Week 7 React

# Hands-on 9: ES6 Features – cricketapp

Scenario:

Create a React application to showcase use of ES6 map(), arrow functions, destructuring, merging arrays and conditional rendering.

Steps:

* Use map() to list 11 players and their scores.
* Use arrow functions to filter scores below 70.
* Use destructuring to split players into Odd/Even teams.
* Merge two arrays (T20 and Ranji) and display them.
* Toggle display using a flag with if/else.

Filename: src/App.js

import React from 'react';

import { BrowserRouter as Router, Routes, Route, Link } from 'react-router-dom';

import Handson1 from './handsons/cricketapp/Handson1';

import Handson2 from './handsons/officespacerentalapp/Handson2';

import Handson3 from './handsons/eventexamplesapp/Handson3';

import Handson4 from './handsons/ticketbookingapp/Handson4';

import Handson5 from './handsons/bloggerapp/Handson5';

import './index.css';

export default function App() {

  return (

    <Router>

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

        <h1>Week-7 React Hands-ons</h1>

        <nav>

          <Link to="/handson1">Handson 1</Link> |{' '}

          <Link to="/handson2">Handson 2</Link> |{' '}

          <Link to="/handson3">Handson 3</Link> |{' '}

          <Link to="/handson4">Handson 4</Link> |{' '}

          <Link to="/handson5">Handson 5</Link>

        </nav>

        <Routes>

          <Route path="/handson1" element={<Handson1 />} />

          <Route path="/handson2" element={<Handson2 />} />

          <Route path="/handson3" element={<Handson3 />} />

          <Route path="/handson4" element={<Handson4 />} />

          <Route path="/handson5" element={<Handson5 />} />

        </Routes>

      </div>

    </Router>

  );

}

Filename: src/index.css

body { font-family: sans-serif; }

Filename: src/index.js

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

import './index.css';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<App />);

Filename: public/index.html

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="utf-8" />

    <link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />

    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <meta name="description" content="Week 7 React Hands-on Apps" />

    <title>Week 7 React Apps</title>

  </head>

  <body>

    <noscript>You need to enable JavaScript to run this app.</noscript>

    <div id="root"></div>

  </body>

</html>

Filename: src/handsons/cricketapp/handson1.js

Code:

import React, { useState } from "react";

import IndianPlayers from "./IndianPlayers";

import ListofPlayers from "./ListofPlayers";

function Handson1() {

  const [flag, setFlag] = useState(false);

  const toggleView = () => setFlag(prev => !prev);

  return (   <div>     <br>     </br>

      <button onClick={toggleView}>

        {flag ?"Show Player Scores": "Show Indian Players (Odd/Even)" }

      </button>

      <hr />

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

    </div> );}

export default Handson1;

Filename: src/handsons/cricketapp/**IndianPlayers.js**

import React from "react";

const T20 = ["Virat", "Rohit", "Pant", "Bumrah"];

const Ranji = ["Gill", "Jadeja", "Rahul", "Ashwin"];

const all = [...T20, ...Ranji];

const players = ["Virat", "Rohit", "Pant", "Jadeja", "Bumrah", "Shami"];

const odd = players.filter((\_, i) => i % 2 === 0);

const even = players.filter((\_, i) => i % 2 === 1);

function IndianPlayers() {

  return (

    <div>

      <h2>Odd Players</h2>

      <ul>{odd.map((p, i) => (

        <li key={i}>{`${getOrdinal(i \* 2 + 1)} : ${p}${i \* 2 + 1}`}</li>

      ))}</ul>

      <h2>Even Players</h2>

      <ul>{even.map((p, i) => (

        <li key={i}>{`${getOrdinal(i \* 2 + 2)} : ${p}${i \* 2 + 2}`}</li>

      ))}</ul>

    </div>

  );

}

function getOrdinal(n) {

  const ord = ['First', 'Second', 'Third', 'Fourth', 'Fifth', 'Sixth'];

  return ord[n - 1] || `${n}th`;

}

export default IndianPlayers;

Filename:src/handsons/ cricketapp/Listof**Players.js**

import React from "react";

const players = [

  { name: "Virat", score: 95 }, { name: "Rohit", score: 88 }, { name: "Rahul", score: 45 },

  { name: "Pant", score: 71 },   { name: "Jadeja", score: 30 },   { name: "Bumrah", score: 60 },

