**9.ES6 Features in React – CricketApp Project**

**1. Features of ES6 (ECMAScript 2015)**

ES6 introduced many improvements to JavaScript, including:

* let and const for block-scoped variable declarations
* Arrow functions for shorter syntax
* Classes and inheritance
* Template literals (backtick strings)
* Default parameters
* Destructuring assignment
* Modules (import, export)
* Promises for async operations
* Map and Set data structures
* Spread (...) and rest parameters

**2. JavaScript let**

* let is used to declare block-scoped variables. Unlike var, it does not allow redeclaration in the same scope and is limited to the block where it is defined.
* js

3. **Difference Between var and let**

| **Feature** | **var** | **let** |
| --- | --- | --- |
| Scope | Function-scoped | Block-scoped |
| Hoisting | Hoisted and initialized to undefined | Hoisted but not initialized |
| Redeclaration | Allowed | Not allowed in the same scope |

**4. JavaScript const**

const is used to declare constants. These cannot be reassigned after initialization. It is also block-scoped like let.

**5. ES6 Class Fundamentals**

Classes in ES6 provide a cleaner syntax for creating constructor functions and handling inheritance. They are syntactic sugar over prototype-based inheritance.

**6. ES6 Class Inheritance**

ES6 classes can inherit from other classes using the extends keyword, and super() to call the parent class constructor.

**7. Arrow Functions**

Arrow functions provide a shorter syntax for writing functions and do not bind their own this.

**8. Set in ES6**

A Set is a collection of unique values. It automatically removes duplicates.

**9. Map in ES6**

A Map is a key-value pair collection that remembers the insertion order of keys.

**10. Using map() Method**

The map() method creates a new array by applying a function to each element of an existing array.

**11. Using Arrow Functions**

Arrow functions provide compact syntax, often used in array operations like:

const players = [50, 70, 90];

const topScorers = players.filter(score => score >= 70);

**12. Destructuring Features**

Destructuring allows unpacking values from arrays or objects into distinct variables.

const [a, b] = [1, 2]; // Array destructuring

const {name, age} = {name: "Virat", age: 34}; // Object destructuring

**Project Setup**

1. Open terminal or CMD
2. Create React App:

npx create-react-app cricketapp

cd cricketapp

3.Create folder structure:

* Inside src, create a folder: components
* Inside components, create:
  + ListofPlayers.js
  + IndianPlayers.js

ListofPlayers.js:

import React from 'react';

const ListofPlayers = () => {

const players = [

{ name: "Virat", score: 95 },

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

{ name: "Dhoni", score: 88 },

{ name: "Kohli", score: 67 },

{ name: "Jadeja", score: 72 },

{ name: "Bumrah", score: 39 },

{ name: "Gill", score: 91 },

{ name: "Hardik", score: 65 },

{ name: "Surya", score: 85 },

{ name: "Pant", score: 70 },

{ name: "Iyer", score: 60 }

]

const allPlayers = players.map((player, index) => (

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

));

const filteredPlayers = players.filter(p => p.score < 70)

.map((player, index) => (

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

));

return (

<div>

<h2>All Players:</h2>

<ul>{allPlayers}</ul>

<h2>Players with score below 70:</h2>

<ul>{filteredPlayers}</ul>

</div>

);

};

export default ListofPlayers;

IndianPlayers.js :

import React from 'react';

const IndianPlayers = () => {

const allPlayers = [

"Virat", "Rohit", "Dhoni", "Kohli", "Jadeja", "Bumrah",

"Gill", "Hardik", "Surya", "Pant", "Iyer"

];

const oddPlayers = allPlayers.filter((\_, i) => i % 2 === 0);

const evenPlayers = allPlayers.filter((\_, i) => i % 2 !== 0);

const [firstOdd, ...restOdd] = oddPlayers;

const [firstEven, ...restEven] = evenPlayers;

const T20players = ["Virat", "Rohit", "Gill"];

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

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

return (

<div>

<h2>Odd Team Players:</h2>

<p>Captain: {firstOdd}</p>

<ul>{restOdd.map((p, i) => <li key={i}>{p}</li>)}</ul>

<h2>Even Team Players:</h2>

<p>Captain: {firstEven}</p>

<ul>{restEven.map((p, i) => <li key={i}>{p}</li>)}</ul>

<h2>Merged Team (T20 + Ranji):</h2>

<ul>{mergedPlayers.map((p, i) => <li key={i}>{p}</li>)}</ul>

</div>

);

};

export default IndianPlayers;

App.js :

import React from 'react';

import ListofPlayers from './components/ListofPlayers';

import IndianPlayers from './components/IndianPlayers';

function App() {

const flag = false; // set to true or false to toggle output

return (

<div className="App">

<h1>Welcome to Cricket App</h1>

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

</div>

);

}

export default App;

**Open terminal:**

npm start

App will open in browser at: http://localhost:3001

Toggle between components:

* + In App.js, set flag = true → shows ListofPlayers
  + Set flag = false → shows IndianPlayers

