WEEK-7 HANDS ON

Exercise 1:

Create a React Application named “cricketapp” with the following components:

App.js

```
import React from 'react';
import './App.css';
import ListofPlayers from './ListofPlayers';
import IndianPlayers from './IndianPlayers';

function App() {
  const flag = true; // Change to false to display IndianPlayers

  return (
    <div className="App">
      <h1>🏏 Cricket App</h1>
      {flag ? <ListofPlayers /> : <IndianPlayers />}
    </div>
  );
}

export default App;
```

ListOfPlayers.js

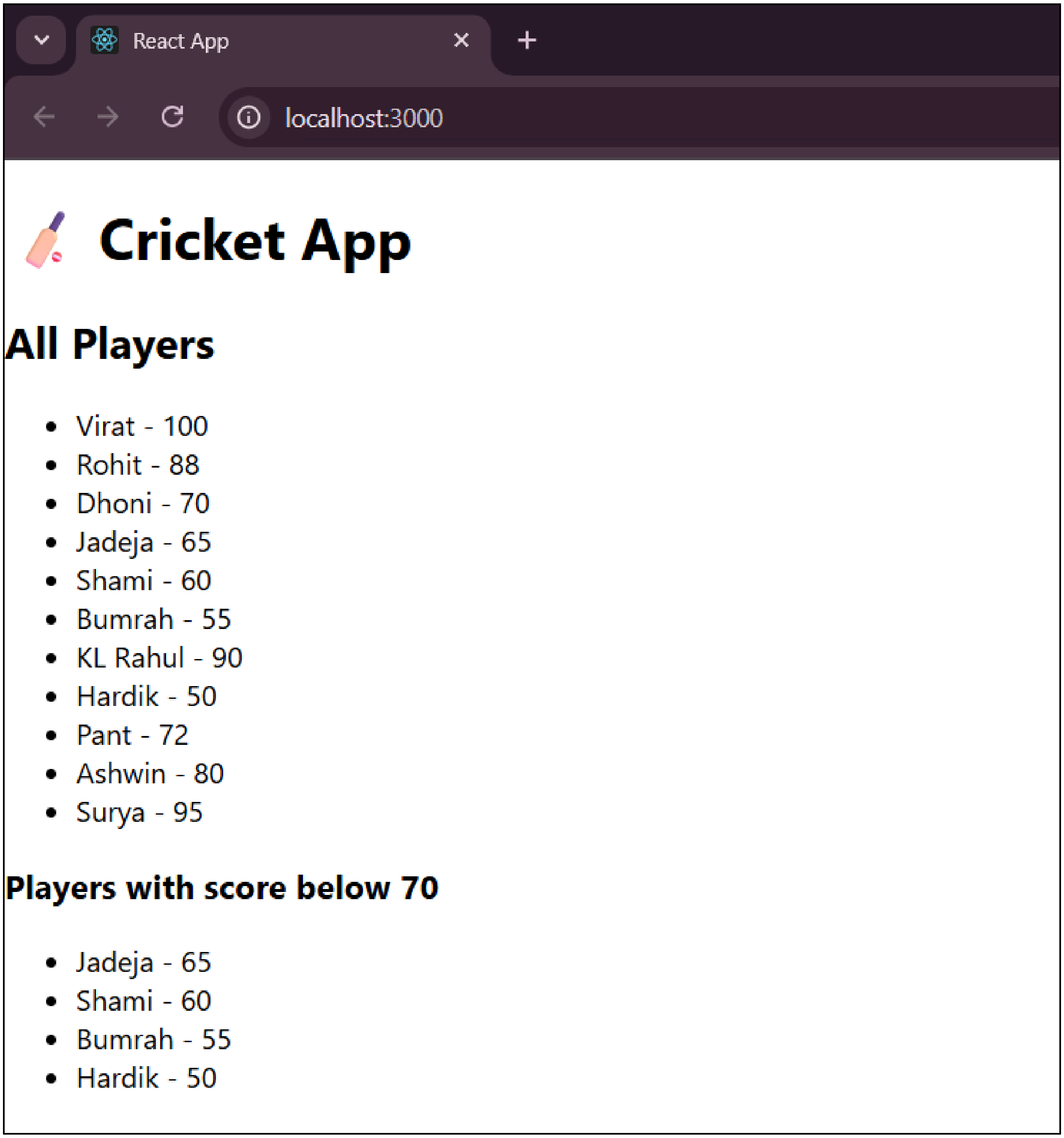
```
import React from 'react';

function ListOfPlayers() {
  const players = [
    { name: 'Virat', score: 100 },
    { name: 'Rohit', score: 88 },
    { name: 'Dhoni', score: 70 },
    { name: 'Jadeja', score: 65 },
    { name: 'Shami', score: 60 },
    { name: 'Bumrah', score: 55 },
    { name: 'KL Rahul', score: 90 },
    { name: 'Hardik', score: 50 },
    { name: 'Pant', score: 72 },
    { name: 'Ashwin', score: 80 },
    { name: 'Surya', score: 95 }
  ];

  const below70 = players.filter(player => player.score < 70);

  return (
    <div>
      <h2>All Players</h2>
      <ul>
        {players.map((player, index) => (
          <li key={index}>{player.name} - {player.score}</li>
        ))}
      </ul>

      <h3>Players with score below 70</h3>
      <ul>
        {below70.map((player, index) => (
          <li key={index}>{player.name} - {player.score}</li>
        ))}
      </ul>
    </div>
  );
}

export default ListOfPlayers;
```

IndianPlayers.js

```
import React from 'react';

function IndianPlayers() {
  const allPlayers = ['Virat', 'Rohit', 'Dhoni', 'Jadeja', 'Bumrah', 'KL
Rahul'];

  // Destructuring to create odd and even team
  const oddTeam = allPlayers.filter((_, index) => index % 2 === 0);
  const evenTeam = allPlayers.filter((_, index) => index % 2 !== 0);

  // Merging T20 and RanjiTrophy players
  const T20players = ['Surya', 'Pant', 'Hardik'];
  const RanjiTrophyPlayers = ['Pujara', 'Rahane'];

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

  return (
    <div>
      <h2>Odd Team Players</h2>
      <ul>
        {oddTeam.map((player, index) => (
          <li key={index}>{player}</li>
        ))}
      </ul>

      <h2>Even Team Players</h2>
      <ul>
        {evenTeam.map((player, index) => (
          <li key={index}>{player}</li>
        ))}
      </ul>

      <h2>Merged Players (T20 + Ranji)</h2>
      <ul>
        {mergedPlayers.map((player, index) => (
          <li key={index}>{player}</li>
        ))}
      </ul>
    </div>
  );
}

export default IndianPlayers;
```

<u>Output:</u>

When flag = true;



