**React.js (wk-07)**

**>>handsOn-9**

* **Use map() method of ES6**
* **Apply arrow functions of ES6**
* **Implement Destructuring features of ES6**

🡺solution:

ListOfPlayers.js

const ListofPlayers = () => {

  const players = [

    { name: 'Virat Kohli', score: 89 },

    { name: 'Rohit Sharma', score: 45 },

    { name: 'KL Rahul', score: 78 },

    { name: 'Rishabh Pant', score: 65 },

    { name: 'Hardik Pandya', score: 92 },

    { name: 'Ravindra Jadeja', score: 55 },

    { name: 'Jasprit Bumrah', score: 25 },

    { name: 'Mohammed Shami', score: 35 },

    { name: 'Yuzvendra Chahal', score: 18 },

    { name: 'Bhuvneshwar Kumar', score: 42 },

    { name: 'Shikhar Dhawan', score: 85 },

  ]

  const playersBelow70 = players.filter((player) => player.score < 70)

  return (

    <div>

      <h2>List of Players</h2>

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

          <li key={index}>

            {player.name} - Score: {player.score}

          </li>))}

      </ul>

      <h2>Players with Scores Below 70:</h2>

      <ul>{playersBelow70.map((player, index) => (

          <li key={index}>

            {player.name} - Score: {player.score}

          </li>))}

      </ul>

    </div>

  )

}

export default ListofPlayers;

IndianPlayers.js

const IndianPlayers = () => {

  const oddTeamPlayers = ["player1", "player3", "player5", "player7", "player9", "player11"]

  const evenTeamPlayers = ["player2", "player4", "player6", "player8", "player10"]

  const T20players = [

    'Virat Kohli',

    'Rohit Sharma',

    'KL Rahul',]

  const RanjiTrophyPlayers = [

    'Ajinkya Rahane',

    'Cheteshwar Pujara',

    'Wriddhiman Saha',]

  const mergedPlayers = [...T20players, ...RanjiTrophyPlayers]

  return (

    <div>

      <h3>Odd Team Players:</h3>

      <ul>

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

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

        ))}

      </ul>

      <h3>Even Team Players:</h3>

      <ul>{evenTeamPlayers.map((player, index) => (

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

      </ul>

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

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

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

))}

      </ul>

    </div>

  )}

export default IndianPlayers

App.js

function App() {

const flag = false;