  { name: "Shami", score: 50 }, { name: "Ashwin", score: 79 }, { name: "Dhawan", score: 63 },

  { name: "Gill", score: 91 }, { name: "Kohli", score: 80 } ];

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

function ListofPlayers() {

  return ( <div>

      <h2>All Players and Scores</h2>

      <ul>

        {players.map((p, i) => (

          <li key={i}>{p.name}: {p.score}</li>    ))}

      </ul>

      <h3>Players with score below 70</h3> <ul>

        {lowScorers.map((p, i) => (

          <li key={i}>{p.name}: {p.score}</li>

        ))}

      </ul>

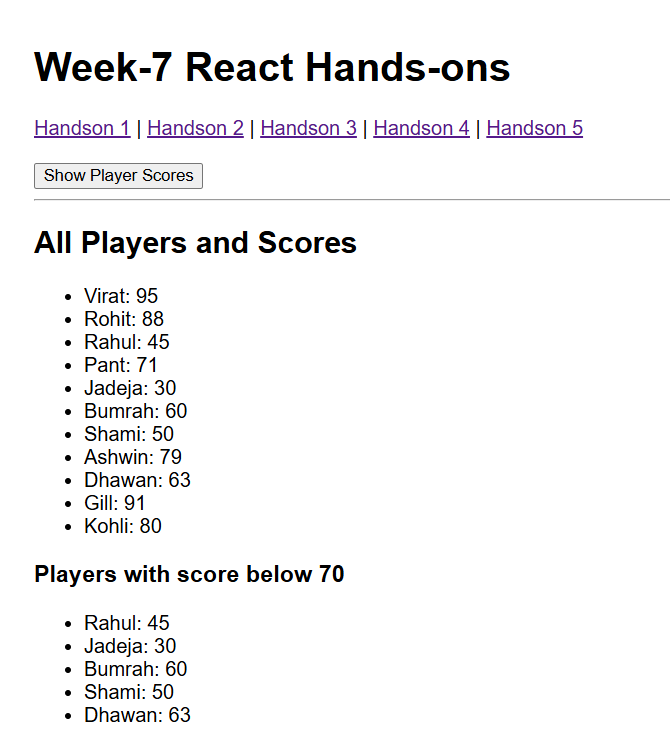
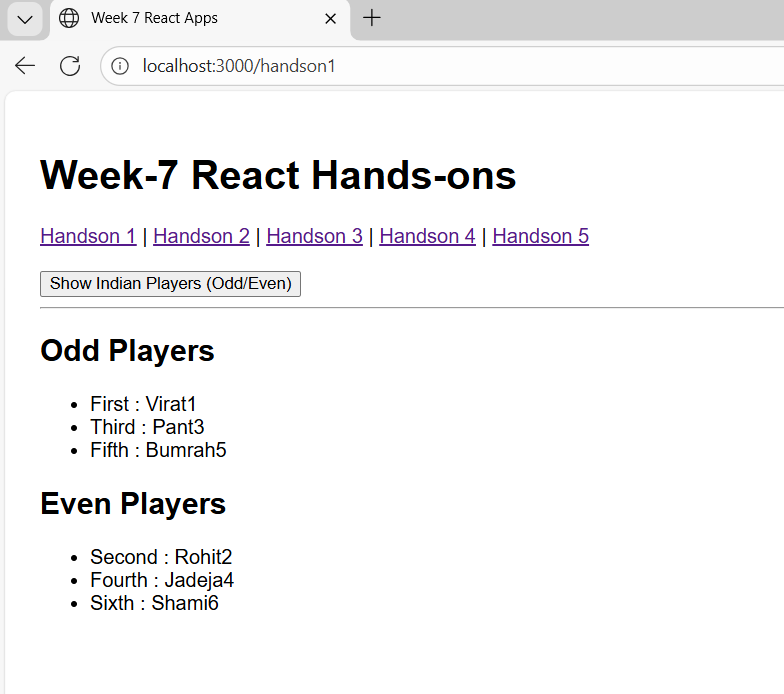
    </div>

  );

}

export default ListofPlayers;

**Output:**

# Hands-on 10: JSX and Inline CSS – officespacerentalapp

Scenario:

Build a React app that uses JSX to render office space rental listings with styled rent values.

