**Digital Nurture 4.0 – Week 7**

**Mandatory hands-on**

**Filename : 9. ReactJS-HOL**

**Objective :**

The objective of this hands-on lab is to strengthen your understanding of ES6 JavaScript features by implementing them in a React application named “cricketapp.” You will work with modern ES6 constructs such as map(), arrow functions, destructuring, let, const, class, inheritance, and collections like Set and Map. The application includes components that handle an array of cricket players, apply score filtering, team separation, array merging, and conditional rendering using a flag variable, thereby improving your React and ES6 skills in a practical, component-driven environment.

**Steps:**

1. **Set Up the Environment**
   1. npx create-react-app cricketapp
   2. cd cricketapp
   3. npm start
2. **Create Components**

Inside src, create two component files: ListofPlayers.js and IndianPlayers.js.

1. **Component: ListofPlayers**

**App.js**

import React from 'react';

import ListofPlayers from './ListofPlayers';

import Scorebelow70 from './Scorebelow70';

import { OddPlayers } from './OddPlayers';

import { EvenPlayers } from './EvenPlayers';

import { ListofIndianPlayers } from './ListofIndianPlayers';

import { IndianTeam, IndianPlayers } from './PlayerData';

const players = [

{ name: 'Jack', score: 50 },

{ name: 'Michael', score: 70 },

{ name: 'John', score: 40 },

{ name: 'Ann', score: 61 },

{ name: 'Elisabeth', score: 61 },

{ name: 'Sachin', score: 95 },

{ name: 'Dhoni', score: 100 },

{ name: 'Virat', score: 84 },

{ name: 'Jadeja', score: 64 },

{ name: 'Raina', score: 75 },

{ name: 'Rohit', score: 80 }

];

var flag = true;

function App() {

if (flag === true) {

return (

<div>

<h1>List of Players</h1>

<ListofPlayers players={players} />

<hr />

<h1>List of Players having Scores Less than 70</h1>

<Scorebelow70 players={players} />

</div>

);

} else {

return (

<div>

<div>

<h1>Indian Team</h1>

<h1>Odd Players</h1>

{OddPlayers(IndianTeam)}

<hr />

<h1>Even Players</h1>

{EvenPlayers(IndianTeam)}

</div>

<hr />

<div>

<h1>List of Indian Players Merged:</h1>

<ListofIndianPlayers IndianPlayers={IndianPlayers} />

</div>

</div>

);

}

}

export default App;

**ListofPlayers.js**

import React from 'react';

function ListofPlayers({ players }) {

return (

<ul>

{players.map((item, index) => {

return (

<li key={index}>

Mr. {item.name} <span>{item.score}</span>

</li>

);

})}

</ul>

);

}

export default ListofPlayers;

**Scorebelow70.js**

import React from 'react';

function Scorebelow70({ players }) {

const players70 = [];

players.map((item) => {

if (item.score <= 70) {

players70.push(item);

}

});

return (

<ul>

{players70.map((item, index) => (

<li key={index}>

Mr. {item.name} <span>{item.score}</span>

</li>

))}

</ul>

);

}

export default Scorebelow70;

**OddPlayers.js**

import React from 'react';

export function OddPlayers([first, , third, , fifth]) {

return (

<ul>

<li>First : {first}</li>

<li>Third : {third}</li>

<li>Fifth : {fifth}</li>

</ul>

);

}

**EvenPlayers.js**

import React from 'react';

export function EvenPlayers([, second, , fourth, , sixth]) {

return (

<ul>

<li>Second : {second}</li>

<li>Fourth : {fourth}</li>

<li>Sixth : {sixth}</li>

</ul>

);

}

**ListofIndianPlayers.js**

import React from 'react';