  return (

    <div className="App">

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

        <h1>Cricket App</h1>

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

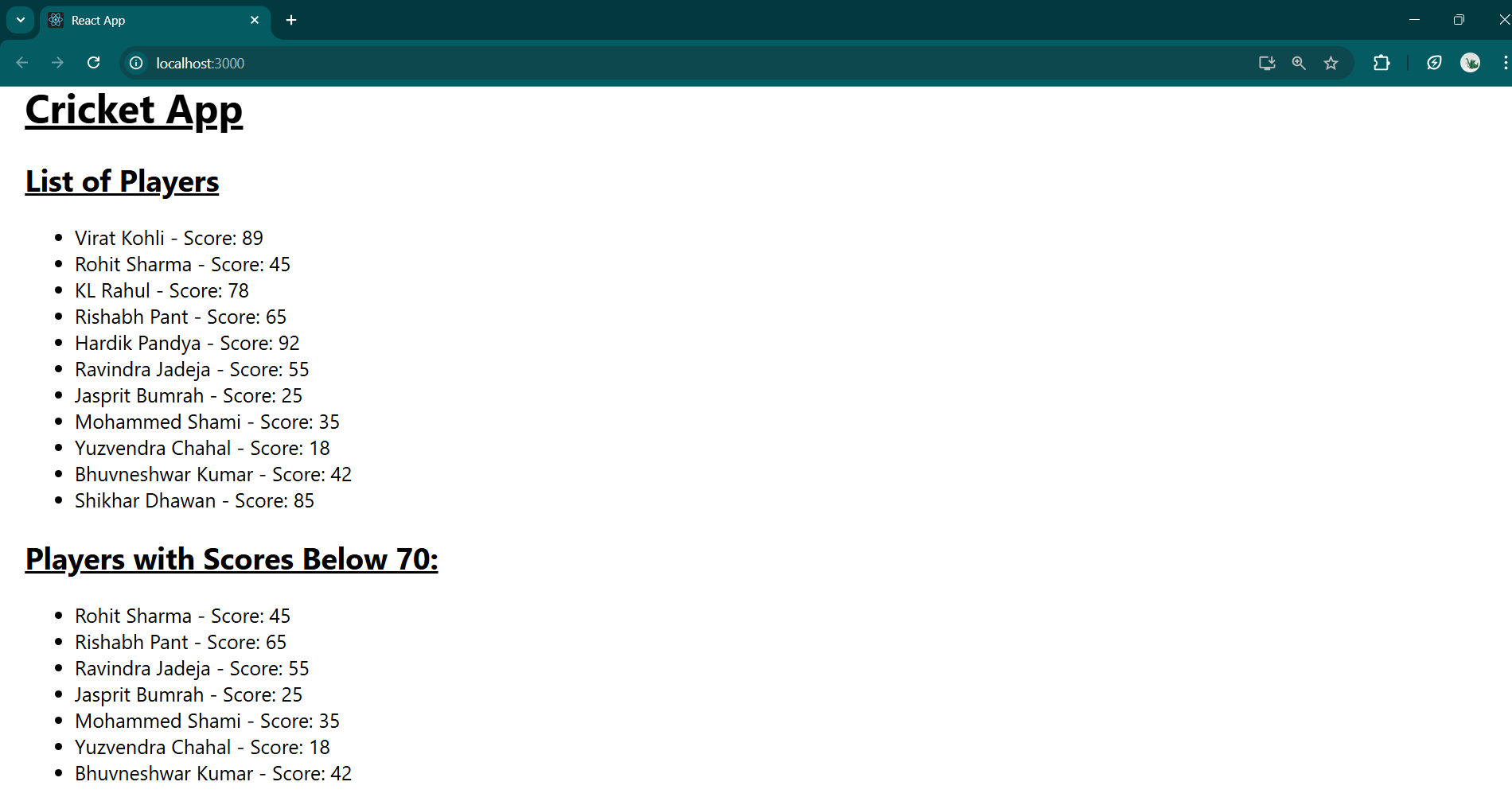
      </div>

    </div>

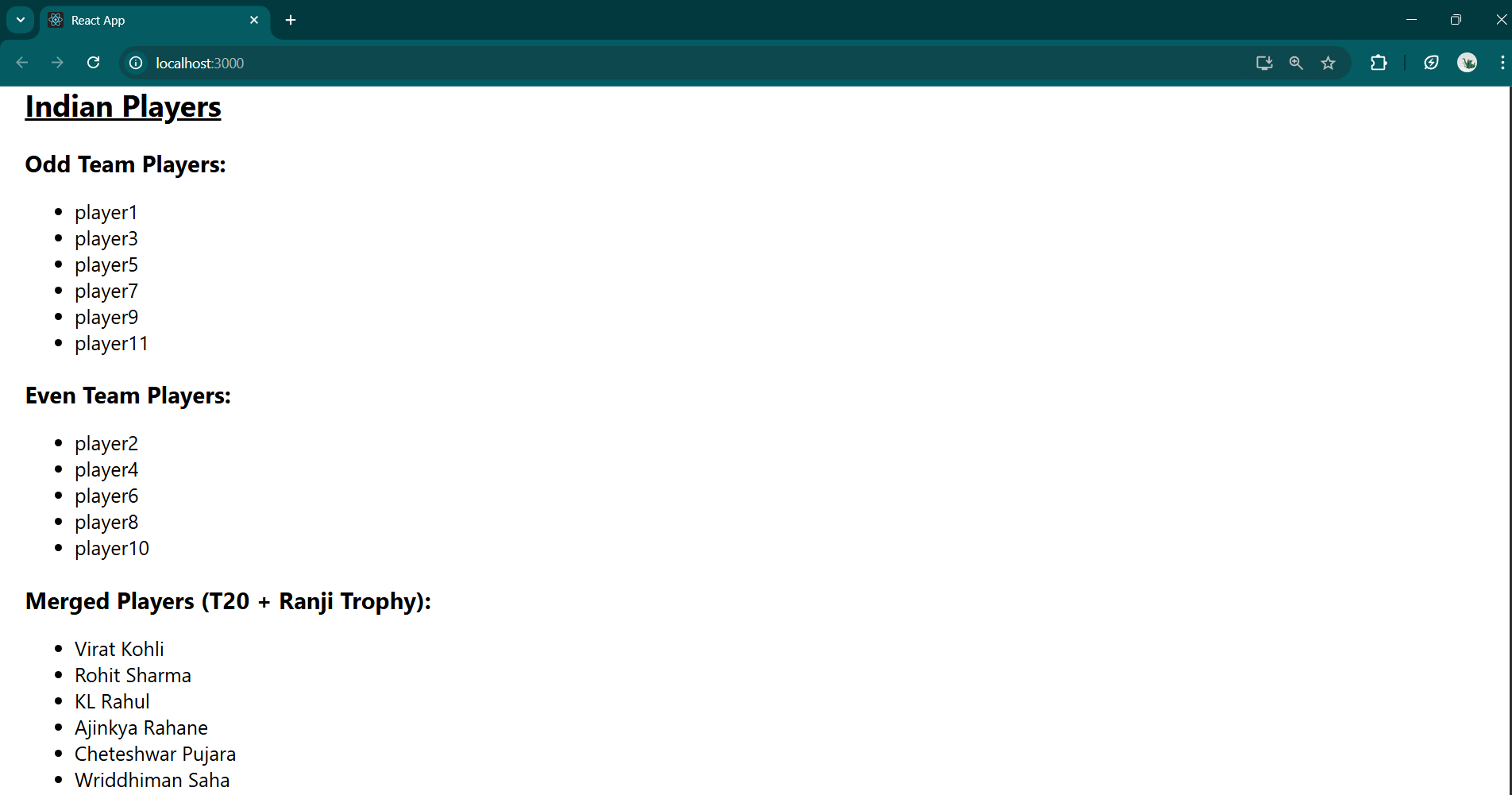
  )}

export default App;