Steps:

* Use JSX to display heading and image.
* Create office object with Name, Rent, Address.
* Loop over list of offices and display.
* Apply inline CSS to Rent: Red if < 60000, Green if > 60000.
* **Filename: officespacerentalapp/Handson2 .js**

Code:

import React from "react";

const offices = [

  { name: "Alpha Tower", rent: 55000, address: "Delhi", image: "/images/office.jpg" },

  { name: "Beta Building", rent: 72000, address: "Mumbai", image: "/images/office1.jpeg" },

  { name: "Gamma Plaza", rent: 61000, address: "Chennai", image: "/images/office2.jpeg" }

];

function Handson2() {

  return (

    <div>

      <h2>Office Space, At Affordable Range</h2>

      <div style={{ display: "flex", gap: "20px", flexWrap: "wrap" }}>

        {offices.map((o, i) => (

          <div key={i} style={{

            border: "1px solid #ccc",

            padding: "10px",

            width: "300px",

            color: o.rent > 60000 ? "green" : "red"

          }}>

            <img

              src={o.image}

              alt={o.name}

              style={{ width: "100%", height: "200px", objectFit: "cover" }}   />

            <h3>Name: {o.name}</h3>

            <h3>Rent: ₹{o.rent}</h3>

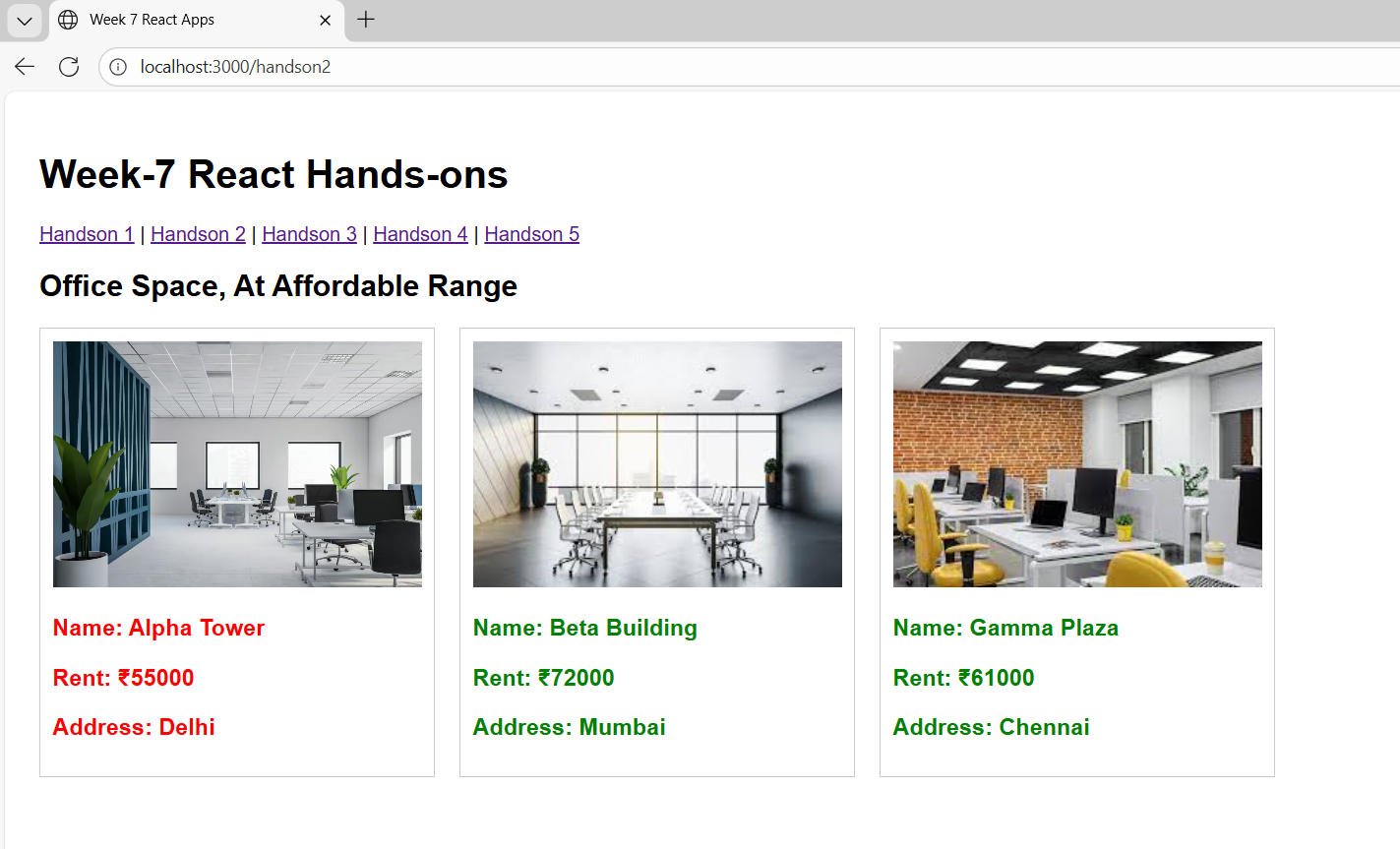
            <h3>Address: {o.address}</h3>