export function ListofIndianPlayers({ IndianPlayers }) {

return (

<ul>

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

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

))}

</ul>

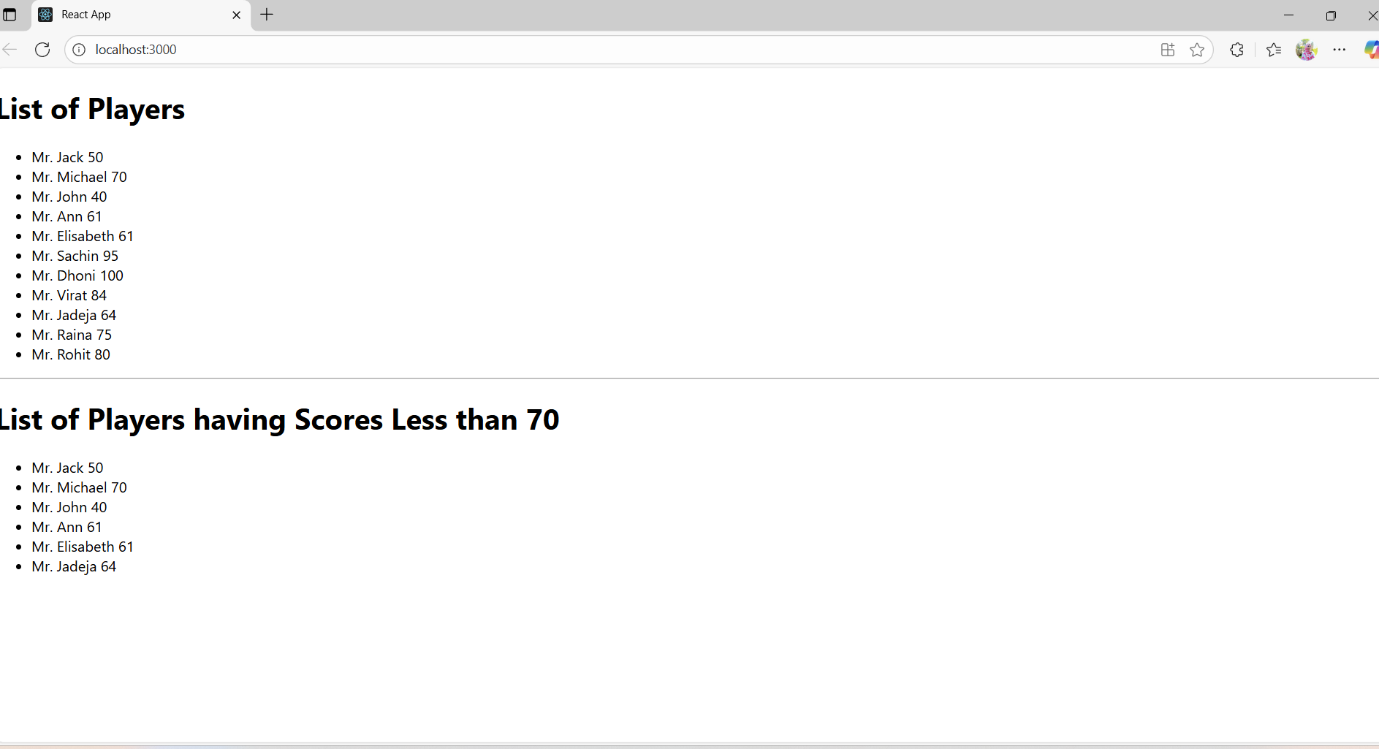
);