🏏 Cricket App

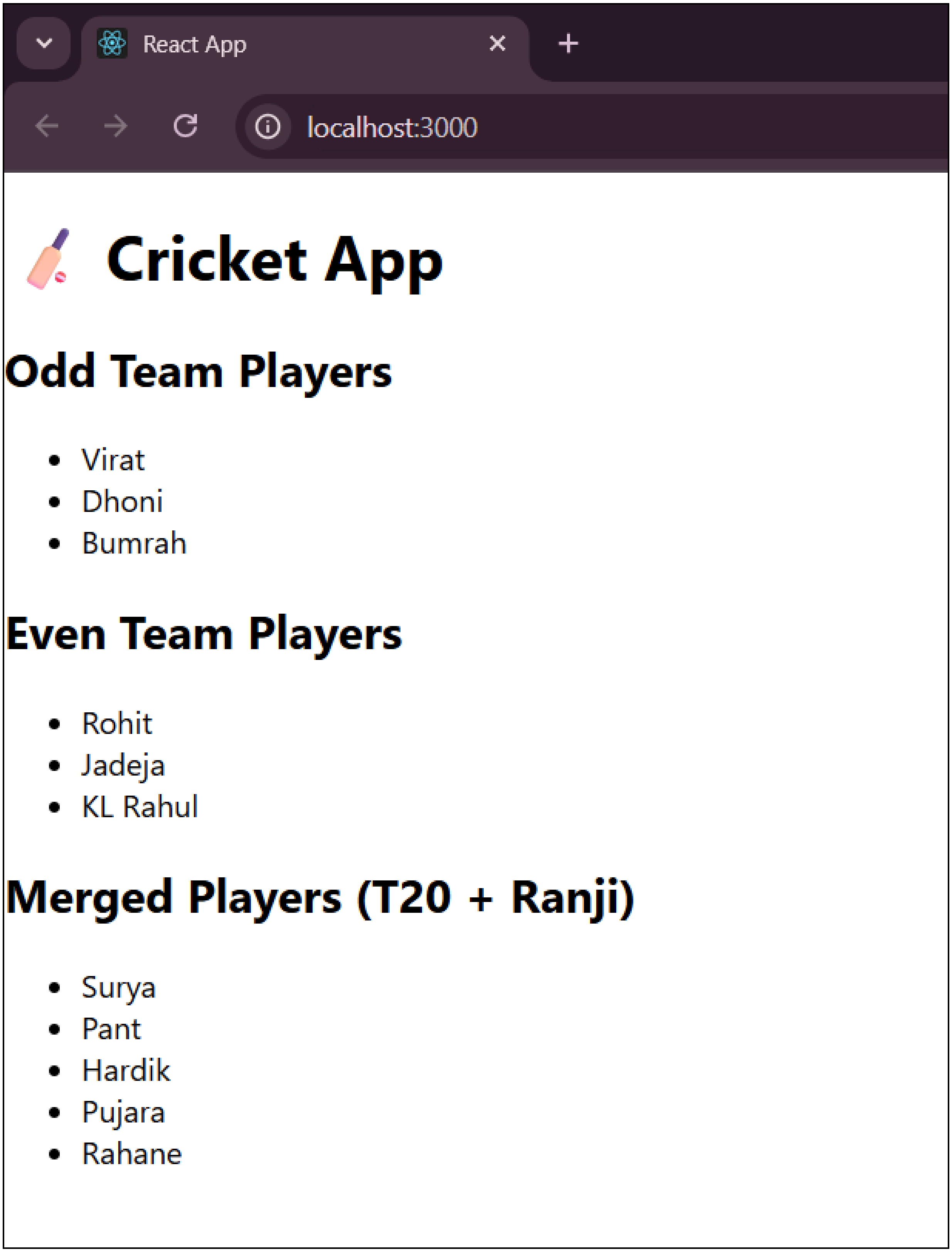
All Players

- Virat - 100
- Rohit - 88
- Dhoni - 70
- Jadeja - 65
- Shami - 60
- Bumrah - 55
- KL Rahul - 90
- Hardik - 50
- Pant - 72
- Ashwin - 80
- Surya - 95

Players with score below 70

- Jadeja - 65
- Shami - 60
- Bumrah - 55
- Hardik - 50

When flag = false;



🏏 Cricket App

Odd Team Players

- Virat
- Dhoni
- Bumrah

Even Team Players

- Rohit
- Jadeja
- KL Rahul

Merged Players (T20 + Ranji)

- Surya
- Pant
- Hardik
- Pujara
- Rahane

Exercise 2:

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

Create an element to display the heading of the page.

Attribute 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.

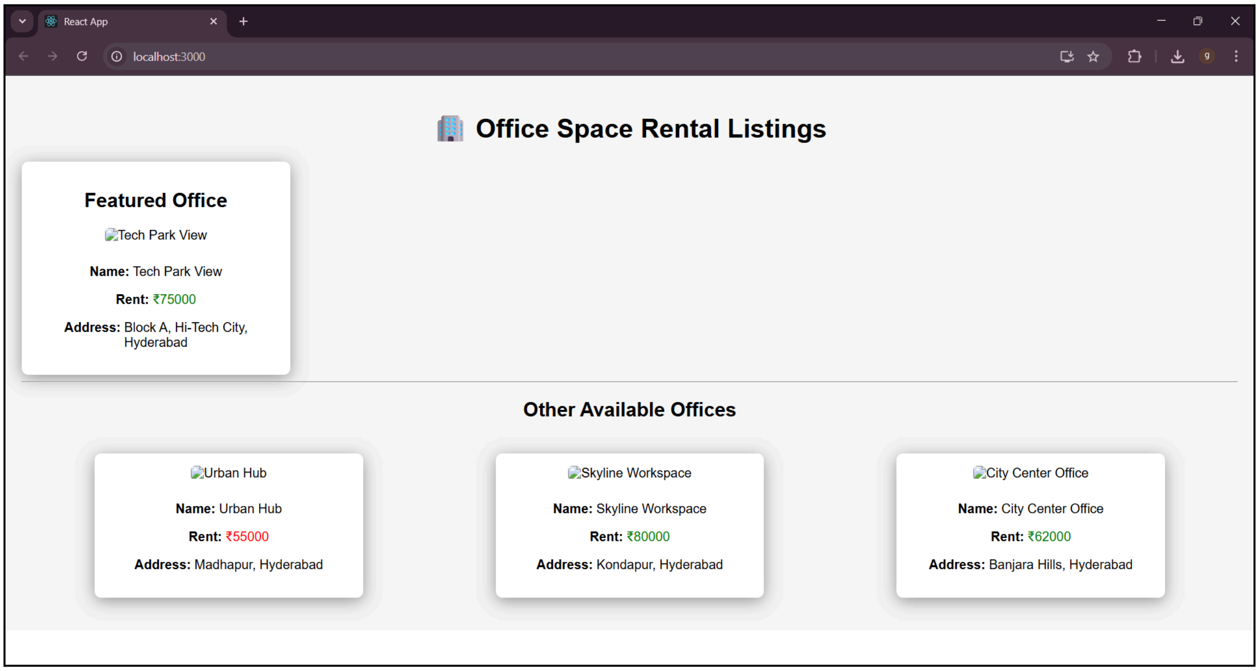
App.js

```
import React from 'react';
import './App.css';

function App() {
  // Heading JSX
  const heading = <h1>🏢 Office Space Rental Listings</h1>;

  // Office object
  const featuredOffice = {
    name: "Tech Park View",
    rent: 75000,
    address: "Block A, Hi-Tech City, Hyderabad",
    imageUrl: "https://via.placeholder.com/300x200?text=Office+Space"
  };

  // List of office space objects
  const officeList = [
    {
      name: "Urban Hub",
      rent: 55000,
      address: "Madhapur, Hyderabad",
      imageUrl: "https://via.placeholder.com/300x200?text=Urban+Hub"
    },
    {
      name: "Skyline Workspace",
      rent: 80000,
      address: "Kondapur, Hyderabad",
      imageUrl:
"https://via.placeholder.com/300x200?text=Skyline+Workspace"
    },
    {
      name: "City Center Office",
      rent: 62000,
      address: "Banjara Hills, Hyderabad",
      imageUrl:
"https://via.placeholder.com/300x200?text=City+Center+Office"
    }
  ];

  return (
    <div className="App">
      {heading}

      <div className="office-card">
        <h2>Featured Office</h2>
        <img src={featuredOffice.imageUrl} alt={featuredOffice.name} />
        <p><strong>Name:</strong> {featuredOffice.name}</p>
        <p>
          <strong>Rent:</strong>{" "}
          <span style={{ color: featuredOffice.rent > 60000 ? 'green' :
'red' }}>
            ₹{featuredOffice.rent}
          </span>
        </p>
        <p><strong>Address:</strong> {featuredOffice.address}</p>
      </div>


      <hr />

      <h2>Other Available Offices</h2>
      <div className='card-container office-card'>
        {officeList.map((office, index) => (
        <div key={index} className="office">
          <img src={office.imageUrl} alt={office.name} />
          <p><strong>Name:</strong> {office.name}</p>
          <p>
            <strong>Rent:</strong>{" "}
            <span style={{ color: office.rent > 60000 ? 'green' : 'red'
}}>
              ₹{office.rent}
            </span>
          </p>
          <p><strong>Address:</strong> {office.address}</p>
        </div>
      ))}
      </div>
    </div>
  );
}

export default App;
```