when flag===true



when flag===false



**>>handsOn-10**

* **Use JSX syntax in React applications**
* **Use inline CSS in JSX**

🡺solution:

App.js

function App() {

  const office ={

      name: 'Shom Tech Hub',

      rent: 75000,

      address: 'Saltlake, Kolkata',

      image:

        'https://images.unsplash.com/photo-1497366811353-6870744d04b2?w=400&h=300&fit=crop',

    };

  const getRentColor = (rent) => {

    return rent < 60000 ? 'red' : 'green'

  }

  return (

    <div>

      <header>

        <h1>Office Space Rental App</h1>

        <p>Find the perfect office space for your business</p>

      </header>

      <main>

        <div>

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

            <div>

              <h3>{office.name}</h3>

              <p>{office.address}</p>

              <p>

                <span style={{color: getRentColor(office.rent)}}>

                  Rent: Rs. {office.rent}

                </span>

              </p>

            </div>

          </div>

      </main>

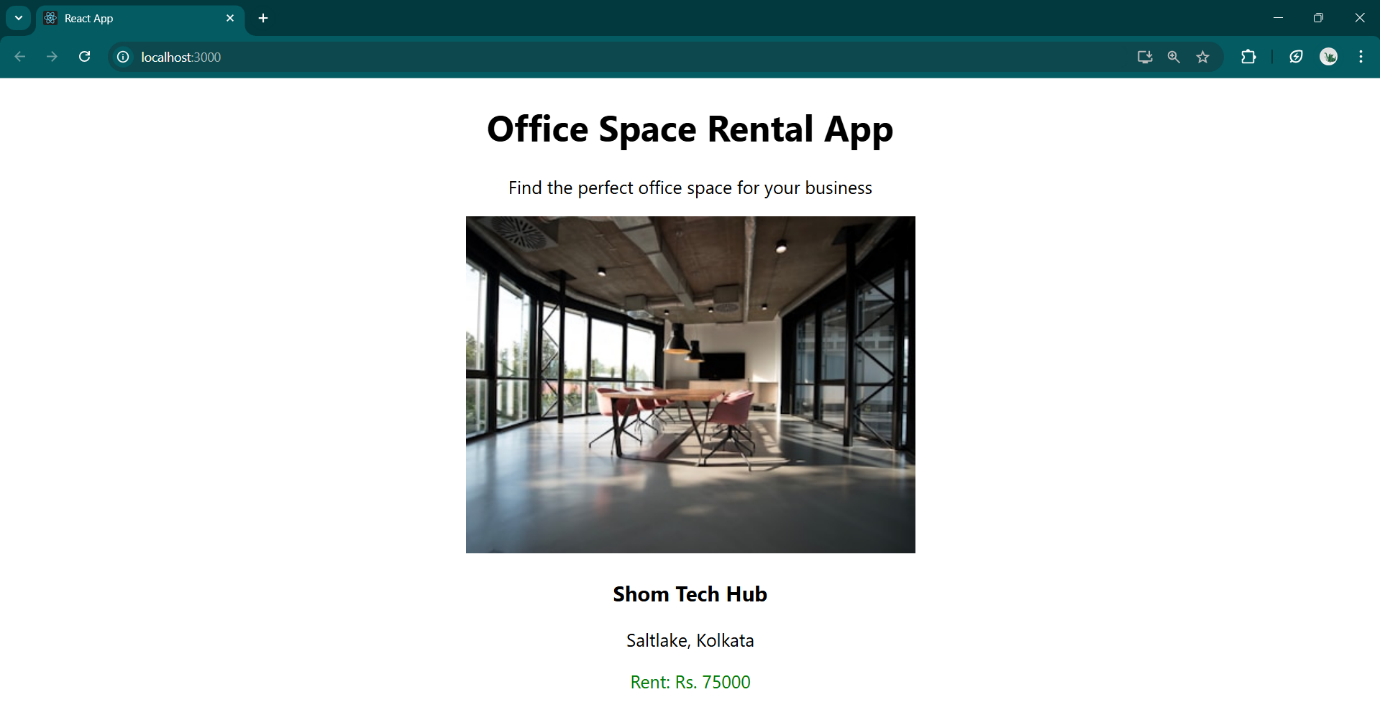
    </div>

  )