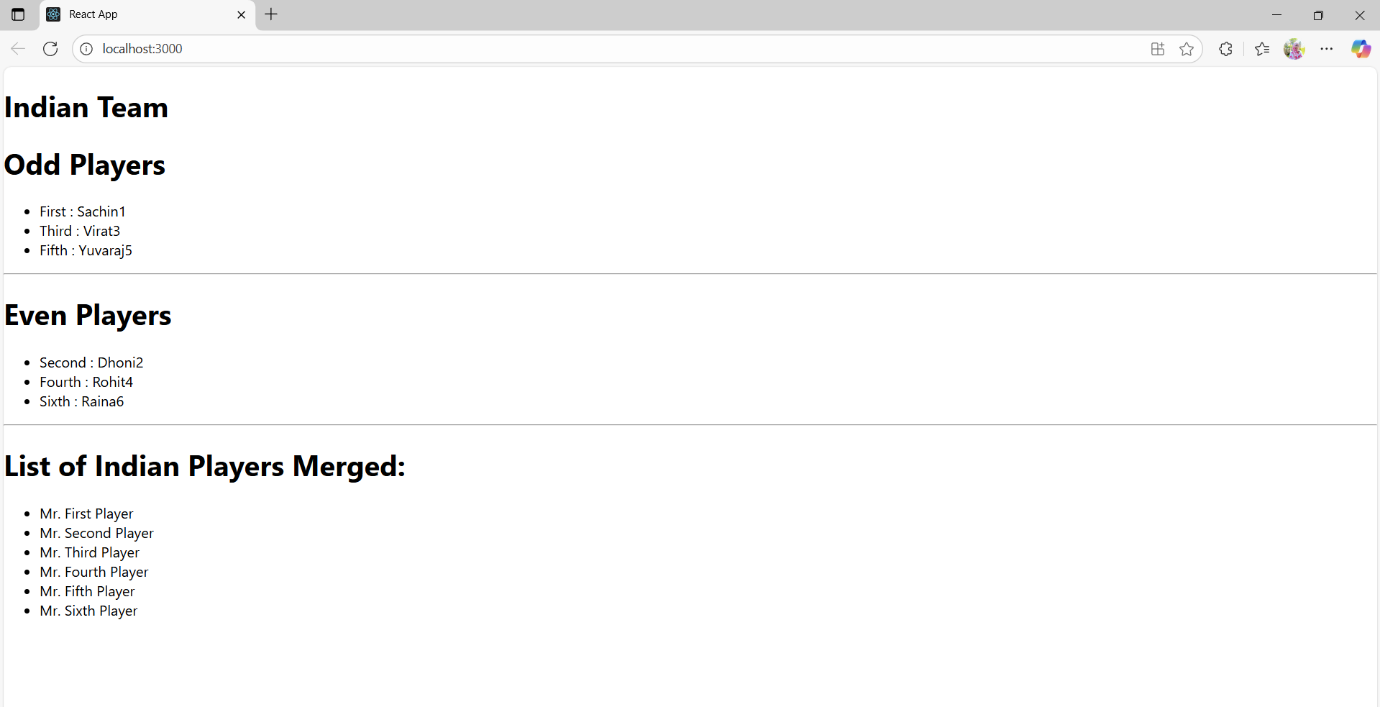
}

1. **Run the App**

npm start

**Output:**

****

****

**Filename : 10. ReactJS-HOL**

**Objective:**

The objective of this hands-on lab is to build a React application named "officespacerentalapp" that demonstrates the use of JSX to create and render elements dynamically, apply attributes like image sources, define and utilize JavaScript objects to display office details such as Name, Rent, and Address, and implement conditional rendering for styling—specifically, displaying the Rent amount in red if it is below ₹60,000 and in green if it is above ₹60,000—while also incorporating inline and external CSS styling techniques within a structured React environment.

**Steps:**

**Step 1: Set up your React App**

npx create-react-app officespacerentalapp

**Step 2 :Update App.js**

import React from 'react';

import './App.css';