          </div> ))} </div> </div> );

}

export default Handson2;

# Output:



# Hands-on 11: React Events – eventexamplesapp

Scenario:

Create an application to demonstrate event handling with buttons and synthetic events.

Steps:

* Create Increment/Decrement buttons with multiple functions.
* Create button to say Welcome using argument.
* Synthetic event to log a message.
* Create CurrencyConvertor to convert INR to Euro.

Filename: eventexamplesapp/App.js

Code:

import React, { useState } from "react";

function CurrencyConvertor() {

  const [amount, setAmount] = useState("");

  const [currency, setCurrency] = useState("Euro");

  const handleSubmit = (e) => {

    e.preventDefault();

    const convertedAmount = parseFloat(amount) \* 80; // Example: 1 Euro = ₹80

    alert(`Converting to ${currency} Amount is ${convertedAmount}`);

  };

  return (

    <div>

      <h1 style={{ color: "green" }}>Currency Convertor!!!</h1>

      <form onSubmit={handleSubmit}>

        <label>

          <strong>Amount:</strong>&nbsp;

          <input

            type="number"

            value={amount}

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

          />

        </label>

        <br /><br />

        <label>

          <strong>Currency:</strong>&nbsp;

          <select

            value={currency}

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

          >

            <option value="Euro">Euro</option>

            {/\* You can add more currencies here \*/}

          </select>

        </label>

        <br /><br />

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

      </form>

    </div>

  );

}

function Handson3() {

  const [count, setCount] = useState(0);

  const sayHello = () => alert("Hello!");

  const increment = () => {

    setCount(count + 1);

    sayHello();

  };

  const decrement = () => setCount(count - 1);

  const sayWelcome = (msg) => alert(msg);

  const handleClick = () => alert("I was clicked");

  return (

    <div>

      <h2>Counter: {count}</h2>

      <button onClick={increment}>Increment</button><br></br>

      <button onClick={decrement}>Decrement</button><br></br>

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

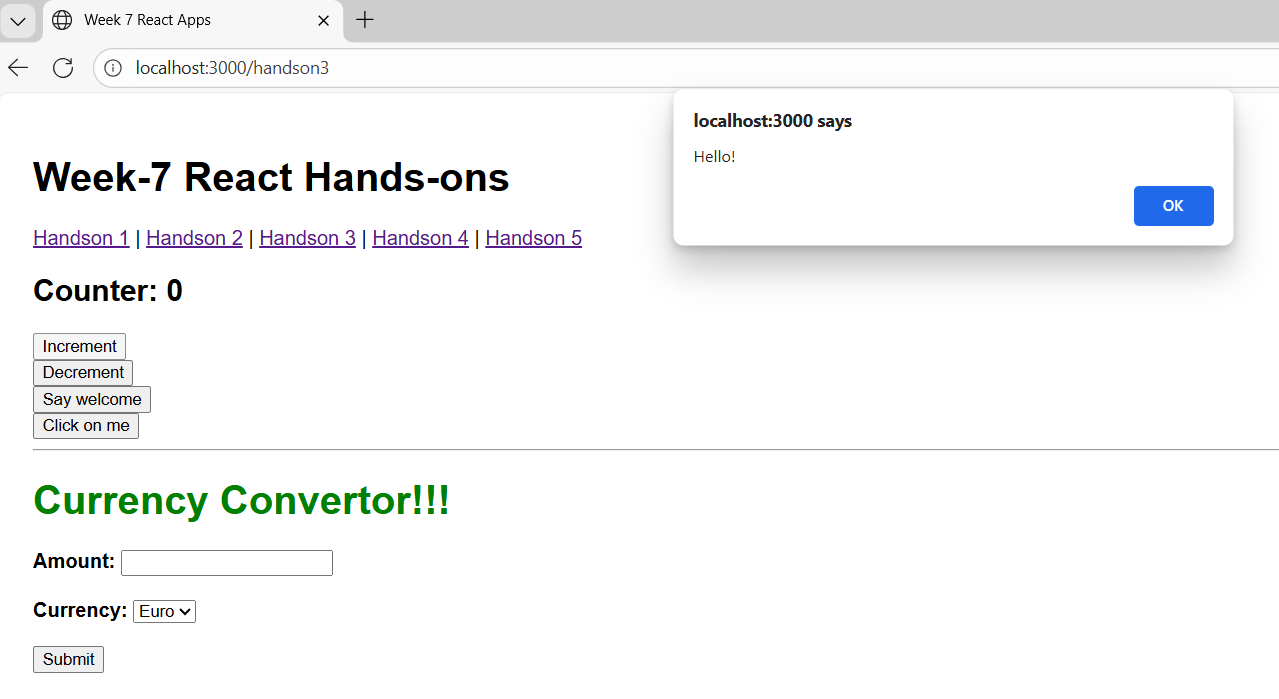
      <button onClick={handleClick}>Click on me</button>