App.css

```
.App {
  text-align: center;
  font-family: Arial, sans-serif;
  padding: 20px;
  background: whitesmoke;
}

/* Container for grid of cards */
.card-container {
  display: grid;
  grid-template-columns: repeat(3, 1fr); /* force 3 cards per row */
  gap: 24px;
  padding: 20px;
  justify-items: center;
}


/* Single card style */
.office-card {
  background-color: #fff;
  border-radius: 8px;
  overflow: hidden;
  width: 300px;
  box-shadow: 0 4px 20px rgba(0, 0, 0, 0.4); /* darker shadow */
  transition: transform 0.2s ease-in-out;
  padding: 15px;
}

.office-card:hover {
  transform: scale(1.02);
}

.office-card img {
  width: 100%;
  height: 180px;
  object-fit: cover;
  border-radius: 6px;
  margin-bottom: 10px;
}
```

Output:



🏢 Office Space Rental Listings

Featured Office

Tech Park View

Name: Tech Park View

Rent: ₹75000

Address: Block A, Hi-Tech City, Hyderabad

Other Available Offices

Urban Hub

Name: Urban Hub

Rent: ₹55000

Address: Madhapur, Hyderabad

Skyline Workspace

Name: Skyline Workspace

Rent: ₹80000

Address: Kondapur, Hyderabad

City Center Office

Name: City Center Office

Rent: ₹62000

Address: Banjara Hills, Hyderabad

Exercise 3:

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

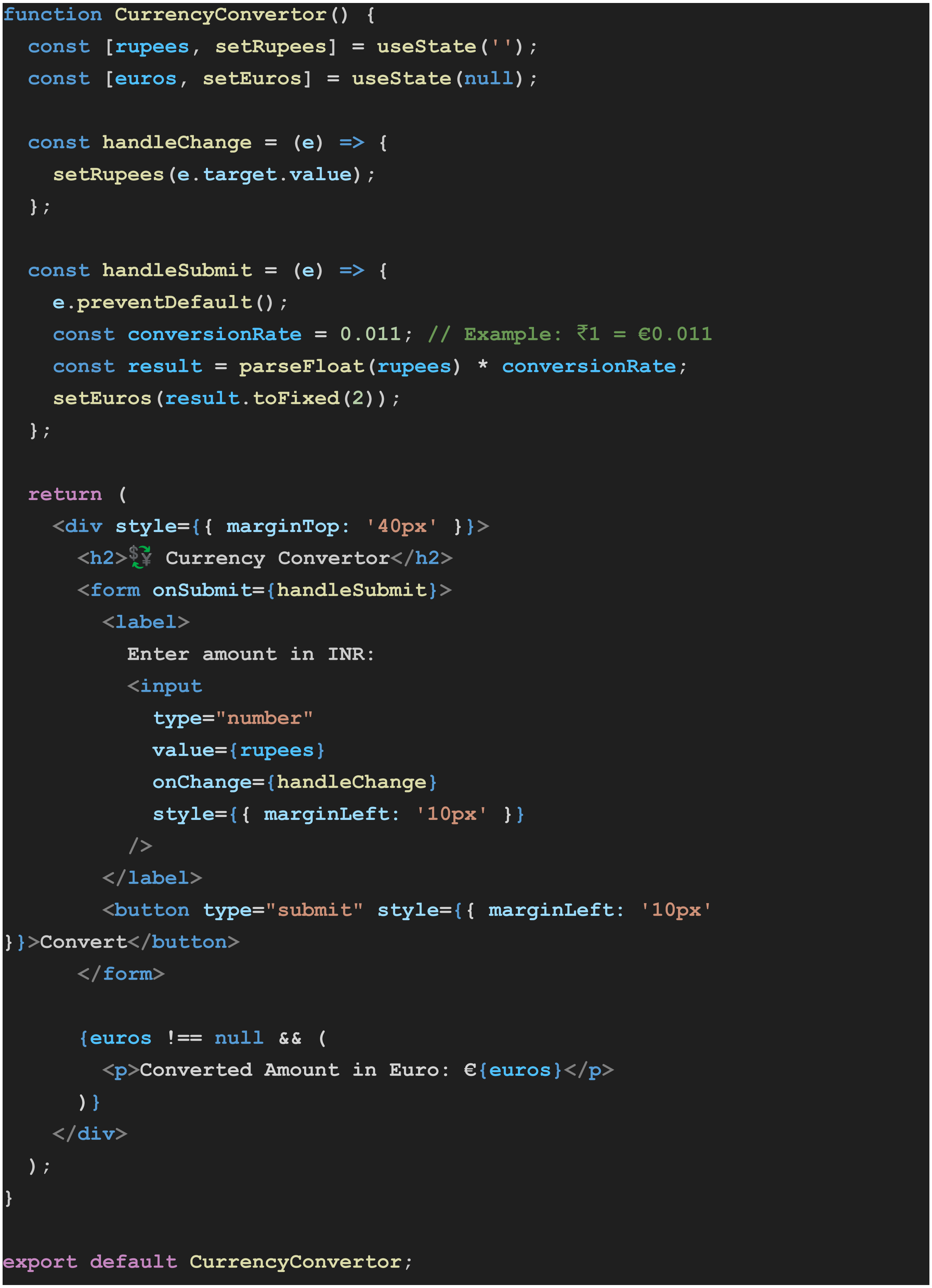
1. Create “Increment” button to increase the value of the counter and “Decrement” button to decrease the value of the counter. The “Increase” button should invoke multiple methods.
   a. To increment the value
   b. Say Hello followed by a static message.
2. Create a button “Say Welcome” which invokes the function which takes “welcome” as an argument.
3. Create a button which invokes synthetic event “OnPress” which display “I was clicked”
4. Create a “CurrencyConvertor” component which will convert the Indian Rupees to Euro when the Convert button is clicked.
5. Handle the Click event of the button to invoke the handleSubmit event and handle the conversion of the euro to rupees.

App.js

```
import React, { useState } from 'react';
import './App.css';
import CurrencyConvertor from './components/CurrencyConvertor';

function App() {
  const [counter, setCounter] = useState(0);

  // --- MULTIPLE FUNCTIONS ---
  const increment = () => {
    setCounter(prev => prev + 1);
  };

  const sayHello = () => {
    alert("Hello! You clicked increment.");
  };

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

  const decrement = () => {
    setCounter(prev => prev - 1);
  };

  // --- FUNCTION WITH ARGUMENT ---
  const sayMessage = (msg) => {
    alert(msg);
  };

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

  return (
    <div className="App">
      <h1>🎯 Event Examples App</h1>

      <h2>Counter: {counter}</h2>
      <button onClick={handleIncrementClick}>Increment</button>
      <button onClick={decrement}>Decrement</button>

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

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

      <br /><br />
      <CurrencyConvertor />
    </div>
  );
}

export default App;
```