import officeImg from './office.jpg';

function App() {

const element = "Office Space";

const ItemName = {

Name: "DBS",

Rent: 50000,

Address: "Chennai"

};

let colors = [];

if (ItemName.Rent <= 60000) {

colors.push("textRed");

} else {

colors.push("textGreen");

}

const jsxatt = <img src={officeImg} width="25%" height="25%" alt="Office Space" />;

return (

<div>

<h1><b>{element} , at Affordable Range</b></h1>

{jsxatt}

<h1>Name: {ItemName.Name}</h1>

<h3 className={colors.join(' ')}>Rent: Rs. {ItemName.Rent}</h3>

<h3>Address: {ItemName.Address}</h3>

</div>

);

}

export default App;

**Step 3: Add CSS Styling**

.textRed {

color: red;

font-weight: bold;

}

.textGreen {

color: green;

font-weight: bold;

}

body {

text-align: center;

margin-top: 40px;

}

**Step 4: Add Image**

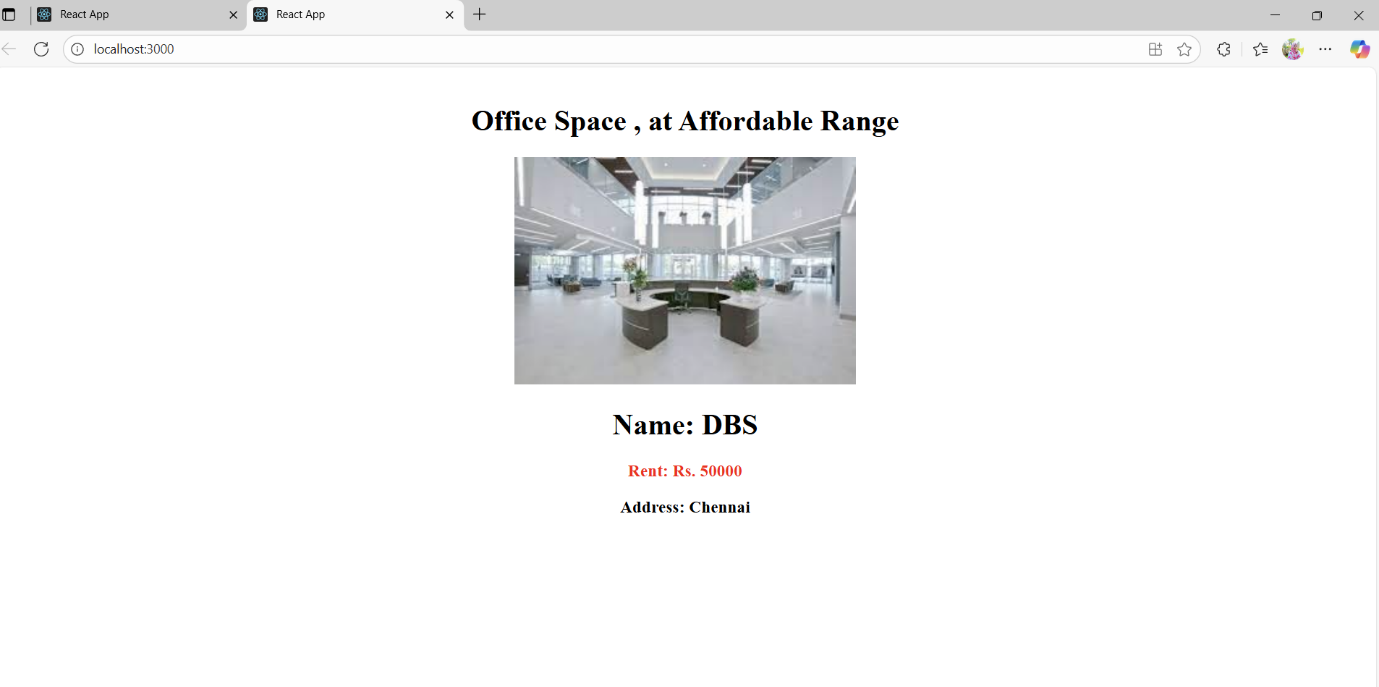
Save any office image as office.jpg in the src folder.