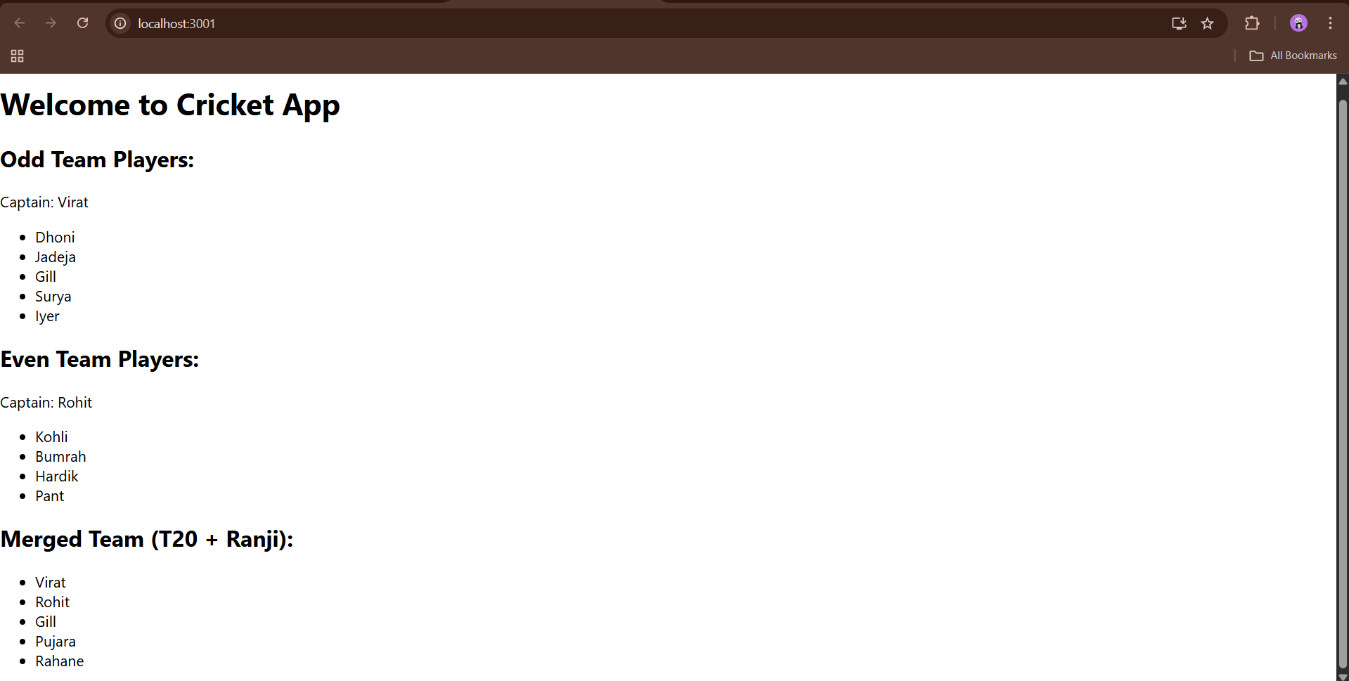
**Expected Output**

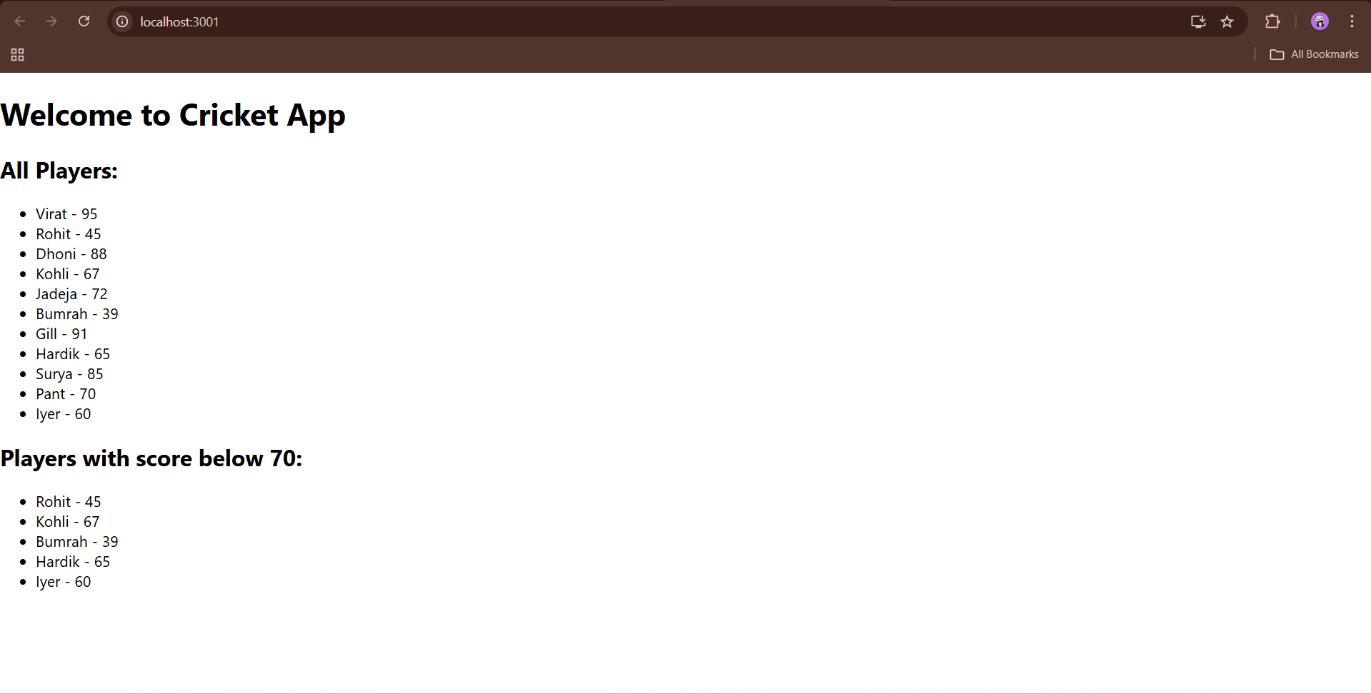
**When flag = true:**

* All players listed
* Filtered players with score < 70 using arrow functions

**When flag = false:**

* Odd & Even team with captain using destructuring
* Merged array of T20 and Ranji players using spread (...)





**10.Office Space Rental App**

**Definitions of Objectives**

**1. Define JSX**

**JSX (JavaScript XML)** is a syntax extension for JavaScript used in React to describe the UI. It looks like HTML but gets transpiled to React.createElement() calls. JSX makes it easier to visualize the structure of the UI directly in the JavaScript code.

const heading = <h1>Office Space Rental App</h1>;

**2. Explain ECMA Script**

**ECMAScript (ES)** is the standardized scripting language specification that JavaScript is based on. Versions like ES5, ES6 (also called ES2015), and beyond introduce new features such as let/const, arrow functions, classes, modules, promises, and more. React apps commonly use modern ES6+ syntax.

**3. Explain React.createElement()**

React.createElement() is a core React API used to create virtual DOM elements. JSX is syntactic sugar that compiles to React.createElement() calls.

Example:

React.createElement("h1", null, "Welcome")

**4. Explain How to Create React Nodes with JSX**

React nodes (UI elements) can be created using JSX by embedding tags in JavaScript. These nodes can include HTML elements, components, or fragments.

Example:

const officeHeading = <h2>Available Spaces</h2>;

**5. Define How to Render JSX to the DOM**

To display JSX elements on the page, React provides the ReactDOM.render() function which renders a React element into the actual DOM.

Example:

ReactDOM.render(<App />, document.getElementById("root"));

**6. Explain How to Use JavaScript Expressions in JSX**

JSX allows embedding JavaScript expressions inside {} brackets.

Example:

const name = "Bangalore Hub";

const element = <h1>Welcome to {name}</h1>;

**7. Explain How to Use Inline CSS in JSX**

In JSX, inline styles are passed as a JavaScript object with camelCased property names.

Example:

const rentStyle = {

color: rent < 60000 ? "red" : "green"

};

<h2 style={rentStyle}>₹{rent}</h2>

**Step 1: Create React App**

1. Open **Command Prompt** or **Terminal**
2. Create a new React app:

npx create-react-app officespacerentalapp

cd officespacerentalapp

**Step 2: Open in Visual Studio Code**

code .

**Step 3: Modify App.js**

Open src/App.js and replace its contents with:

import React from "react";

function App() {

const officeList = [

{

name: "Workspace One",

rent: 45000,

address: "MG Road, Bangalore",

image: "https://via.placeholder.com/150"

},

{

name: "Startup Hub",

rent: 75000,

address: "Indiranagar, Bangalore",

image: "https://via.placeholder.com/150"

},

{

name: "TechPark View",

rent: 60000,

address: "Electronic City, Bangalore",

image: "https://via.placeholder.com/150"

}

];

const heading = <h1>Office Space Rental App</h1>;

return (

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

{heading}

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

const rentStyle = {

color: office.rent < 60000 ? "red" : "green"

};

return (

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

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

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

<p style={rentStyle}>Rent: ₹{office.rent}</p>

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

</div>

);

})}

</div>

);