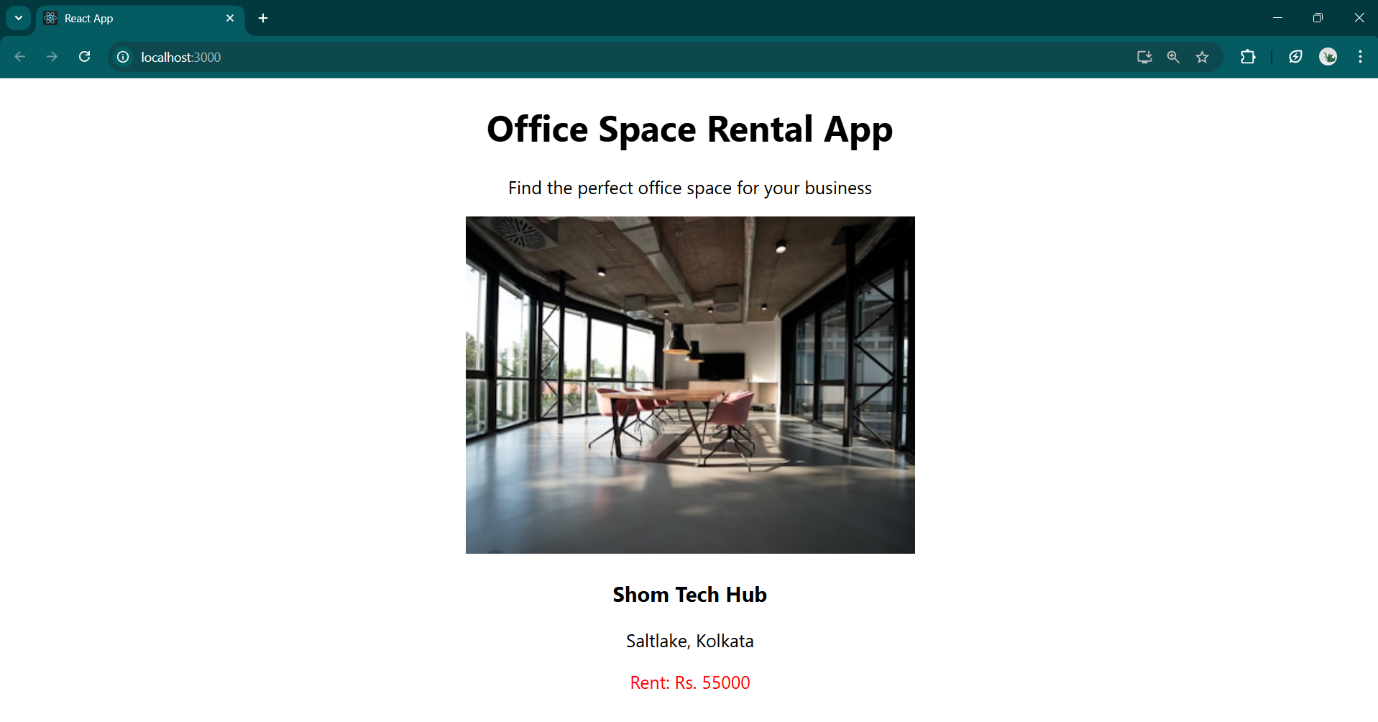
}

export default App;

when price>60,000:



when price<60,000:



**>>handsOn-11**

* **Implement Event handling concept in React applications**
* **Use this keyword**
* **Use synthetic event**

🡺solution

CurrencyConverter.js

const CurrencyConverter = () => {

  const [amount, setAmount] = useState('')

  const RUPEES\_TO\_EURO\_RATE = 0.011

  const handleAmountChange = (event) => {

    setAmount(event.target.value)

  }

  const handleSubmit = (event) => {

    event.preventDefault()

    if (amount && !isNaN(amount)) {

      const convertedEuros = (parseFloat(amount) \* RUPEES\_TO\_EURO\_RATE).toFixed(2)

      alert(`₹${amount} = €${convertedEuros}`)

    } else {

      alert('Please enter a valid amount in rupees')

    }

  }

  return (

    <div>

      <h2>Currency Converter</h2>

      <form onSubmit={handleSubmit}>

        <div>

          <label htmlFor="amount">Indian Rupees:</label>

          <input

            type="number"

            id="amount"

            value={amount}

            onChange={handleAmountChange}

            placeholder="Enter amount in rupees"

          />

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

        </div>

      </form>

    </div>

  )

}

export default CurrencyConverter;

App.js

function App() {

  const [counter, setCounter] = useState(0)

  const incrementValue = () => {

    setCounter(counter + 1)

  }

  const sayHello = () => {

    alert('Hello! Work, Eat, Sleep, Repeat...')