**Step 5: Run The App**

npm start

<http://localhost:3000>

**Output:**

****

**Filename : 11. ReactJS-HOL**

**Objective:**

The objective of this hands-on lab is to implement event handling in a React application by creating interactive form elements and buttons that demonstrate the use of event handlers, the this keyword, synthetic events, and conditional logic; specifically, users will interact with buttons to increment and decrement a counter, trigger custom alert messages such as "Hello!", "welcome", and "I was clicked", and perform currency conversion by handling form submission, thereby gaining practical experience in managing state and events in React components.

**Steps:**

**Step 1: Create React App**

npx create-react-app eventexamplesapp

**Step 2: App.js**

import React, { Component } from 'react';

import './App.css';

class App extends Component {

constructor(props) {

super(props);

this.state = {

count: 5,

amount: '',

currency: ''

};

}

incrementValue = () => {

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

this.sayHello();

};

sayHello = () => {

alert("Hello! Member1");

};

decrementValue = () => {

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

};

sayWelcome = (msg) => {

alert(msg);

};

handleClick = (e) => {

alert("I was clicked");

};

handleAmountChange = (e) => {

this.setState({ amount: e.target.value });

};

handleCurrencyChange = (e) => {

this.setState({ currency: e.target.value });

};

handleSubmit = (e) => {

e.preventDefault();

const { amount, currency } = this.state;

if (currency.toLowerCase() === "euro") {

const result = amount \* 80;

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

} else {

alert("Currency not supported");

}

};

render() {

return (

<div>

<p>{this.state.count}</p>

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

<br />

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

<br />

<button onClick={() => this.sayWelcome("welcome")}>Say welcome</button>

<br />

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

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

<form onSubmit={this.handleSubmit}>

<div>

<label>Amount:</label>

<input

type="text"

value={this.state.amount}

onChange={this.handleAmountChange}

/>

</div>

<div>

<label>Currency:</label>

<textarea

value={this.state.currency}

onChange={this.handleCurrencyChange}

></textarea>

</div>

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

</form>

</div>

);

}

}

export default App;

**Step 3:** **App.css**

h1 {

font-size: 30px;

font-weight: bold;

}

button {

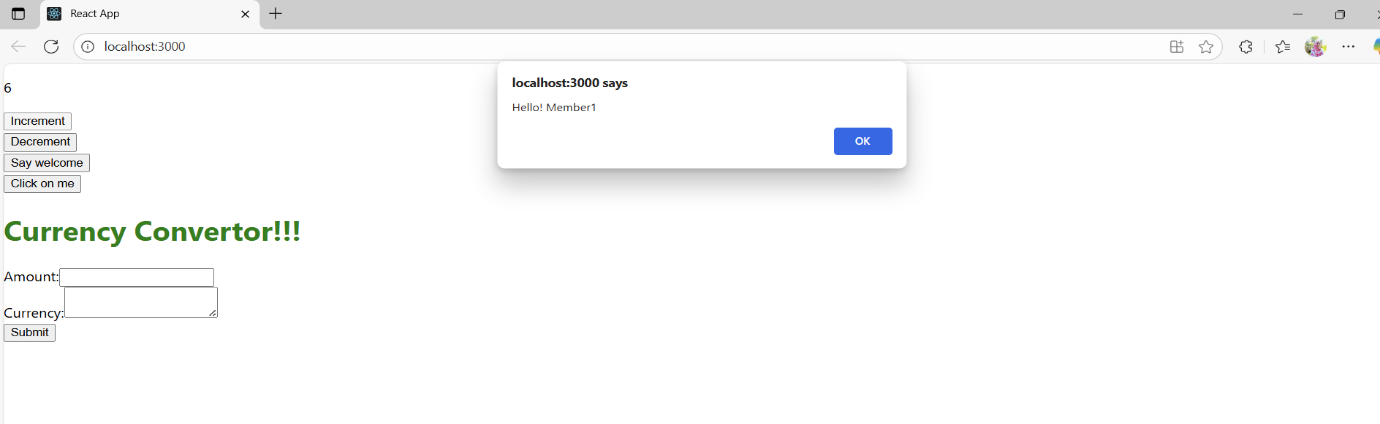
margin: 5px;