CurrencyConvertor.js

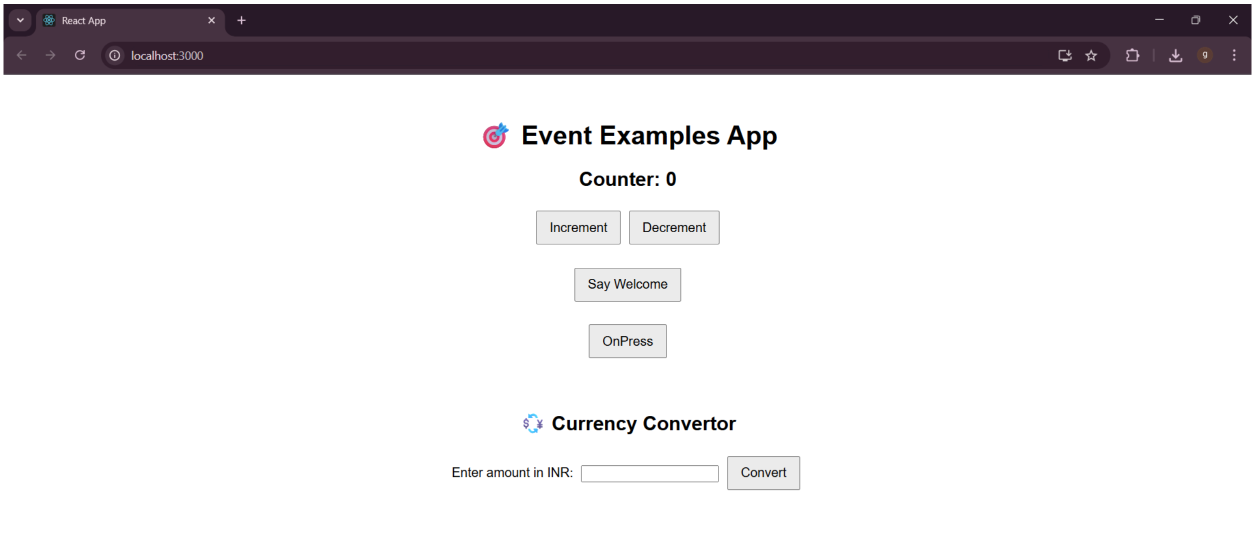
```
import React, { useState } from 'react';
```



```
function CurrencyConvertor() {
  const [rupees, setRupees] = useState('');
  const [euros, setEuros] = useState(null);

  const handleChange = (e) => {
    setRupees(e.target.value);
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    const conversionRate = 0.011; // Example: ₹1 = €0.011
    const result = parseFloat(rupees) * conversionRate;
    setEuros(result.toFixed(2));
  };

  return (
    <div style={{ marginTop: '40px' }}>
      <h2>💱 Currency Convertor</h2>
      <form onSubmit={handleSubmit}>
        <label>
          Enter amount in INR:
          <input
            type="number"
            value={rupees}
            onChange={handleChange}
            style={{ marginLeft: '10px' }}
          />
        </label>
        <button type="submit" style={{ marginLeft: '10px'
}}>Convert</button>
      </form>

      {euros !== null && (
        <p>Converted Amount in Euro: €{euros}</p>
      )}
    </div>
  );
}

export default CurrencyConvertor;
```

Output:

Initial state

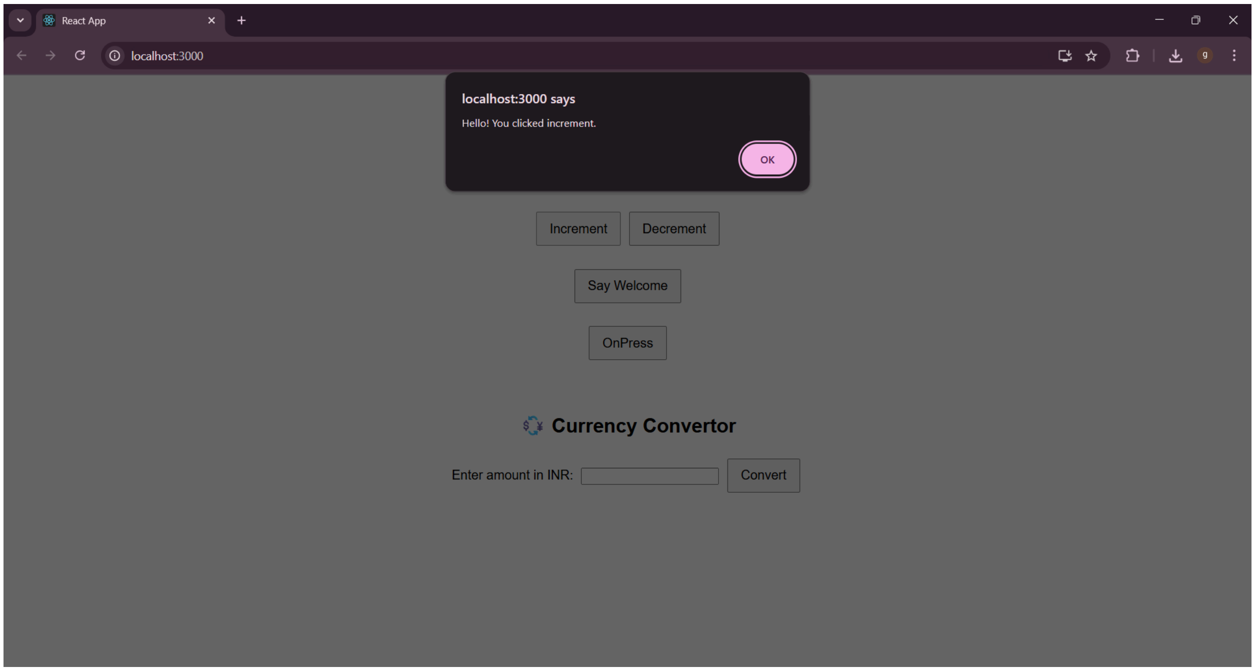


🎯 Event Examples App

Counter: 0

Enter amount in INR:

Increment

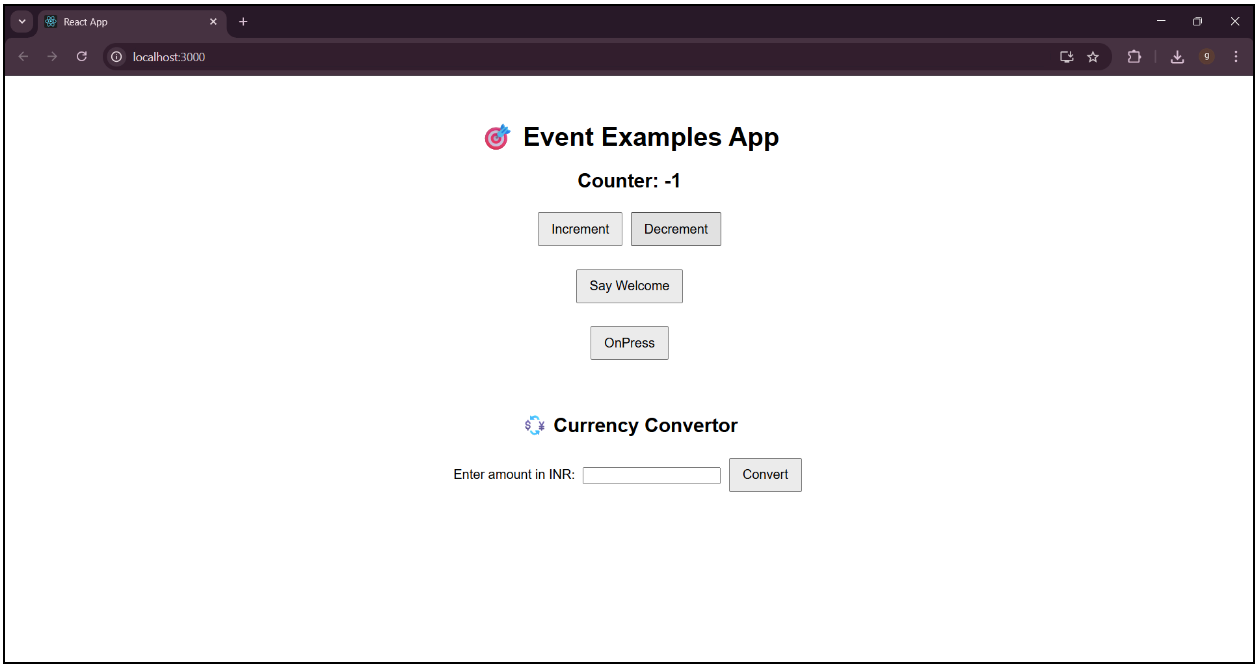


localhost:3000 says

Hello! You clicked increment.