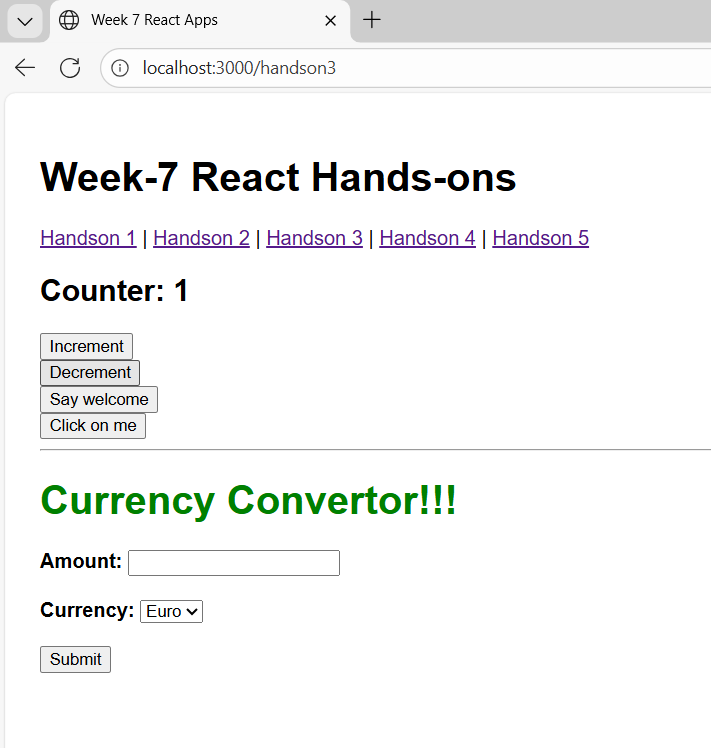
  <hr />

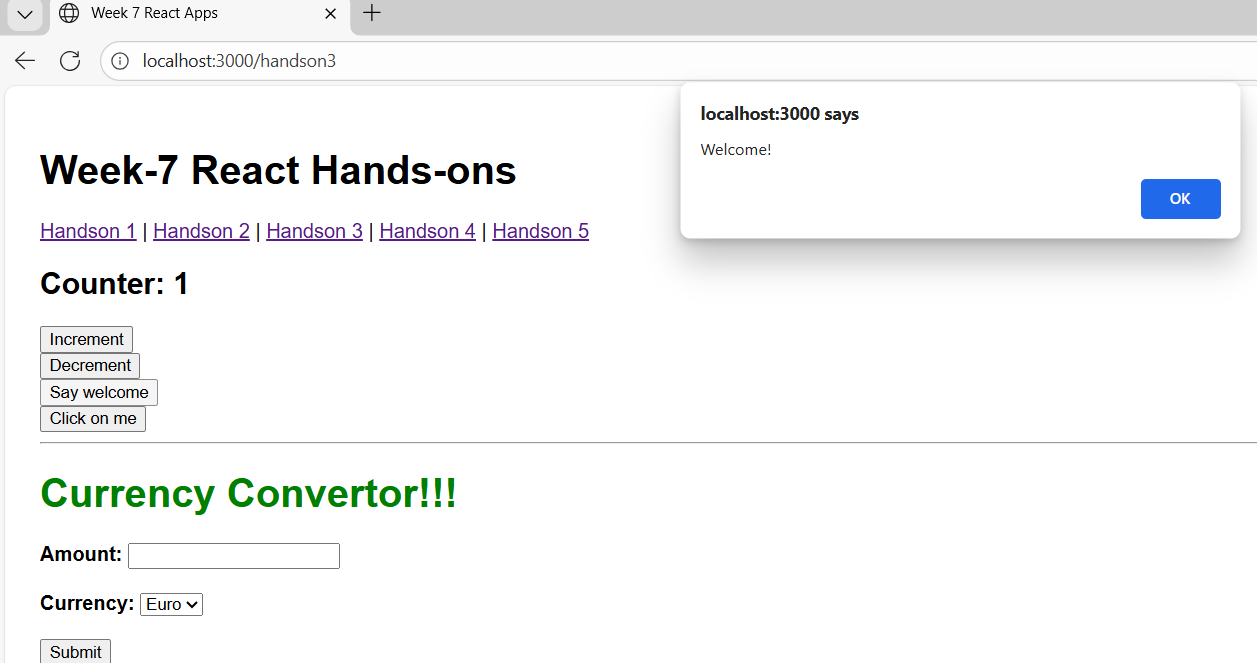