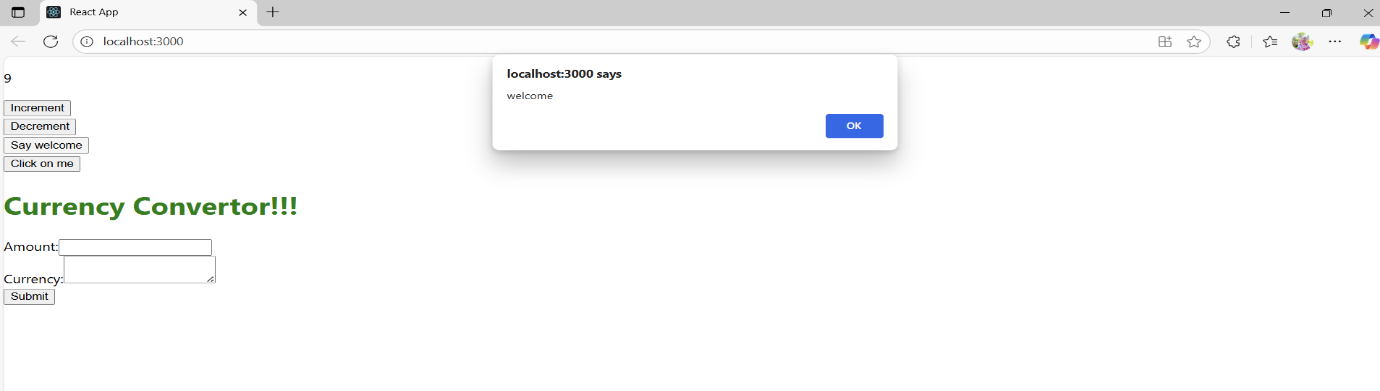
}

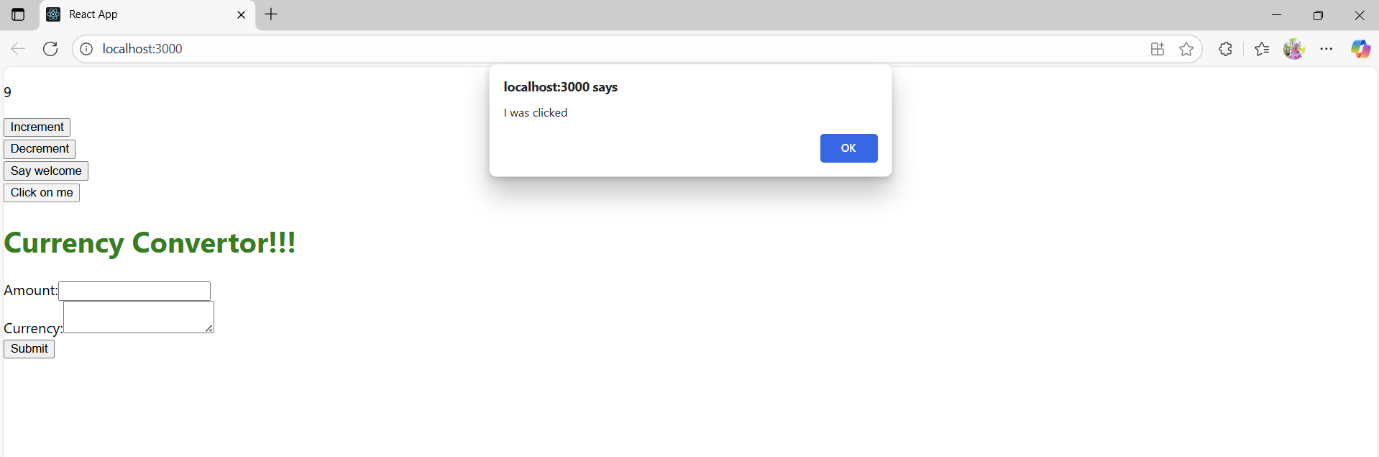
**Step 4: Run The App**

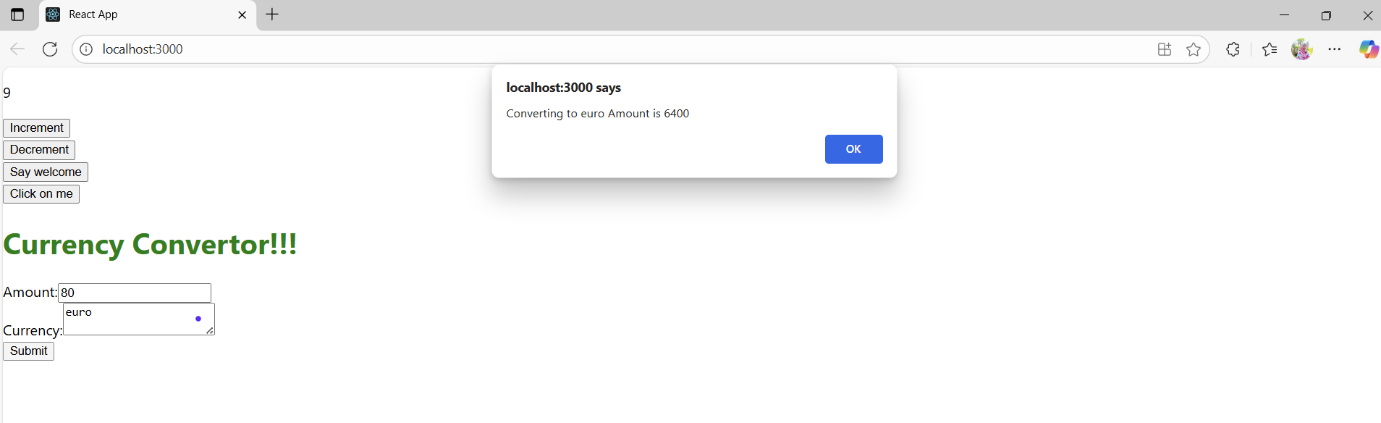
npm start

**Output:**

****

****

****

****

**Filename : 12. ReactJS-HOL**

**Objective:**

The objective of this project is to implement conditional rendering in a React application by creating a simple login system where the interface dynamically displays different content based on the user's login status—showing a "Please sign up" message with a Login button for guests, and a "Welcome back" message with a Logout button for logged-in users—along with proper layout styling to align the content with appropriate spacing from the left side of the screen.

**Steps:**

**Step 1: Create a new React App**

npx create-react-app login-app

**Step 2: App.js**

import React from 'react';

import './App.css';

function LoginButton(props) {

return (

<button onClick={props.onClick}>

Login

</button>

);

}

function LogoutButton(props) {

return (

<button onClick={props.onClick}>

Logout

</button>

);

}

function UserGreeting() {

return <h1>Welcome back</h1>;

}

function GuestGreeting() {

return <h1>Please sign up.</h1>;

}

function Greeting(props) {

const isLoggedIn = props.isLoggedIn;

if (isLoggedIn) {

return <UserGreeting />;

}

return <GuestGreeting />;

}

class App extends React.Component {

constructor(props) {

super(props);

this.handleLoginClick = this.handleLoginClick.bind(this);

this.handleLogoutClick = this.handleLogoutClick.bind(this);

this.state = { isLoggedIn: false };

}

handleLoginClick() {

this.setState({ isLoggedIn: true });

}

handleLogoutClick() {

this.setState({ isLoggedIn: false });

}

render() {

const isLoggedIn = this.state.isLoggedIn;

let button;

if (isLoggedIn) {

button = <LogoutButton onClick={this.handleLogoutClick} />;

} else {

button = <LoginButton onClick={this.handleLoginClick} />;

}

return (

<div>

<Greeting isLoggedIn={isLoggedIn} />

{button}

</div>

);