Enter amount in INR:

Decrement



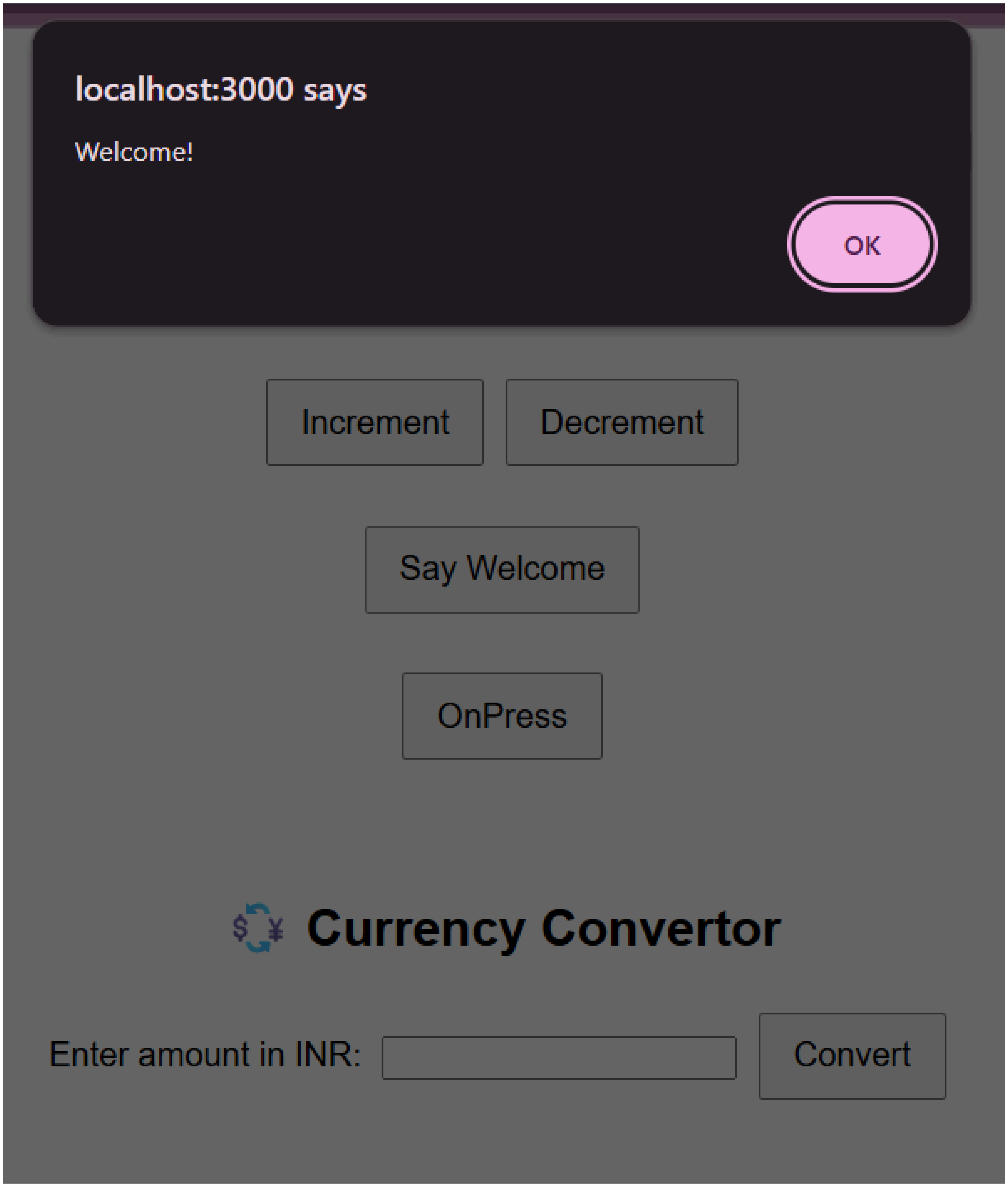
🎯 Event Examples App

Counter: -1

💱 Currency Convertor

Enter amount in INR:

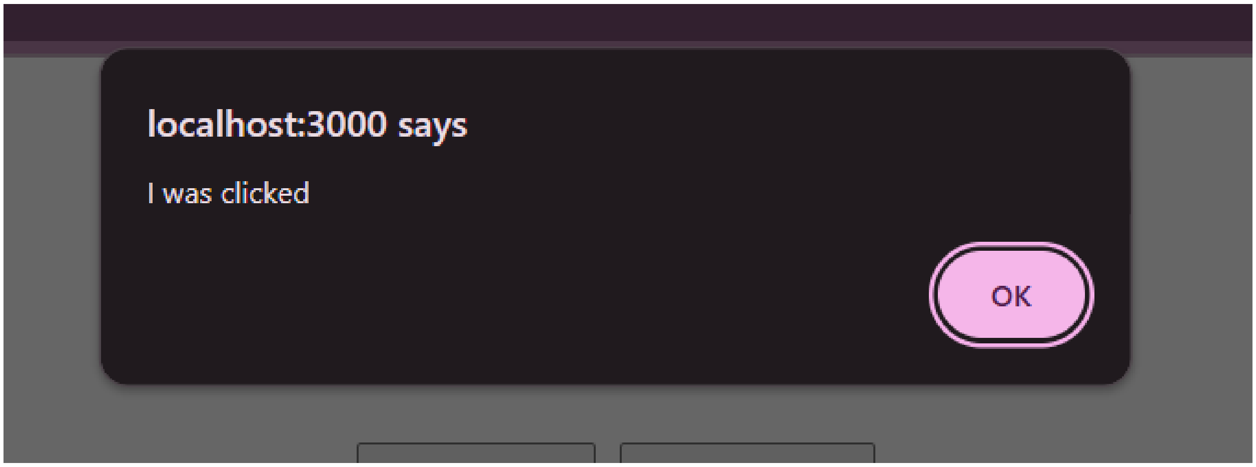
Welcome



localhost:3000 says

Welcome!

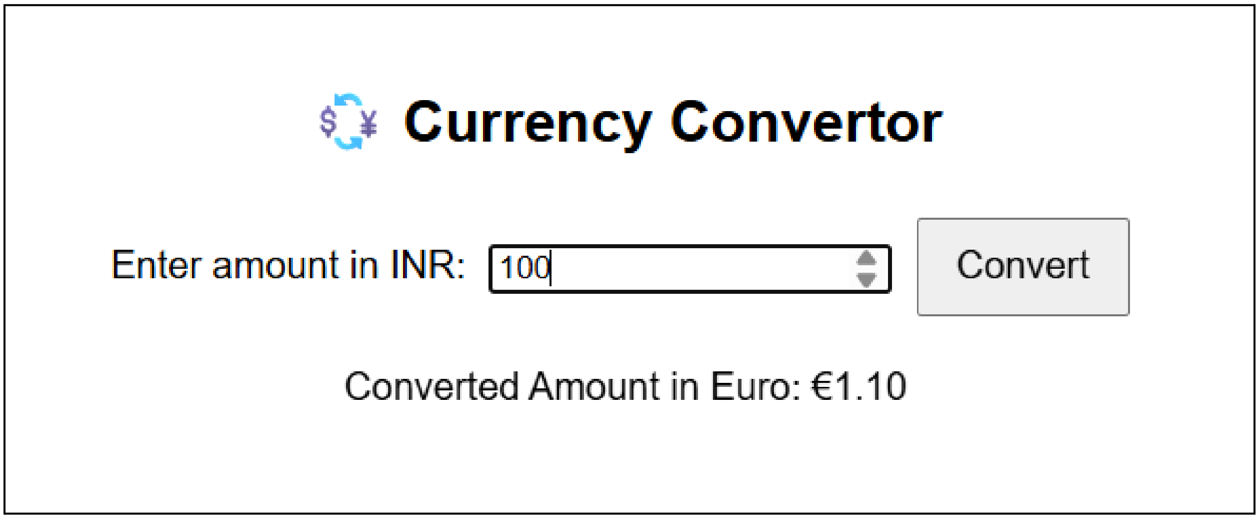
Onpress



localhost:3000 says

I was clicked

Currency Convertor



Currency Convertor

Enter amount in INR: 100

Converted Amount in Euro: €1.10

Exercise 4:

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.