}

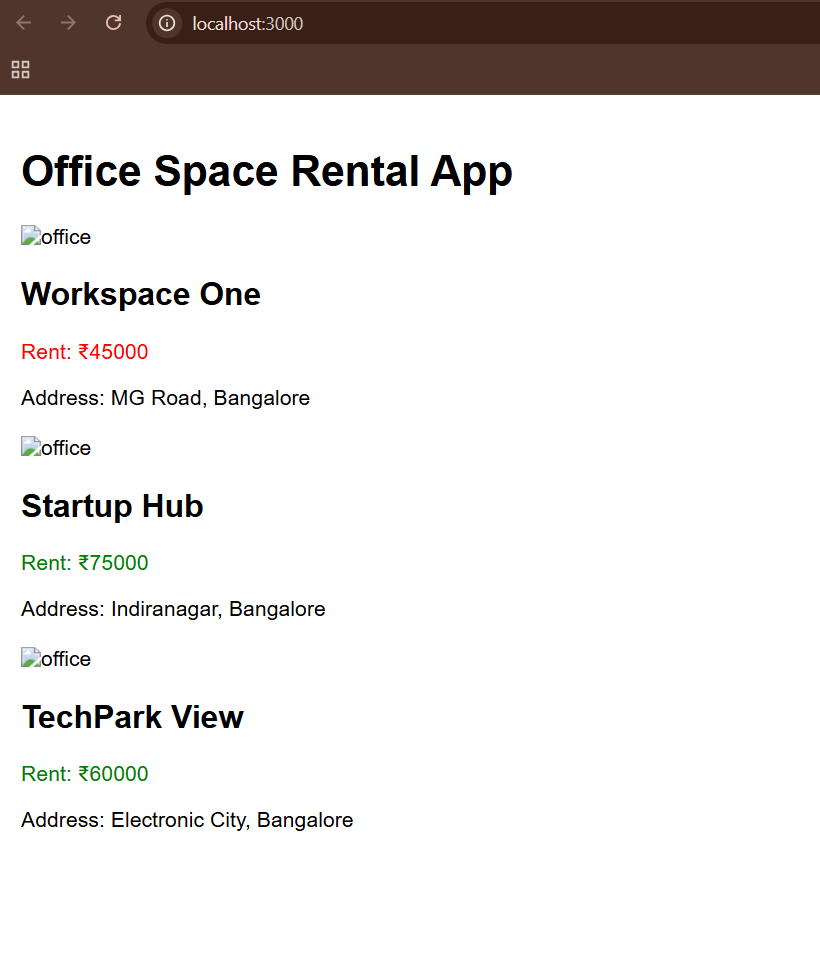
export default App;

**Step 4: Start the Development Server**

**npm start**

**Expected Output**

You should see:

* A heading: Office Space Rental App
* List of 3 office spaces
* Rents:
  + In red if rent < ₹60000
  + In green if rent ≥ ₹60000
  + 

**11.eventexamplesapp**

**Objectives**

**1. React Events**

React uses its own event system known as Synthetic Events that wraps around the native browser events. These events work similarly to regular DOM events, but with better cross-browser support.

Example: onClick, onChange, onSubmit (camelCase syntax)

**2. Event Handlers**

Event handlers are functions that get triggered when a specific event occurs (e.g., a button click or form submission). In React, event handlers are assigned using JSX.

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

**3. Synthetic Event**

A **SyntheticEvent** is a cross-browser wrapper around the browser’s native event. It combines the behavior of different browsers into one consistent API in React.

Example:

function handleClick(e) {

console.log(e); // e is a SyntheticEvent

}

**4. React Event Naming Convention**

React event names follow **camelCase** rather than lowercase.

| **HTML** | **React** |
| --- | --- |
| onclick | onClick |
| onchange | onChange |
| onsubmit | onSubmit |

**1.Create a React App**

npx create-react-app eventexamplesapp

cd eventexamplesapp

npm start

App.js

import React from "react";

import "./App.css";

import CurrencyConverter from "./CurrencyConverter";

class App extends React.Component {

constructor(props) {

super(props);

this.state = {

count: 0

};

this.handleIncrement = this.handleIncrement.bind(this);

this.sayHello = this.sayHello.bind(this);

this.handleWelcome = this.handleWelcome.bind(this);

this.handleSyntheticEvent = this.handleSyntheticEvent.bind(this);

}

handleIncrement() {

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

this.sayHello();

}

sayHello() {

console.log("Hello! This is a static message.");

}

handleDecrement = () => {

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

};

handleWelcome(message) {

alert(message);

}

handleSyntheticEvent(e) {

alert("I was clicked");

console.log("Synthetic Event object:", e);

}

render() {

return (

<div className="App">

<h1>React Event Handling Example</h1>

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

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

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

<br /><br />

<button onClick={() => this.handleWelcome("Welcome!")}>

Say Welcome

</button>

<br /><br />

<button onClick={this.handleSyntheticEvent}>

Click for Synthetic Event

</button>

<br /><br />

<CurrencyConverter />

</div>

);

}

}

export default App;

CurrencyConverter.js

import React, { useState } from "react";

function CurrencyConverter() {

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

const [euro, setEuro] = useState("");

const handleSubmit = () => {

const euroValue = parseFloat(rupees) / 90;

setEuro(euroValue.toFixed(2));