}

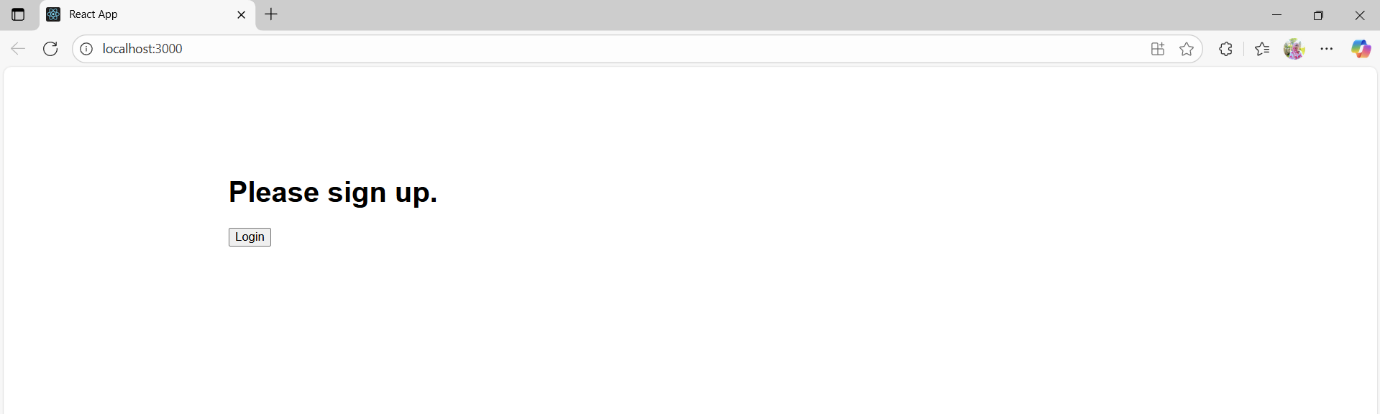
}

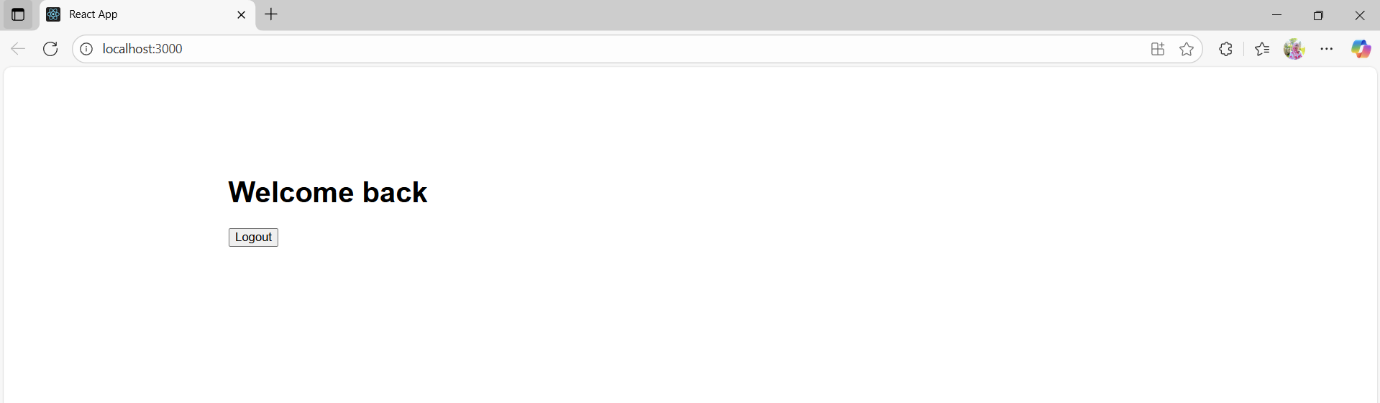
export default App;

**Step 3: Run your application**

npm start

**Output:**

****

****

**Filename : 13. ReactJS-HOL**

**Objective:**

The objective of this ReactJS project is to create a structured web application named “bloggerapp” that demonstrates how to render multiple components-Course Details, Book Details, and Blog Details-by effectively using React’s component-based architecture, props, and the map() function for dynamic rendering of lists, while also practicing styling and layout management using CSS for a clear and visually separated presentation.

**Steps:**

**Step 1:** **Create a new React app**

npx create-react-app bloggerapp

cd bloggerapp

**Step 2: Data.js**

export const books = [

{ id: 101, bname: 'Master React', price: 670 },

{ id: 102, bname: 'Deep Dive into Angular 11', price: 800 },

{ id: 103, bname: 'Mongo Essentials', price: 450 }

];

export const courses = [

{ cname: 'Angular', date: '4/5/2021' },

{ cname: 'React', date: '6/3/20201' }

];

export const blogs = [

{

title: 'React Learning',

author: 'Stephen Biz',

content: 'Welcome to learning React!'

},

{

title: 'Installation',

author: 'Schewzdenier',

content: 'You can install React from npm.'