App.js

```
import React, { useState } from 'react';
import './App.css';
import GuestPage from './components/GuestPage';
import UserPage from './components/UserPage';

function App() {
  const [loggedIn, setLoggedIn] = useState(false);

  const handleAuth = () => {
    setLoggedIn(prev => !prev);
  };

  return (
    <div className="App">
      <header>
        <h1>✈️ Ticket Booking App</h1>
        <button onClick={handleAuth}>
          {loggedIn ? 'Logout' : 'Login'}
        </button>
      </header>

      <hr />

      {loggedIn ? <UserPage /> : <GuestPage />}
    </div>
  );
}

export default App;
```

FlightList.js

```
import React from 'react';

const flights = [
  { id: 1, flight: "IndiGo 6E 123", from: "Hyderabad", to: "Delhi", time:
"08:00 AM" },
  { id: 2, flight: "Air India AI 456", from: "Mumbai", to: "Chennai",
time: "10:30 AM" },
  { id: 3, flight: "Vistara UK 789", from: "Bangalore", to: "Kolkata",
time: "03:45 PM" },
];
```

```
function FlightList() {
  return (
    <div>
      <h2>Available Flights</h2>
      <ul>
        {flights.map(flight => (
          <li key={flight.id}>
            ✈️ {flight.flight} - {flight.from} to {flight.to} at {flight.time}
          </li>
        ))}
      </ul>
    </div>
  );
}

export default FlightList;
```

UserPage.js

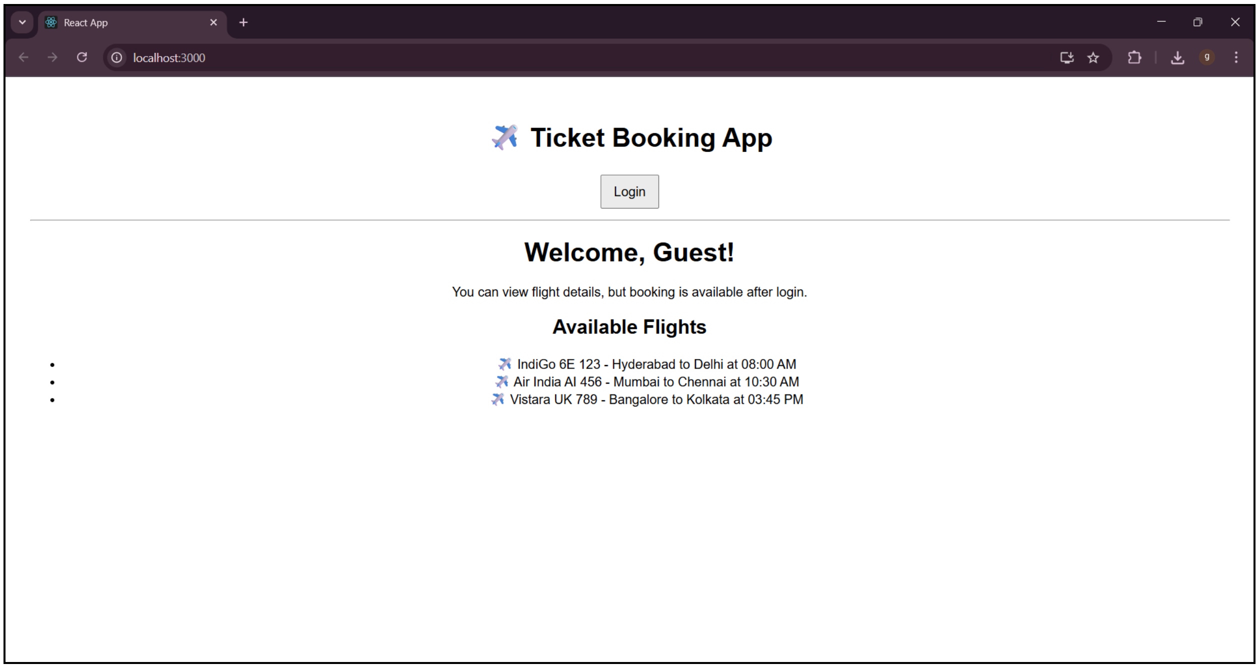
```
import React from 'react';
import FlightList from './FlightList';

function UserPage() {
  return (
    <div>
      <h1>Welcome, User!</h1>
      <p>You can now book tickets for the following flights:</p>
      <FlightList />

      <div style={{ marginTop: '20px' }}>
        <h3>Booking Form</h3>
        <form onSubmit={(e) => {
          e.preventDefault();
          alert("Ticket booked successfully!");
        }}>
          <label>
            Select Flight ID:
            <input type="number" min="1" max="3" required />
          </label>
          <br /><br />
          <button type="submit">Book Ticket</button>
        </form>
      </div>
    </div>
  );
}

export default UserPage;
```

GuestPage.js

```
import React from 'react';
import FlightList from './FlightList';

function GuestPage() {
  return (
    <div>
      <h1>Welcome, Guest!</h1>
      <p>You can view flight details, but booking is available after login.</p>
      <FlightList />
    </div>
  );
}

export default GuestPage;
```

Output:

Initial state



✈️ Ticket Booking App

Login

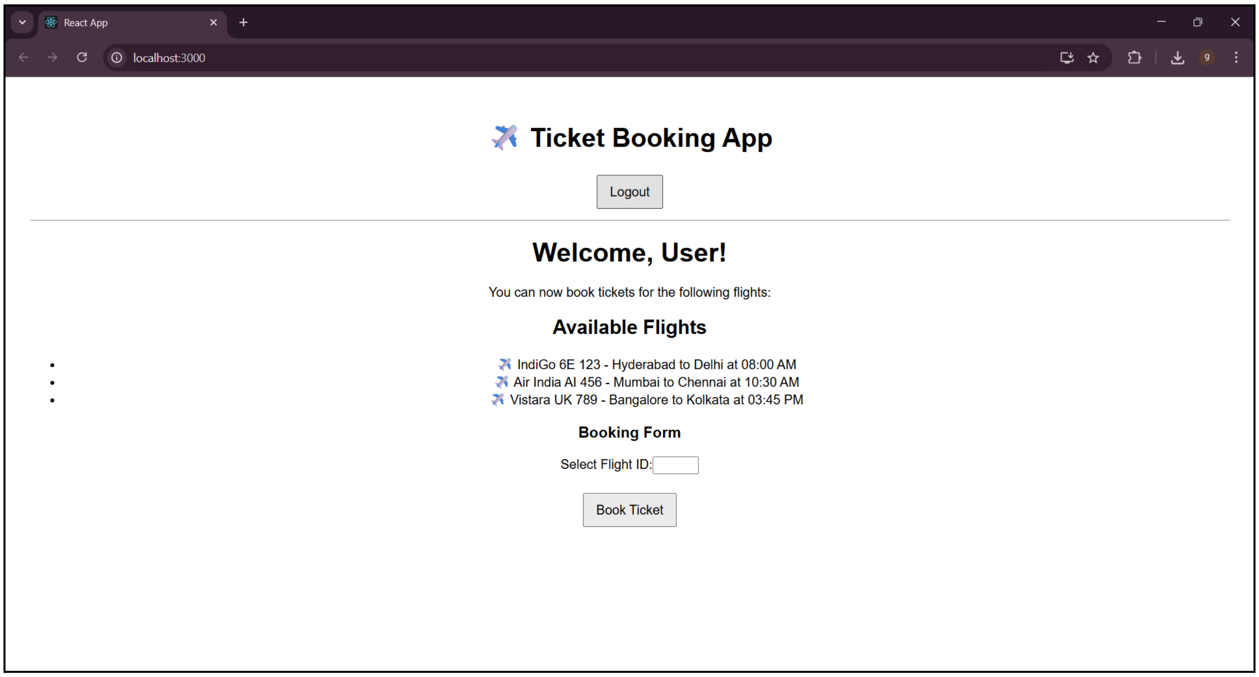
Welcome, Guest!

You can view flight details, but booking is available after login.

Available Flights

- ✈️ IndiGo 6E 123 - Hyderabad to Delhi at 08:00 AM
- ✈️ Air India AI 456 - Mumbai to Chennai at 10:30 AM
- ✈️ Vistara UK 789 - Bangalore to Kolkata at 03:45 PM

After Login



✈️ Ticket Booking App

Logout

Welcome, User!