};

return (

<div>

<h2>Currency Converter</h2>

<input

type="number"

placeholder="Enter INR"

value={rupees}

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

/>

<button onClick={handleSubmit}>Convert</button>

<h3>Euro: €{euro}</h3>

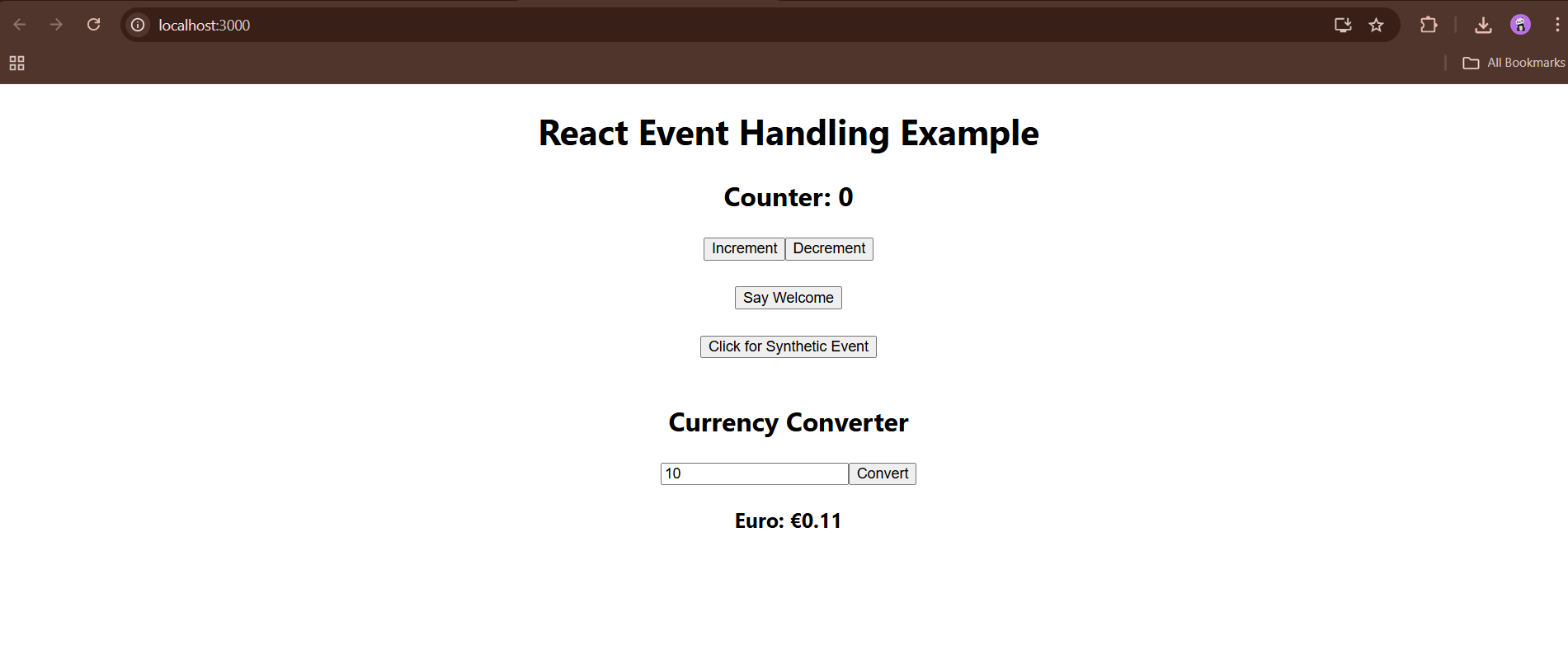
</div>

);

}

export default CurrencyConverter;

**Output**

* **Increment** and **Decrement** buttons update counter
* **Say Welcome** button shows an alert with "Welcome!"
* **Synthetic Event** button shows alert "I was clicked"
* **Currency Converter** converts INR to Euro on click
* 

**12. Ticket Booking App**

**1. Create the React App**

Open the terminal or command prompt and run:

npx create-react-app ticketbookingapp

cd ticketbookingapp

**2. Open the Project in VS Code**

code .

**3. Replace the content of src/App.js with the following:**

import React, { useState } from 'react';

import './App.css';

function FlightDetails() {

return (

<div>

<h2>Flight Details</h2>

<ul>

<li>Chennai → Delhi @ 9:00 AM</li>

<li>Mumbai → Bangalore @ 12:00 PM</li>

<li>Hyderabad → Kolkata @ 3:00 PM</li>

</ul>

</div>

);

}

function App() {

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

const handleLogin = () => setIsLoggedIn(true);

const handleLogout = () => setIsLoggedIn(false);

return (

<div className="App">

{isLoggedIn ? (

<>

<h1>Welcome back</h1>

<FlightDetails />

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

</>

) : (

<>

<h1>Please sign up.</h1>

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

</>

)}

</div>

);

}

export default App;

1. **Replace the content of src/App.css with the following:**

.App {

text-align: center;

margin-top: 100px;

font-family: Arial, sans-serif;

}

button {

padding: 8px 16px;

font-size: 16px;