  }

  const handleIncrement = () => {

    incrementValue()

    sayHello()

  }

  const handleDecrement = () => {

    setCounter(counter - 1)

  }

  const sayWelcomeMessage = (text) => {

    alert(`${text}! Keep shining!`)

  }

  const handleWelcomeClick = () => {

    sayWelcomeMessage('Welcome')

  }

  const handleOnPress = (event) => {

    event.preventDefault()

    alert('I was clicked')

  }

  return (

    <div>

      <h1>Event Examples App</h1>

      <div>

        <h2>Counter: {counter}</h2>

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

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

      </div>

      <div>

        <button onClick={handleWelcomeClick}>Say Welcome</button>

      </div>

      <div>

        <button onClick={handleOnPress}>OnPress Event</button>

      </div>

      <div>

        <CurrencyConverter />

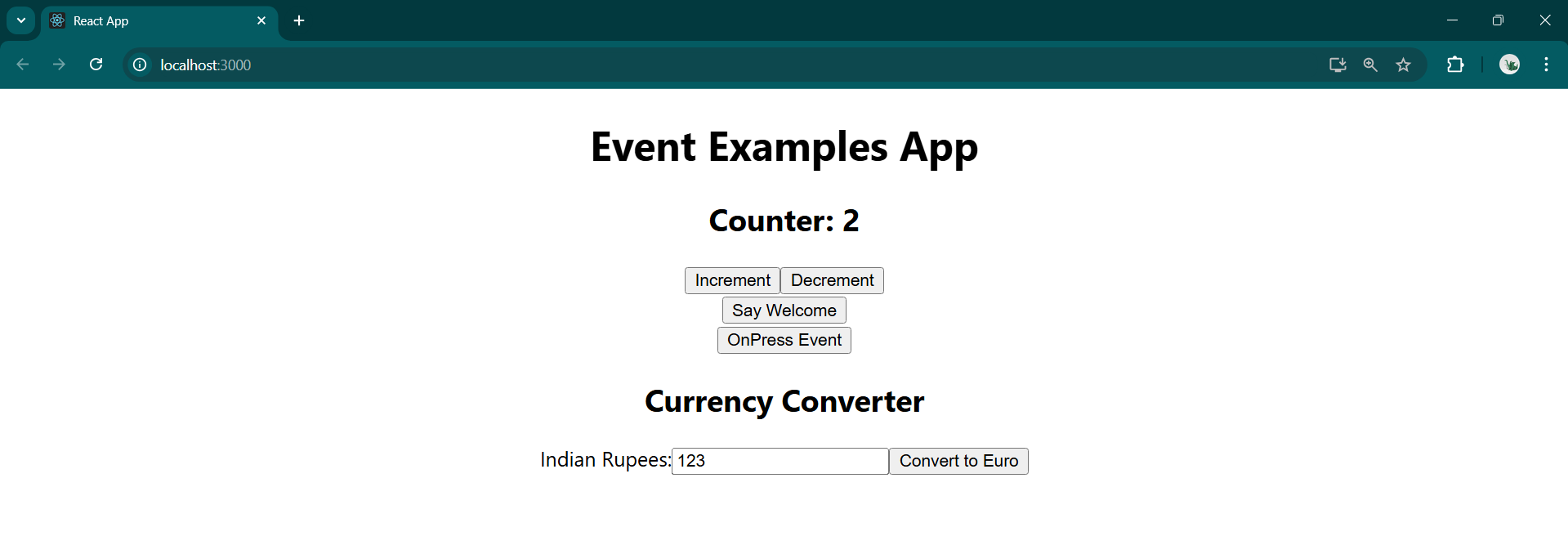
      </div>

    </div>

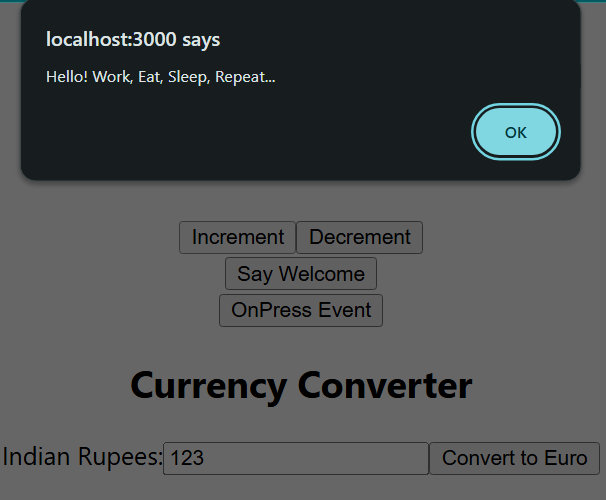
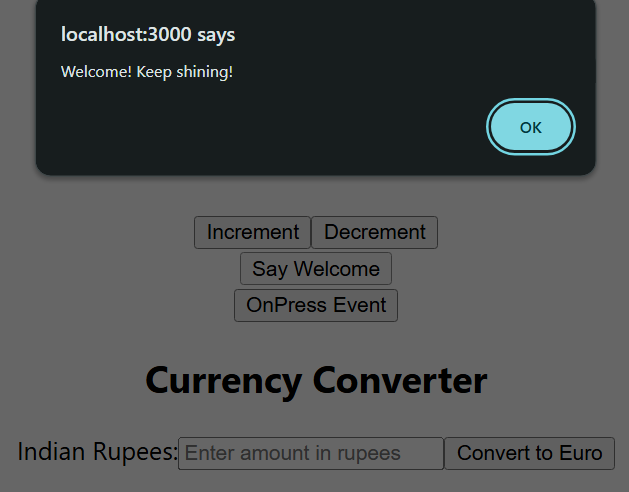
  )