You can now book tickets for the following flights:

Available Flights

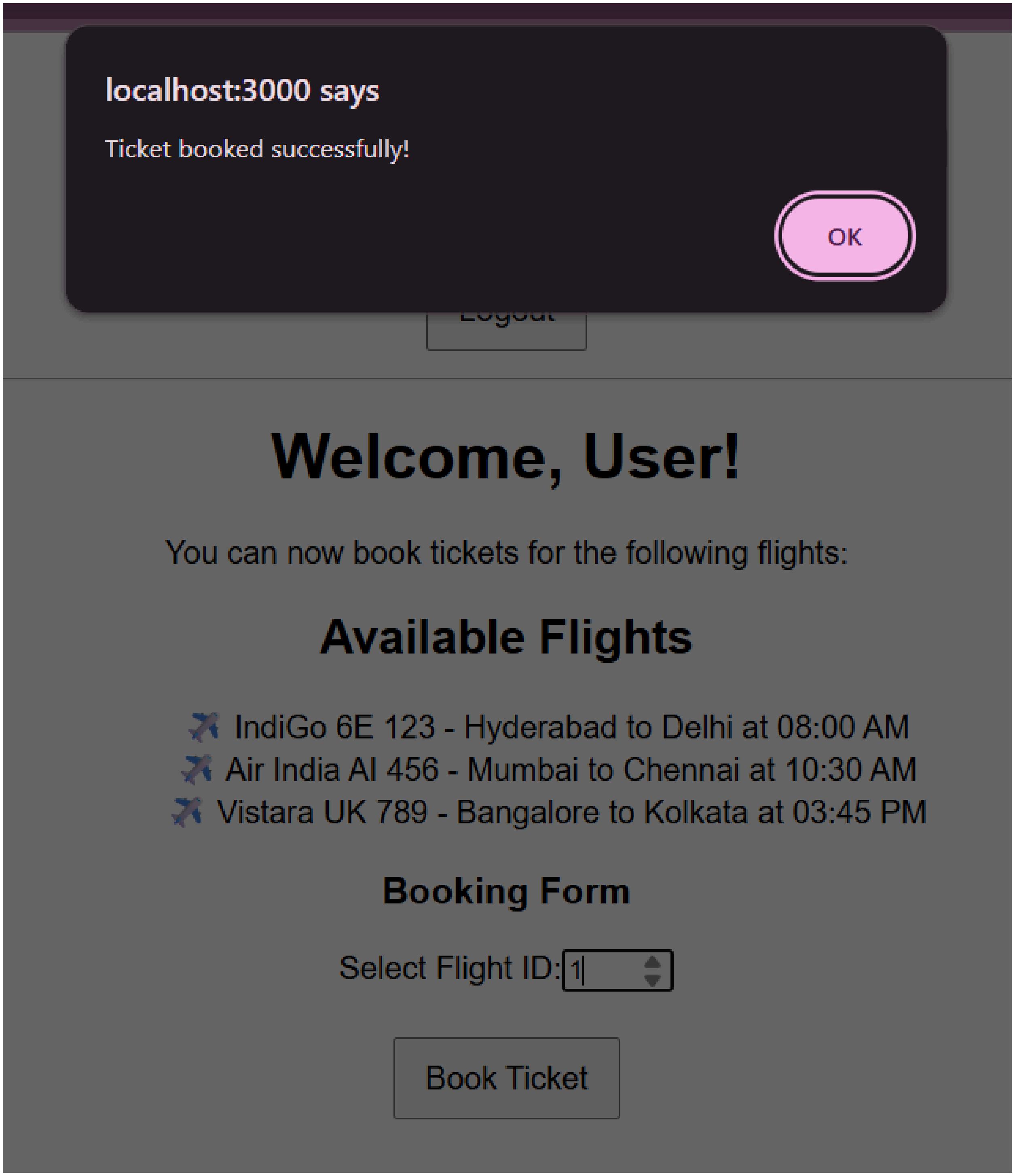
- ✈️ IndiGo 6E 123 - Hyderabad to Delhi at 08:00 AM
- ✈️ Air India AI 456 - Mumbai to Chennai at 10:30 AM
- ✈️ Vistara UK 789 - Bangalore to Kolkata at 03:45 PM

Booking Form

Select Flight ID:

Book Ticket

After Entering Flight Serial Number



localhost:3000 says

Ticket booked successfully!

Welcome, User!

You can now book tickets for the following flights:

Available Flights

✈️ IndiGo 6E 123 - Hyderabad to Delhi at 08:00 AM
✈️ Air India AI 456 - Mumbai to Chennai at 10:30 AM
✈️ Vistara UK 789 - Bangalore to Kolkata at 03:45 PM

Booking Form

Select Flight ID: 1

Exercise 5:

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.

App.js

```
import React, { useState } from 'react';
import './App.css';
import BookDetails from './components/BookDetails';
import BlogDetails from './components/BlogDetails';
import CourseDetails from './components/CourseDetails';

function App() {
  const [view, setView] = useState("book");
  const [showBlog, setShowBlog] = useState(false);
  const [showCourse, setShowCourse] = useState(true);

  // Method 1: if...else rendering
  const renderUsingIfElse = () => {
    if (view === "book") {
      return <BookDetails />;
    } else if (view === "blog") {
      return <BlogDetails />;
    } else {
      return <CourseDetails />;
    }
  };

  // Method 2: Switch rendering
  const renderUsingSwitch = () => {
    switch (view) {
      case "book": return <BookDetails />;
      case "blog": return <BlogDetails />;
      case "course": return <CourseDetails />;
      default: return <p>No view selected</p>;
    }
  };
  return (
    <div className="App">
      <h1>📖 Blogger App</h1>

      <div style={{ marginBottom: '20px' }}>
        <button onClick={() => setView("book")}>Show Book</button>
        <button onClick={() => setView("blog")}>Show Blog</button>
        <button onClick={() => setView("course")}>Show Course</button>
      </div>

      <hr />
      <h2>Method 1: Using if...else</h2>
      {renderUsingIfElse()}

      <hr />
      <h2>Method 2: Using && (logical AND)</h2>
      {showBlog && <BlogDetails />}
      <button onClick={() => setShowBlog(!showBlog)}>
        Toggle Blog Component (Currently {showBlog ? "Shown" : "Hidden"})
      </button>

      <hr />
      <h2>Method 3: Using Ternary Operator</h2>
      {showCourse ? <CourseDetails /> : <p>Course component is
hidden.</p>}
      <button onClick={() => setShowCourse(!showCourse)}>
        Toggle Course Component
      </button>

      <hr />
      <h2>Method 4: Using switch case</h2>
      {renderUsingSwitch()}
    </div>
  );
}

export default App;
```

CourseDetails.js

```
import React from 'react';

function CourseDetails() {
  return (
    <div>
      <h2>🎓 Course Details</h2>
      <p>Course: Full Stack Development</p>
      <p>Instructor: Angela Yu</p>
    </div>
  );
}

export default CourseDetails;
```