      <CurrencyConvertor />    </div>  );}

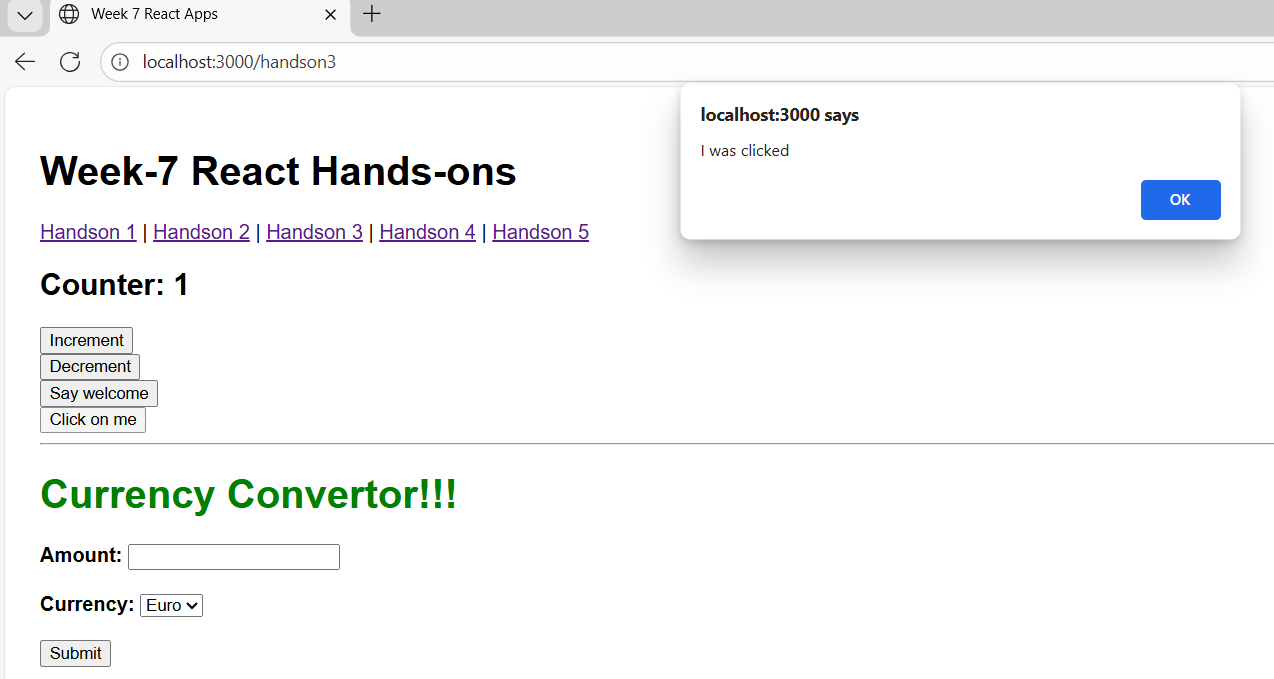
export default Handson3;