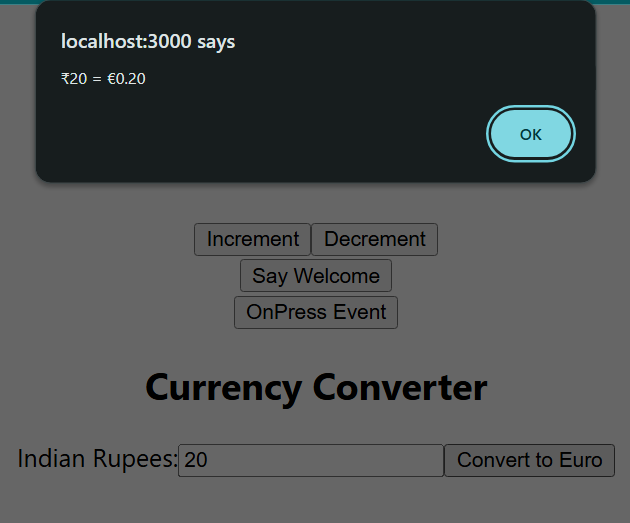
}

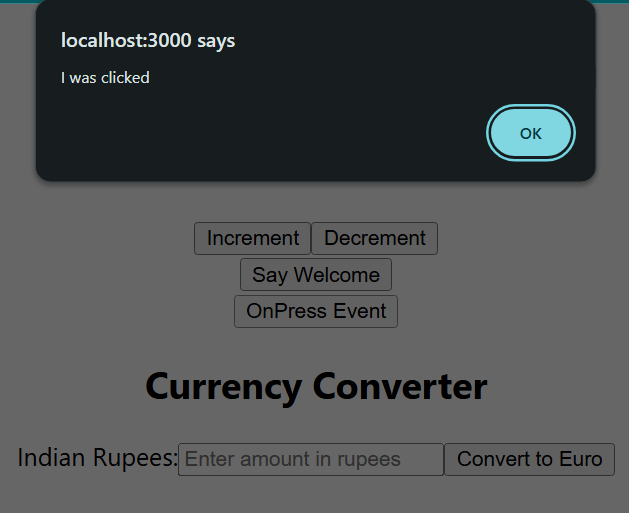
export default App;



Clicking on: “Increment” Clicking on: “Say Welcome”



Clicking on: “OnPress Event” Clicking on: “Convert to Euros”



**>>HandsOn-12**

* **Implement conditional rendering in React applications**

🡺 solution:

UserPage.js

const UserPage = ({ onLogout, user }) => {

  return (

    <div style={{textAlign:"center"}}>

      <header>

        <h1>Flight Booking System</h1>

        <div>

          <p>Welcome back, {user.name}!</p>

          <button className="logout-btn" onClick={onLogout}>

            Logout

          </button>

        </div>

      </header>

    </div>

  )

}

export default UserPage;

GuestPage.js

const GuestPage = ({ onLogin }) => {

  return (

    <div style={{textAlign:"center"}}>

      <header >

        <h1>Flight Booking System - Guest View</h1>

        <div>

          <p>Please Sign in to book tickets</p>

          <button className="login-btn" onClick={onLogin}>

            Login

          </button>

        </div>

      </header>

    </div>

  )

}

export default GuestPage

App.js

function App() {

  const [isLoggedIn, setIsLoggedIn] = useState(false)

  const [user, setUser] = useState(null)

  const handleLogin = () => {

    setUser({ name: 'Shom' })

    setIsLoggedIn(true)

  }

  const handleLogout = () => {

    setIsLoggedIn(false)

    setUser(null)