BlogDetails.js

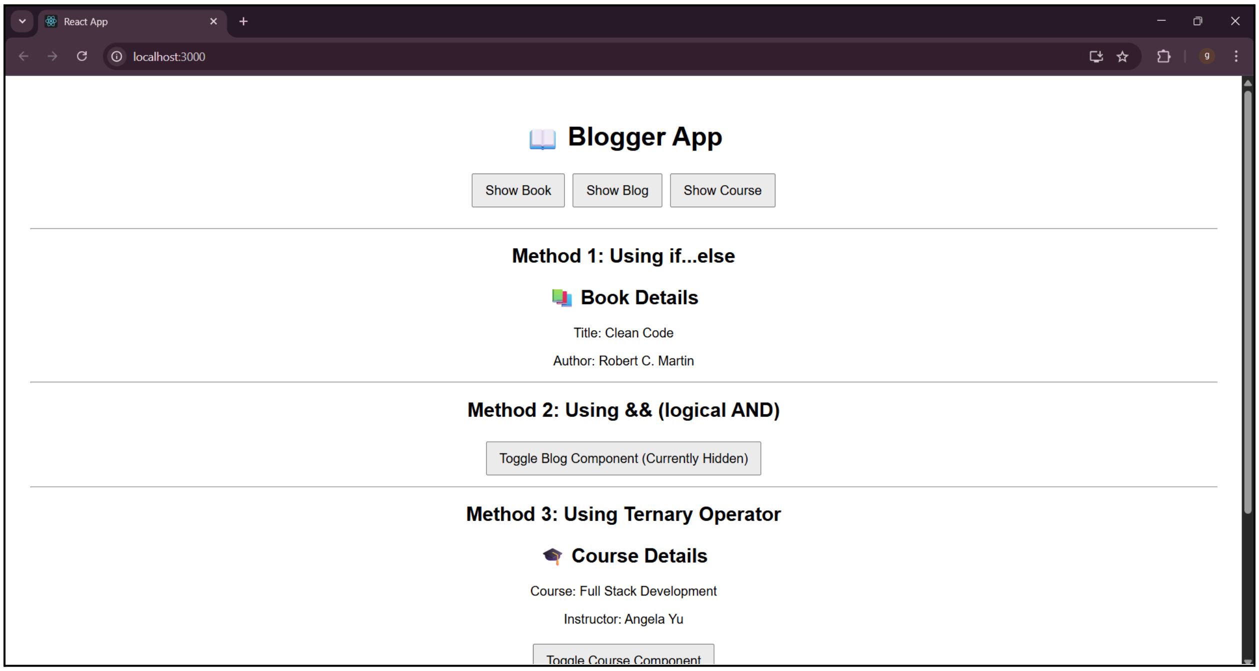
```
import React from 'react';

function BlogDetails() {
  return (
    <div>
      <h2>📝 Blog Details</h2>
      <p>Title: Why React is Awesome</p>
      <p>Author: Dan Abramov</p>
    </div>
  );
}

export default BlogDetails;
```

BookDetails.js

```
import React from 'react';

function BookDetails() {
  return (
    <div>
      <h2>📚 Book Details</h2>
      <p>Title: Clean Code</p>
      <p>Author: Robert C. Martin</p>
    </div>
  );
}

export default BookDetails;
```

Output:

Initial State/Book Details



📖 Blogger App

Method 1: Using if...else

📚 Book Details

Title: Clean Code

Author: Robert C. Martin

Method 2: Using && (logical AND)

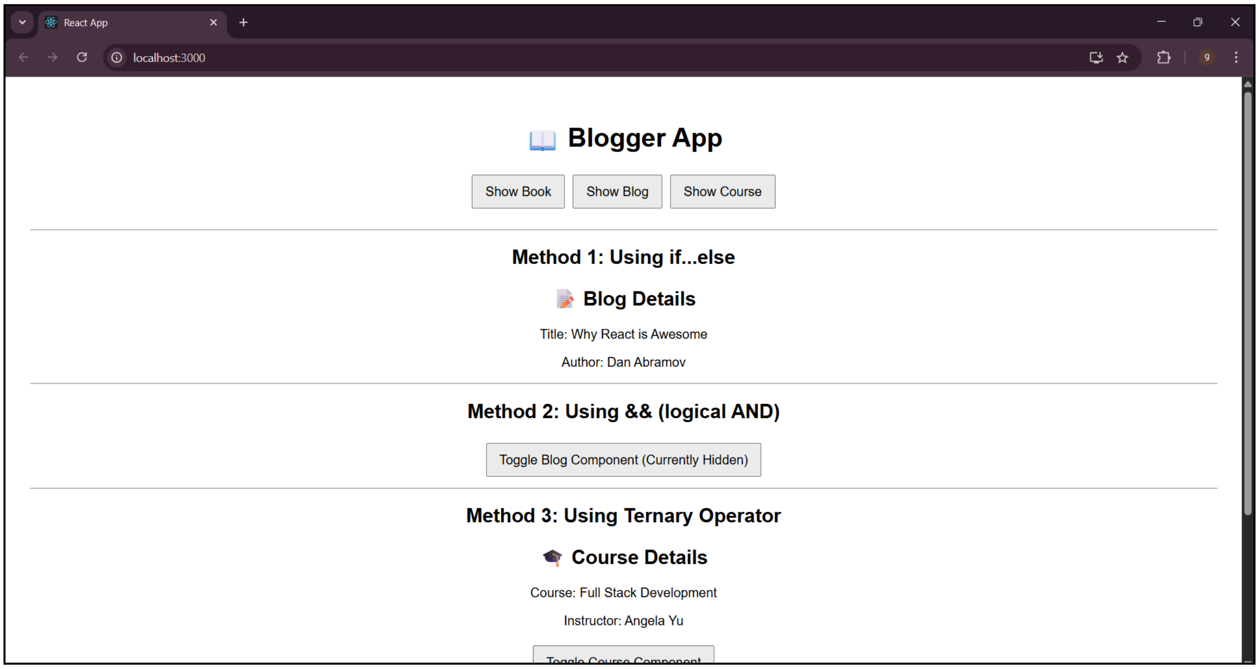
Method 3: Using Ternary Operator

🎓 Course Details

Course: Full Stack Development

Instructor: Angela Yu

Show Blog



📖 Blogger App

Method 1: Using if...else

📝 Blog Details

Title: Why React is Awesome

Author: Dan Abramov

Method 2: Using && (logical AND)

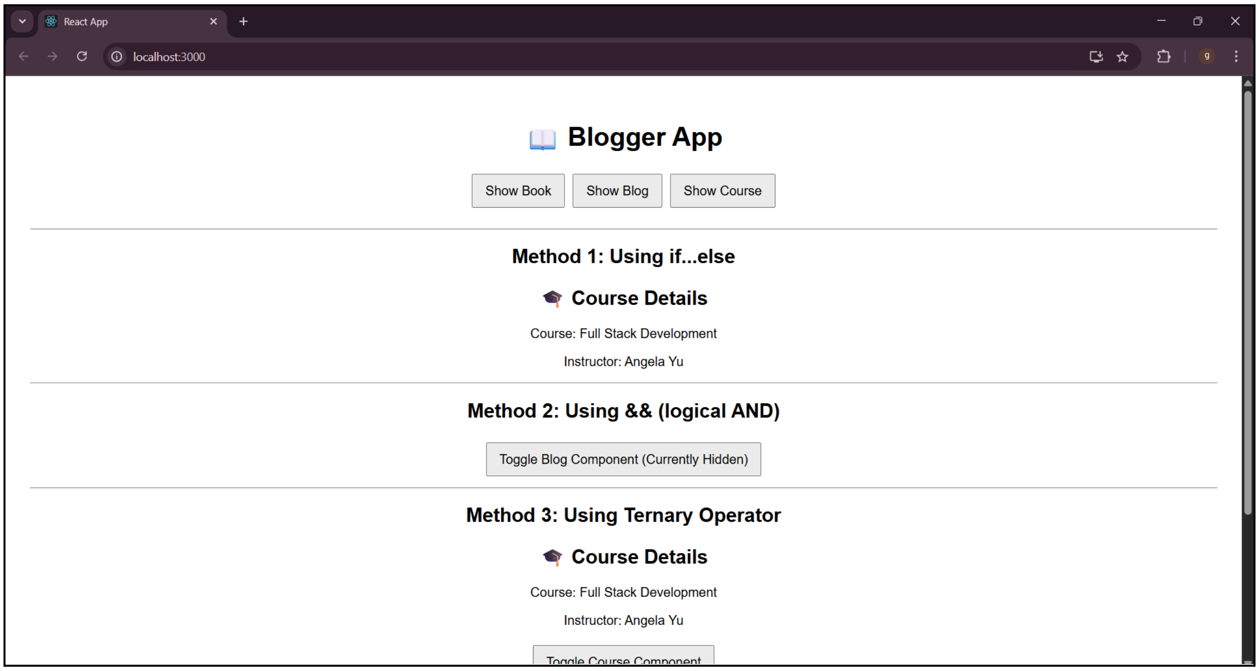
Method 3: Using Ternary Operator

🎓 Course Details

Course: Full Stack Development

Instructor: Angela Yu

Show Course



📖 Blogger App

Method 1: Using if...else

🎓 Course Details

Course: Full Stack Development

Instructor: Angela Yu

Method 2: Using && (logical AND)

Method 3: Using Ternary Operator

🎓 Course Details

Course: Full Stack Development

Instructor: Angela Yu