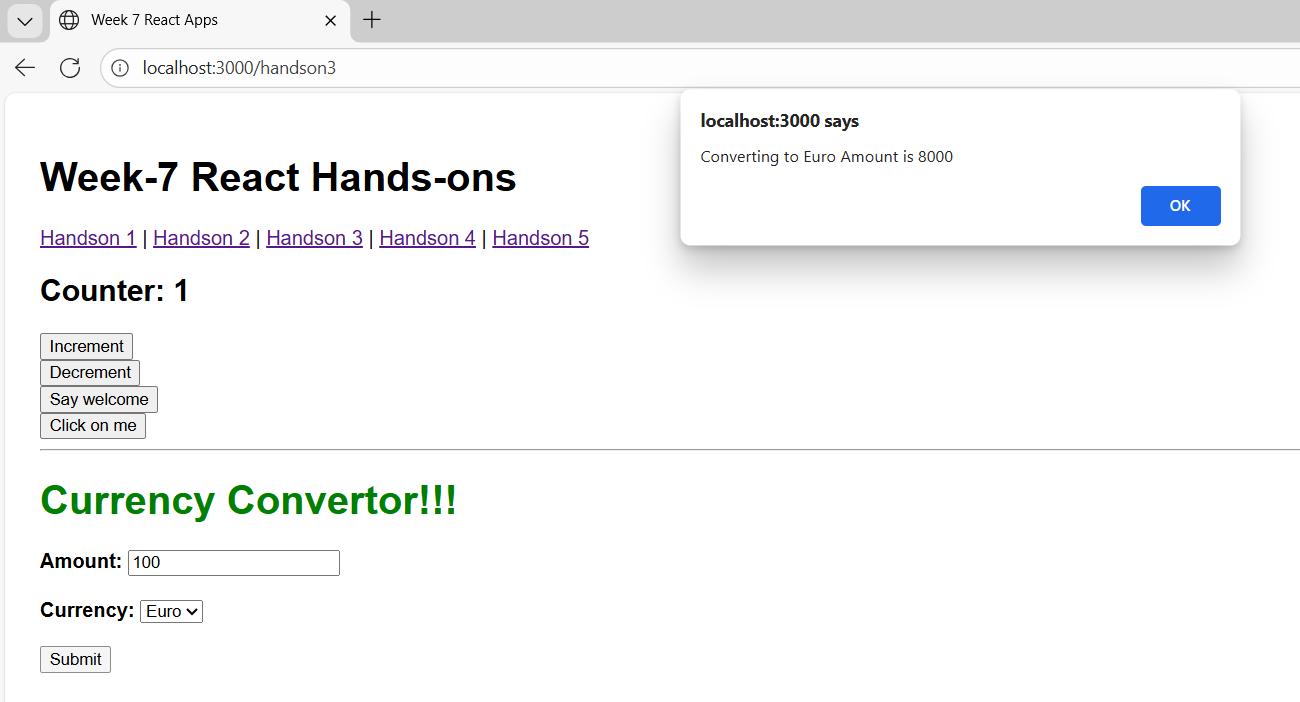
# Output:











# Hands-on 12: Conditional Rendering – ticketbookingapp

Scenario:

Create a flight booking page with conditional rendering based on login status.

Steps:

* Guest can view flights.
* Logged-in users can book tickets.
* Use Login and Logout buttons to toggle views.

# Filename:ticketbookingapp/Handson4.js

Code:

import React, { useState } from "react";

function Greeting({ isLoggedIn }) {

  if (isLoggedIn) {

    return <h2>Welcome back</h2>;

  }

  return <h2>Please sign up.</h2>;

}

function Handson4() {

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

  const handleClick = () => {

    setIsLoggedIn(!isLoggedIn);