  }

  return (

    <div>

      {isLoggedIn ? (

        <UserPage onLogout={handleLogout} user={user} />

      ) : (

        <GuestPage onLogin={handleLogin} />

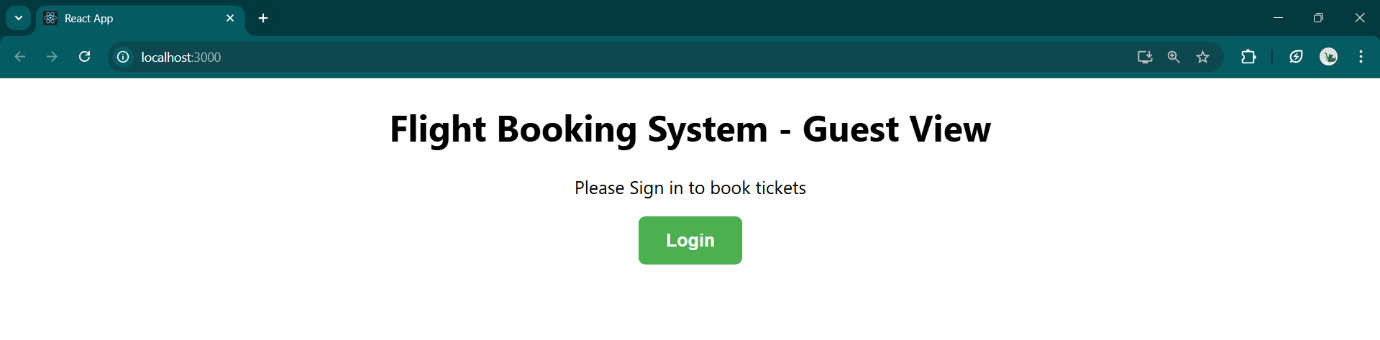
      )}

    </div>

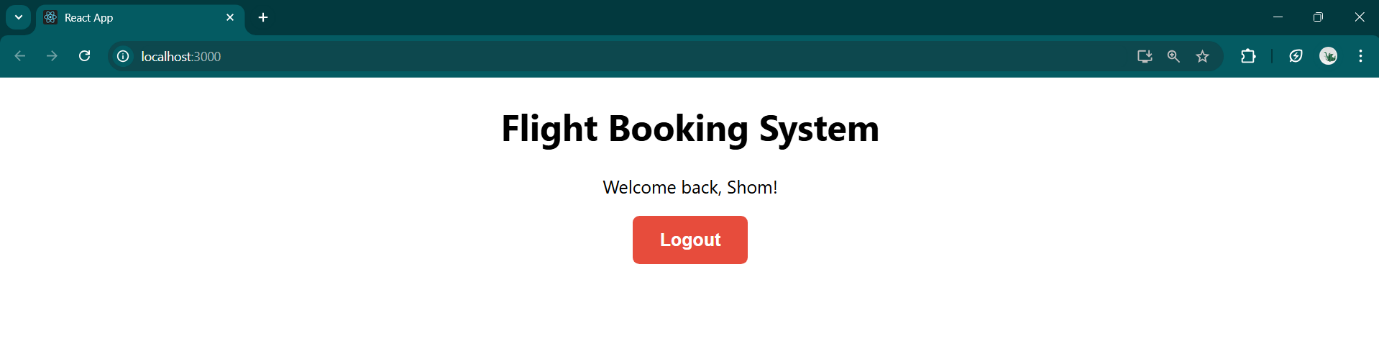
  )

}

export default App



while logged out:



while logged in:

**>>HandsOn-13:**

* Implement conditional rendering in React applications

🡺 solution:

BookDetails.js

const BookDetails = () => {

  return (

    <div>

      <h1 style={{color:"green"}}>Book Details Component</h1>

    </div>

  )

}

export default BookDetails

BlogDetails.js

const BlogDetails = () => {

  return (

    <div>

      <h1 style={{color:"green"}}>Blog Details Component</h1>

    </div>

  )

}

export default BlogDetails

CourseDetails.js

const CourseDetails = () => {

  return (

    <div>

      <h1 style={{color:"green"}}>Course Details Component</h1>

    </div>

  )

}

export default CourseDetails

App.js

function App() {

  const [activeComponent, setActiveComponent] = useState('books')