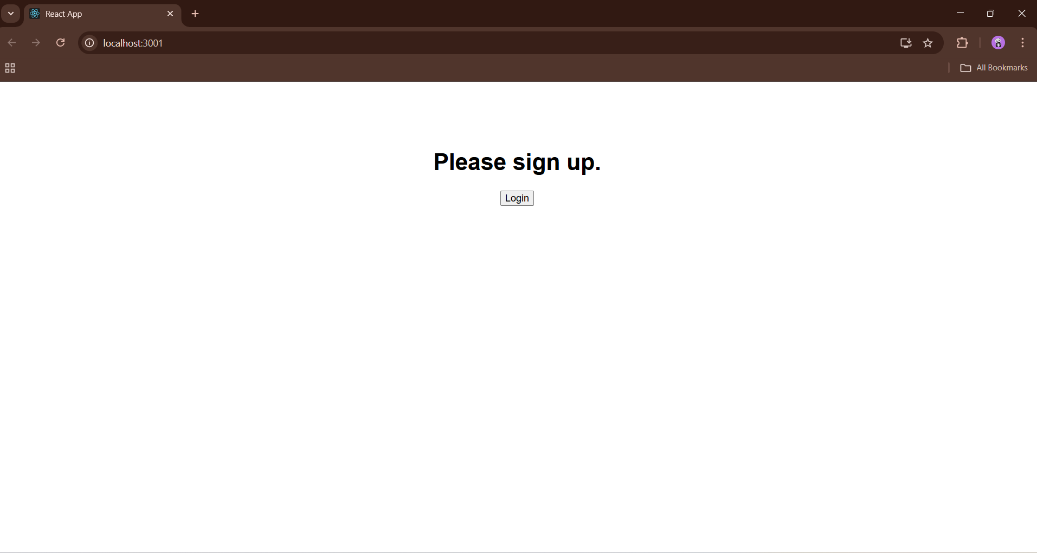
margin-top: 10px;

cursor: pointer;

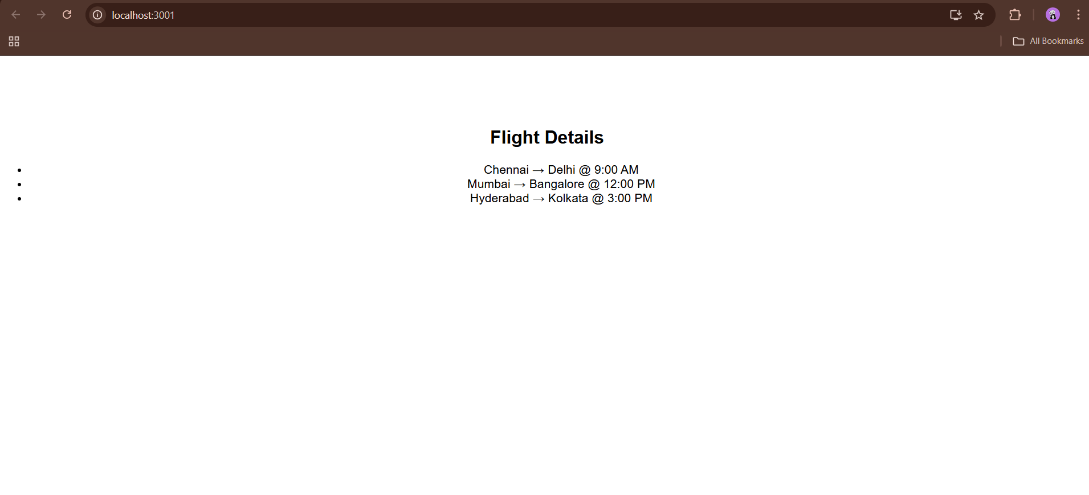
}

**Expected Output**

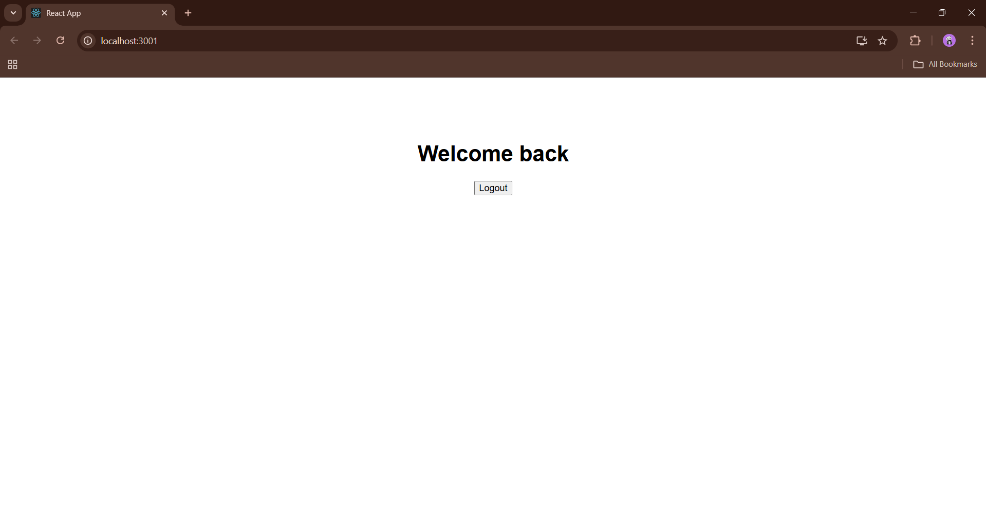
* **Initial View:**  
  Please sign up. and a **Login** button.