}

];

**Step 3: BookDetails.js**

import React from 'react';

const BookDetails = (props) => {

return (

<ul>

{props.books.map((book) => (

<div key={book.id}>

<h3>{book.bname}</h3>

<h4>{book.price}</h4>

</div>

))}

</ul>

);

};

export default BookDetails;

**Step 4: CourseDetails.js**

import React from 'react';

const CourseDetails = (props) => {

return (

<div>

{props.courses.map((course, index) => (

<div key={index}>

<h3>{course.cname}</h3>

<h4>{course.date}</h4>

</div>

))}

</div>

);

};

export default CourseDetails;

**Step 5: BlogDetails.js**

import React from 'react';

const BlogDetails = (props) => {

return (

<div>

{props.blogs.map((blog, index) => (

<div key={index}>

<h3>{blog.title}</h3>

<b>{blog.author}</b>

<p>{blog.content}</p>

</div>

))}

</div>

);

};

export default BlogDetails;

**Step 6: App.css**

.App {

display: flex;

justify-content: center;

margin-top: 50px;

font-family: Arial;

}

.mystyle1, .st2, .v1 {

padding: 20px;

}

.mystyle1 {

border-right: 4px solid green;

}

.st2 {

border-right: 4px solid green;

}

h1 {

font-size: 25px;

font-weight: bold;

}

**Step 8: App.js**

import React from 'react';

import './App.css';

import BookDetails from './BookDetails';

import BlogDetails from './BlogDetails';

import CourseDetails from './CourseDetails';

import { books, courses, blogs } from './data';

function App() {

return (

<div className="App">

<div className="mystyle1">

<h1>Course Details</h1>

<CourseDetails courses={courses} />

</div>

<div className="st2">

<h1>Book Details</h1>

<BookDetails books={books} />

</div>

<div className="v1">

<h1>Blog Details</h1>

<BlogDetails blogs={blogs} />

</div>

</div>

);

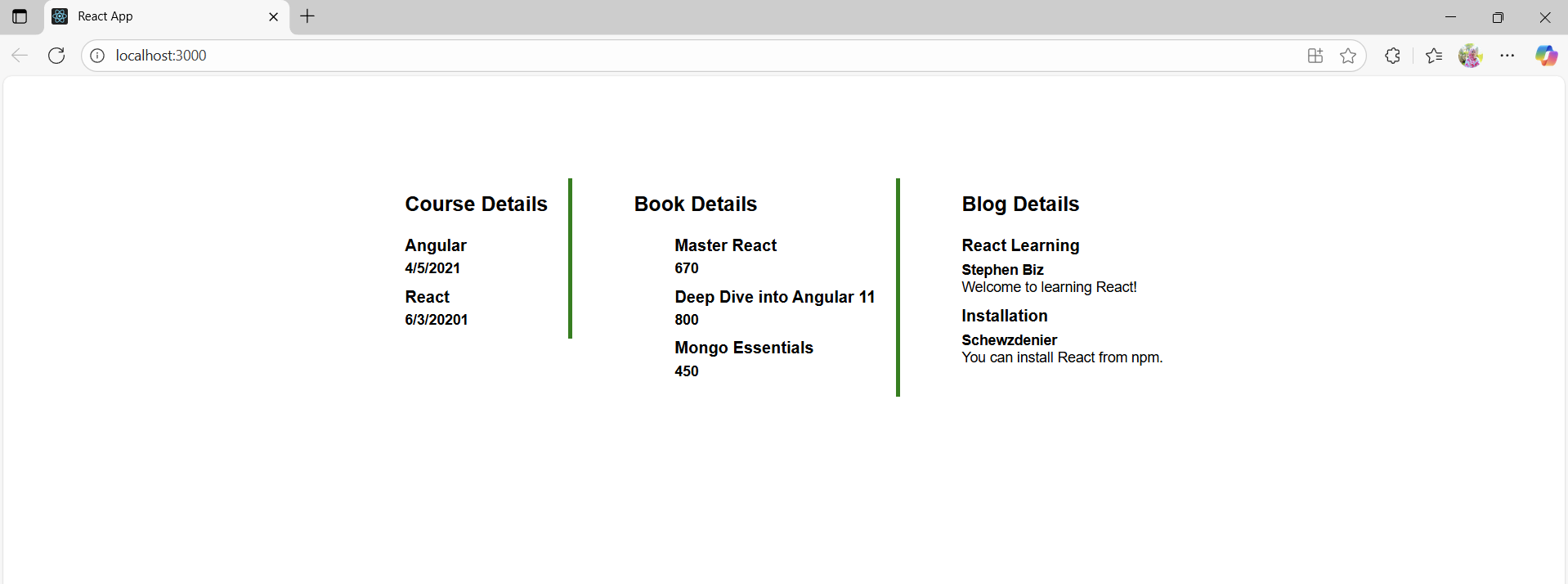
}

export default App;

**Step 8: Run the App**

npm start

**Output:**

****