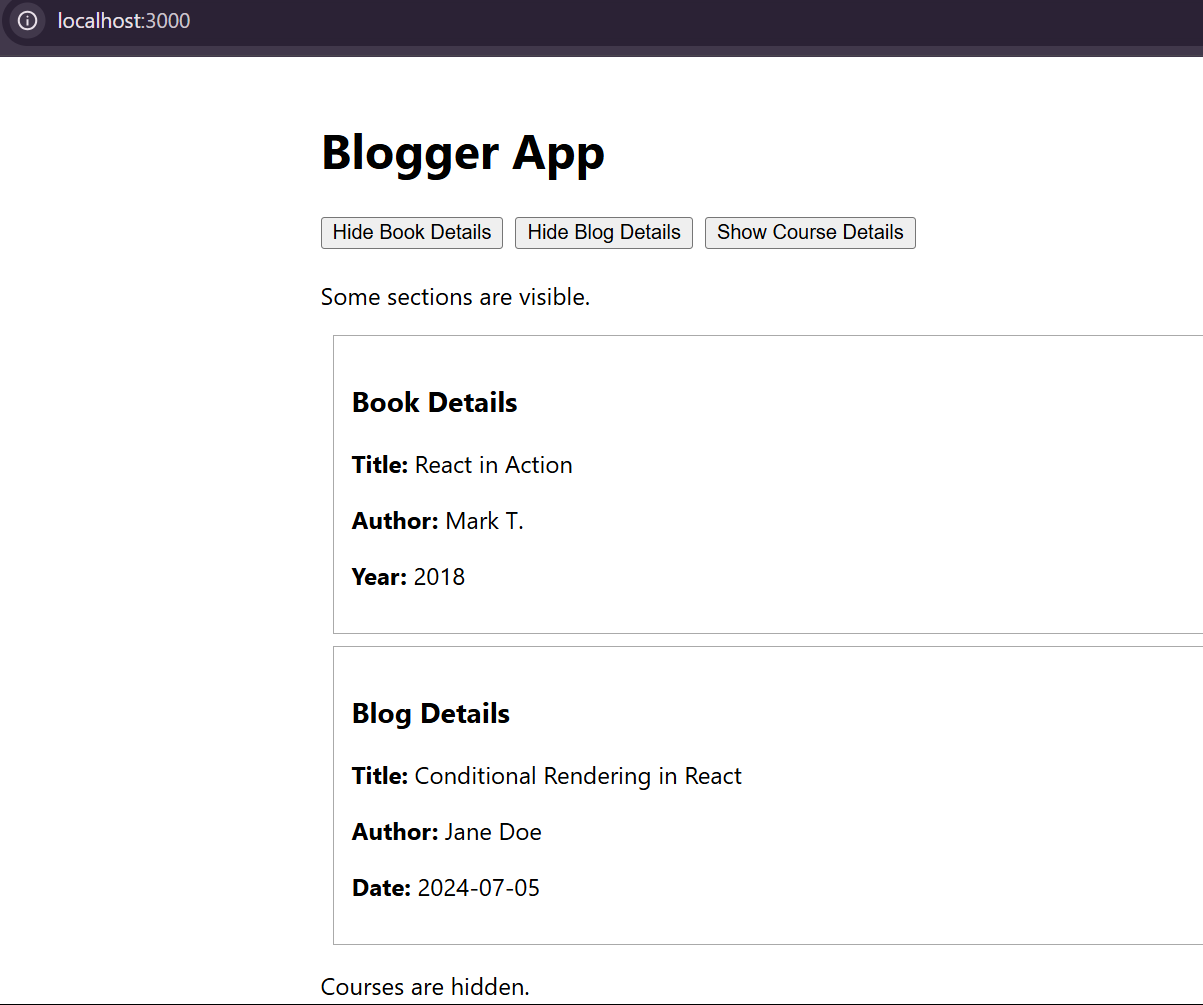
**Implement this with as many ways possible of Conditional Rendering**