* **After Login:**
  + Welcome back message
  + List of **Flight Details**



* **After Logout:**
  + Back to Guest view with **Login** button again



**13.Blogger App**

**Project Description:**

Create a React Application named “bloggerapp” with 3 components:

1. BookDetails
2. BlogDetails
3. CourseDetails

Sample Code

**BookDetails.js**

import React from 'react';

function BookDetails() {

return (

<div>

<h2>Book Details</h2>

<ul>

<li>Title: React Explained</li>

<li>Author: Zac Gordon</li>

</ul>

</div>

);

}

export default BookDetails;

BlogDetails.js

import React from 'react';

function BlogDetails() {

return (

<div>

<h2>Blog Details</h2>

<p>This blog discusses the latest in frontend development using React.</p>

</div>

);

}

export default BlogDetails;

CourseDetails.js:

import React from 'react';

function CourseDetails() {

return (

<div>

<h2>Course Details</h2>

<p>Course Name: React for Beginners</p>

<p>Instructor: John Doe</p>

</div>

);

}

export default CourseDetails;

App.js

import React, { useState } from 'react';

import BookDetails from './components/BookDetails';

import BlogDetails from './components/BlogDetails';

import CourseDetails from './components/CourseDetails';

function App() {

const [selected, setSelected] = useState('');