  };

  return (

    <div style={{ padding: "30px" }}>

      <Greeting isLoggedIn={isLoggedIn} />

      <button onClick={handleClick}>

        {isLoggedIn ? "Logout" : "Login"}

      </button>

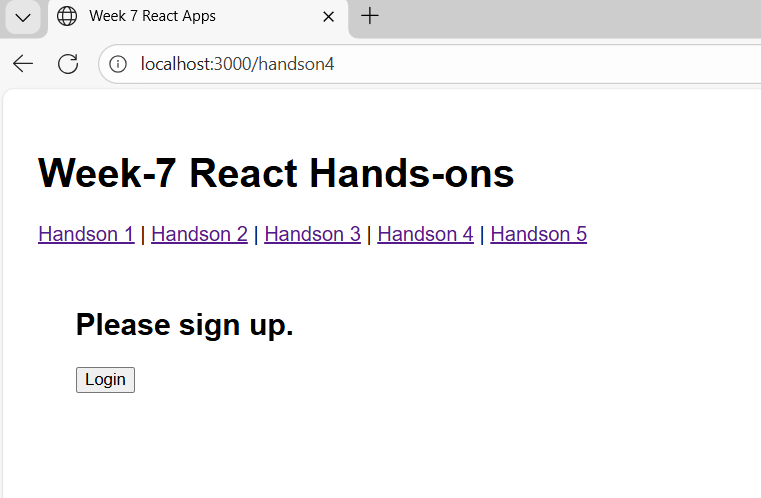
    </div>

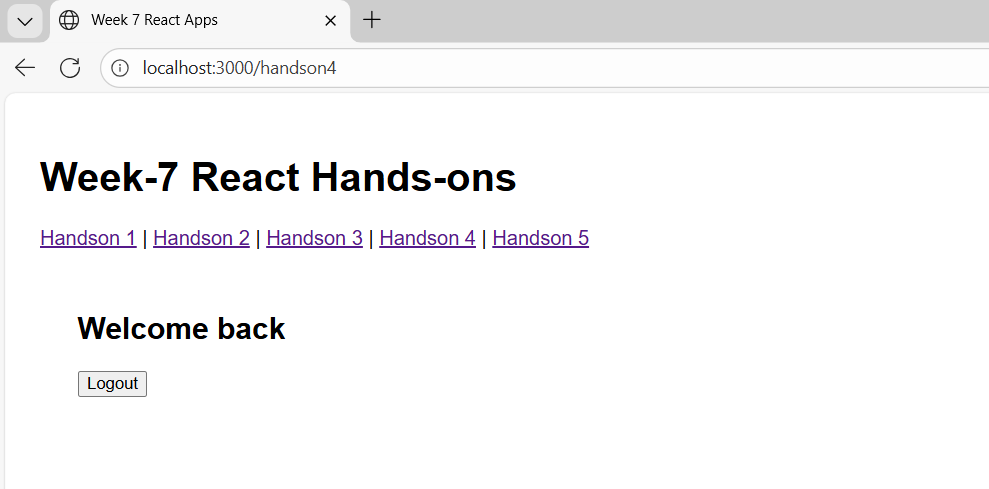
  );

}

export default Handson4;

**Output:**





# Hands-on 13: Multiple Components & Conditional Rendering – bloggerapp

# Scenario:

Develop a React app with Book, Blog and Course details using various conditional rendering techniques.

Steps:

* Create 3 components: BookDetails, BlogDetails, CourseDetails.
* Use multiple conditional rendering styles to toggle views.

Filename: bloggerapp/Handson5.js

Code:

import React from "react";

import "./handson5.css";

function Handson5() {

  return (

    <div className="container">

      <div className="column">

        <h2>Course Details</h2>

        <h3>Angular</h3>

        <p>4/5/2021</p>

        <h3>React</h3>

        <p>6/3/20201</p>

      </div>

      <div className="column">

        <h2>Book Details</h2>

        <p><strong>Master React</strong></p>

        <p>670</p>

        <p><strong>Deep Dive into Angular 11</strong></p>

        <p>800</p>

        <p><strong>Mongo Essentials</strong></p>

        <p>450</p>

      </div>

      <div className="column">

        <h2>Blog Details</h2>

        <h3>React Learning</h3>

        <p><strong>Stephen Biz</strong></p>

        <p>Welcome to learning React!</p>

        <h3>Installation</h3>

        <p><strong>Schwezdneier</strong></p>

        <p>You can install React from npm.</p>

      </div>

    </div>

  );

}

export default Handson5;

Filename: bloggerapp/Handson5.css

.container {

  display: flex;

  justify-content: space-around;

  padding: 20px;

  gap: 30px;

}

.column {

  padding-left: 20px;

  width: 30%;

  position: relative;

}

.column:not(:last-child)::after {

  content: "";

  position: absolute;

  top: 0;

  right: -15px;

  width: 4px;

  height: 100%;

  background-color: green;

}

**Output:**