**INPUT AND CODE:**

1. Create the React App  
      
   npx create-react-app bloggerapp  
   cd bloggerapp  
     
   2. Create the Components  
   A. **BookDetails.js**  
      
   function BookDetails({ book }) {  
   if (!book) return null; // Prevent rendering if no book  
   return (  
   <div style={{ border: "1px solid #aaa", padding: 12, margin: 8 }}>  
   <h3>Book Details</h3>  
   <p><strong>Title:</strong> {book.title}</p>  
   <p><strong>Author:</strong> {book.author}</p>  
   <p><strong>Year:</strong> {book.year}</p>  
   </div>  
   );  
   }  
     
   export default BookDetails;  
     
   B. **BlogDetails.js**  
      
   function BlogDetails({ blog }) {  
   return blog ? (  
   <div style={{ border: "1px solid #aaa", padding: 12, margin: 8 }}>  
   <h3>Blog Details</h3>  
   <p><strong>Title:</strong> {blog.title}</p>  
   <p><strong>Author:</strong> {blog.author}</p>  
   <p><strong>Date:</strong> {blog.date}</p>  
   </div>  
   ) : null;  
   }  
     
   export default BlogDetails;  
     
   C. **CourseDetails.js**  
   function CourseDetails({ courses }) {  
   return (  
   <div style={{ border: "1px solid #aaa", padding: 12, margin: 8 }}>  
   <h3>Course Details</h3>  
   {courses && courses.length > 0 ? (  
   <ul>  
   {courses.map(course => (  
   <li key={course.id}>  
   {course.name} ({course.duration} weeks)  
   </li>  
   ))}  
   </ul>  
   ) : (  
   <p>No courses available.</p>  
   )}  
   </div>  
   );  
   }  
     
   export default CourseDetails;  
     
   3. **Edit App.js** to Demonstrate Conditional Rendering  
      
   import React, { useState } from 'react';  
   import BookDetails from './BookDetails';  
   import BlogDetails from './BlogDetails';  
   import CourseDetails from './CourseDetails';  
     
   function App() {  
   const [showBook, setShowBook] = useState(true);  
   const [showBlog, setShowBlog] = useState(true);  
   const [showCourses, setShowCourses] = useState(true);  
     
   const book = { title: "React in Action", author: "Mark T.", year: 2018 };  
   const blog = { title: "Conditional Rendering in React", author: "Jane Doe", date: "2024-07-05" };  
   const courses = [  
   { id: 1, name: "React Basics", duration: 4 },  
   { id: 2, name: "Advanced React", duration: 6 },  
   { id: 3, name: "React Testing", duration: 3 }  
   ];  
     
   function renderSection() {  
   if (showBook && showBlog && showCourses) {  
   return <p>All sections are visible.</p>;  
   } else if (!showBook && !showBlog && !showCourses) {  
   return <p>No sections are visible.</p>;  
   } else {  
   return <p>Some sections are visible.</p>;  
   }  
   }  
     
   return (  
   <div style={{ maxWidth: 600, margin: "40px auto" }}>  
   <h1>Blogger App</h1>  
   {/\* Toggle buttons \*/}  
   <div style={{ marginBottom: 20 }}>  
   <button onClick={() => setShowBook(b => !b)}>  
   {showBook ? "Hide" : "Show"} Book Details  
   </button>  
   <button onClick={() => setShowBlog(b => !b)} style={{ marginLeft: 8 }}>  
   {showBlog ? "Hide" : "Show"} Blog Details  
   </button>  
   <button onClick={() => setShowCourses(b => !b)} style={{ marginLeft: 8 }}>  
   {showCourses ? "Hide" : "Show"} Course Details  
   </button>  
   </div>  
   {}  
   {renderSection()}  
   {}  
   {showBook && <BookDetails book={book} />}  
   {showBlog ? <BlogDetails blog={blog} /> : null}  
   {}  
   {showCourses ? <CourseDetails courses={courses} /> : <p>Courses are hidden.</p>}  
   </div>  
   );  
   }  
     
   export default App;  
      
   4. Run the App  
      
   npm start  
   Open <http://localhost:3000>

**OUTPUT:**

****

****