return (

<div>

<h1>Blogger App</h1>

<button onClick={() => setSelected('book')}>Book</button>

<button onClick={() => setSelected('blog')}>Blog</button>

<button onClick={() => setSelected('course')}>Course</button>

<hr />

{selected === 'book' ? (

<BookDetails />

) : selected === 'blog' ? (

<BlogDetails />

) : selected === 'course' ? (

<CourseDetails />

) : (

<p>Please select a section to view.</p>

)}

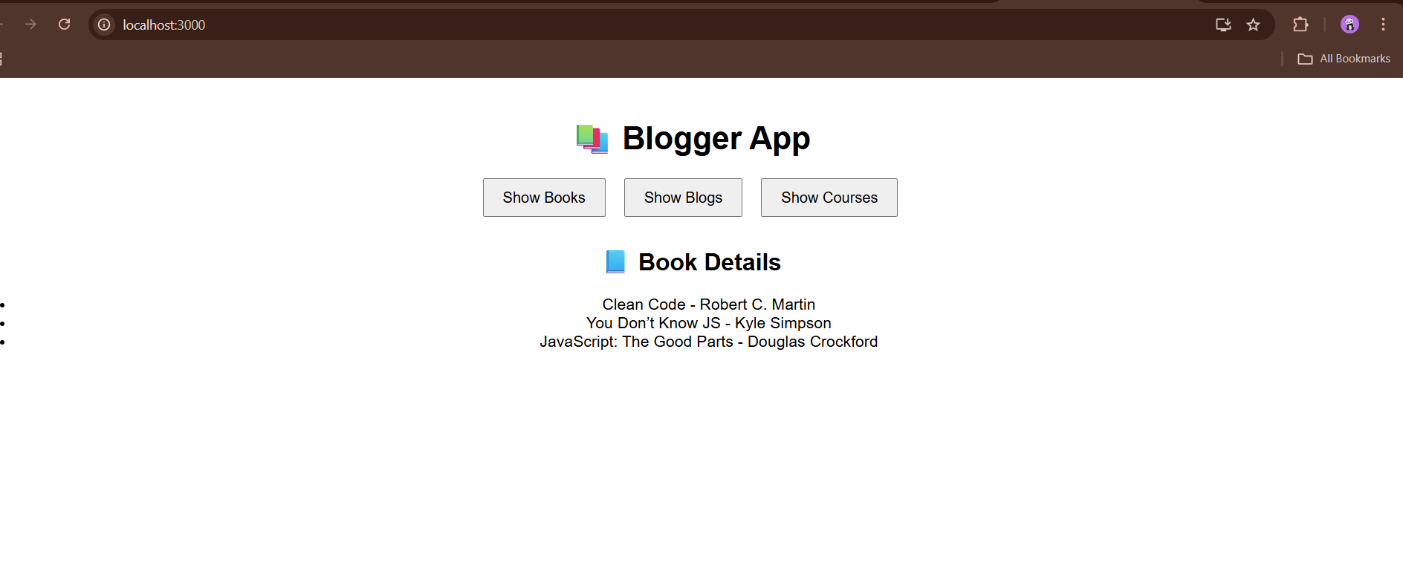
</div>

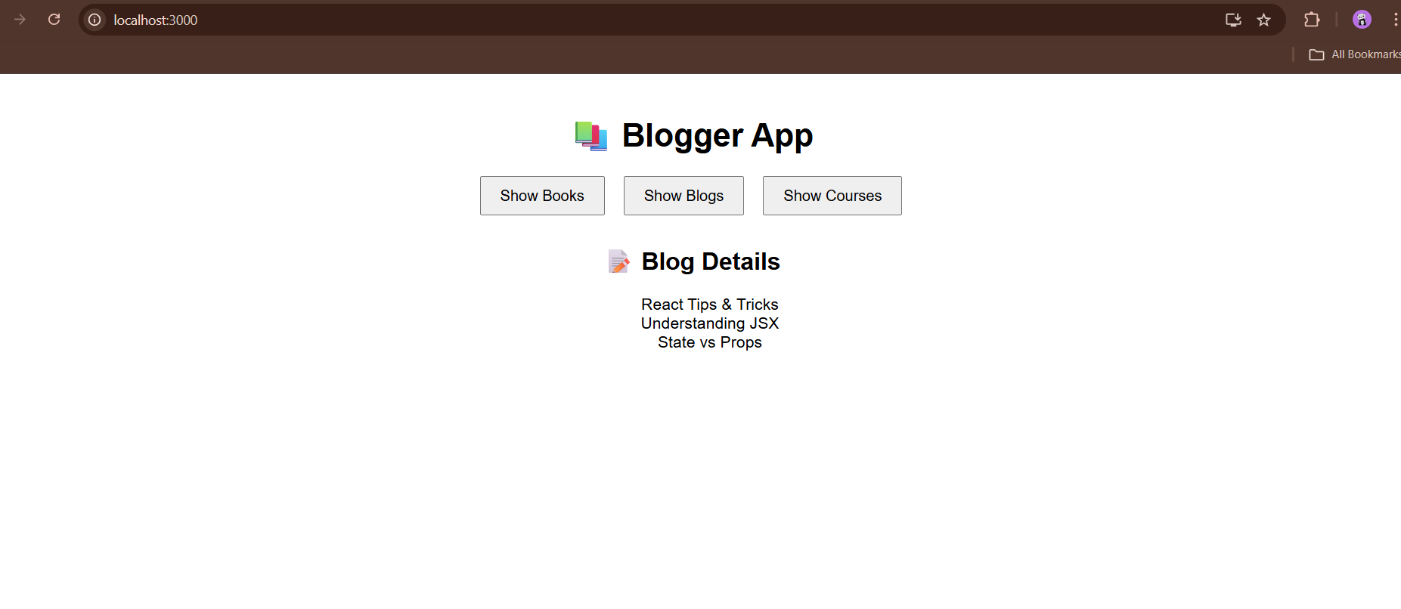
);

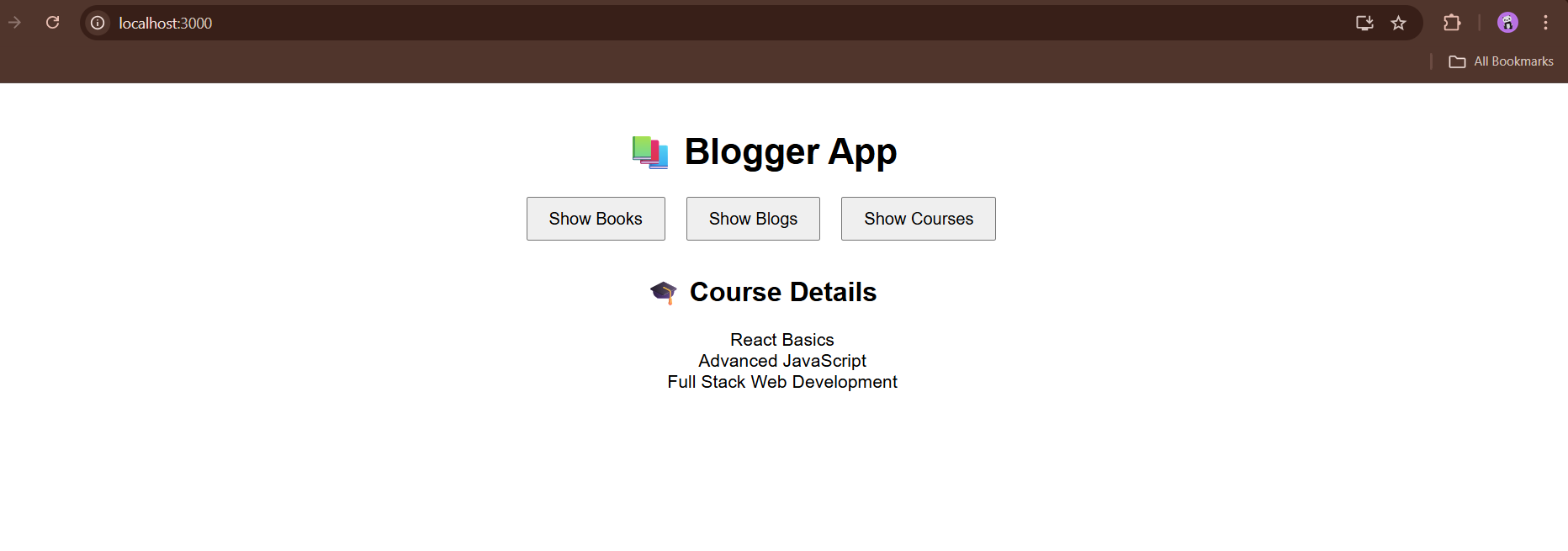
}

export default App;

**Output**

****

****

****