  const componentMap = {

    books: <BookDetails />,

    blogs: <BlogDetails />,

    courses: <CourseDetails />,

  }

  return (

    <div className="App">

      <div className="main-app">

        <header style={{color:"blue"}}>

          <h1>BloggerApp</h1>

        </header>

        <nav>

          <div style={{ textAlign: 'center' }}>

            <button

              onClick={() => setActiveComponent('books')}

            >

              Books

            </button>

            <button

              onClick={() => setActiveComponent('blogs')}

            >

              Blogs

            </button>

            <button

              onClick={() => setActiveComponent('courses')}

            >

              Courses

            </button>

          </div>

        </nav>

        <main>

          <div>

            {componentMap[activeComponent]}

          </div>

        </main>

        <footer>

          <p>

            Currently viewing: <strong>{activeComponent}</strong>

          </p>

          <p>Switch between components using the navigation above</p>

        </footer>

      </div>

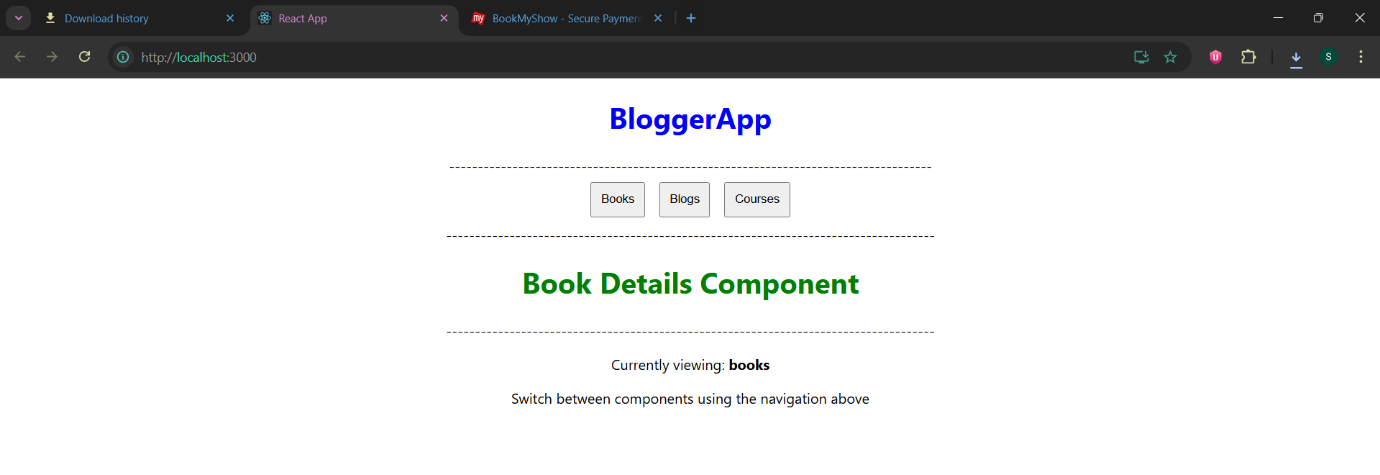
    </div>

  )

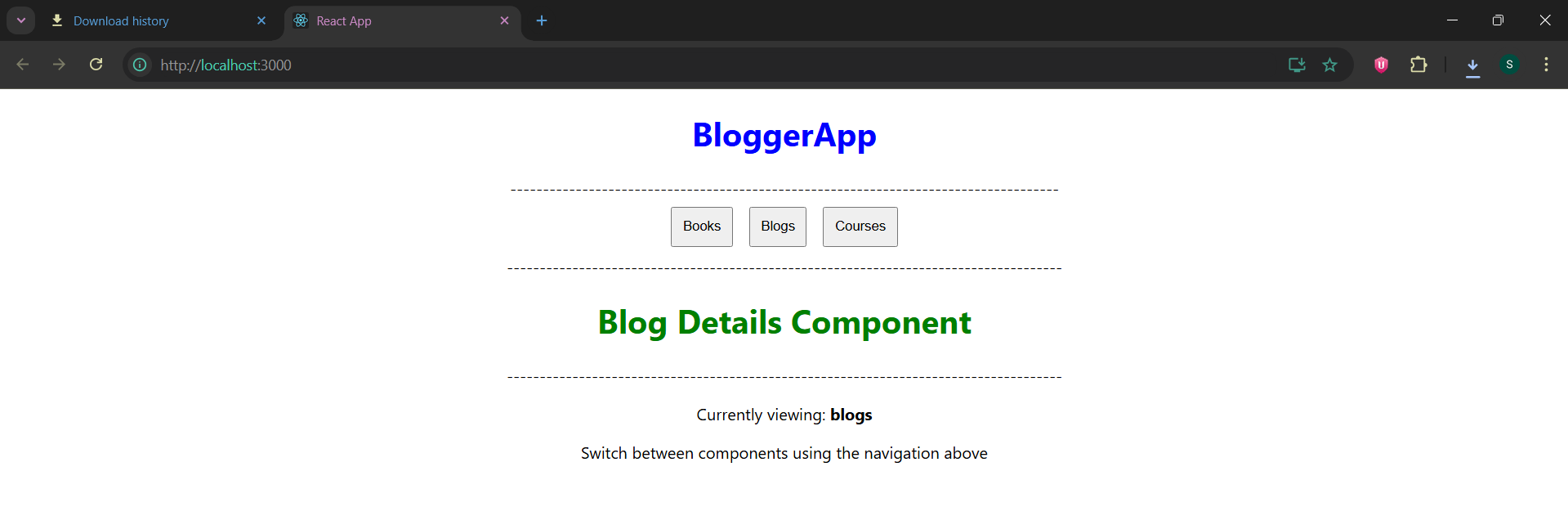
}

export default App

When clicked on: Books



When clicked on: Blogs



When clicked on: Courses

