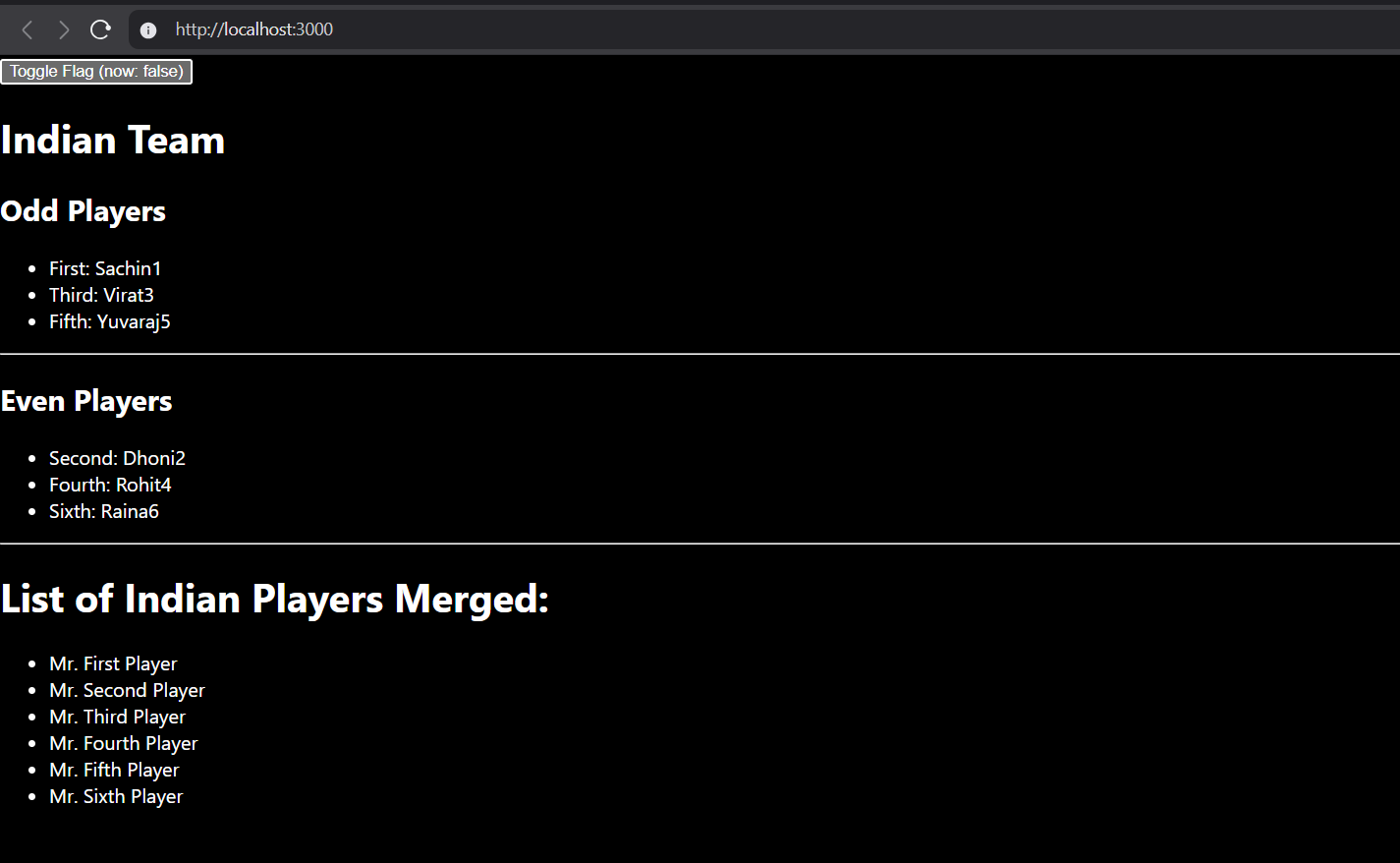
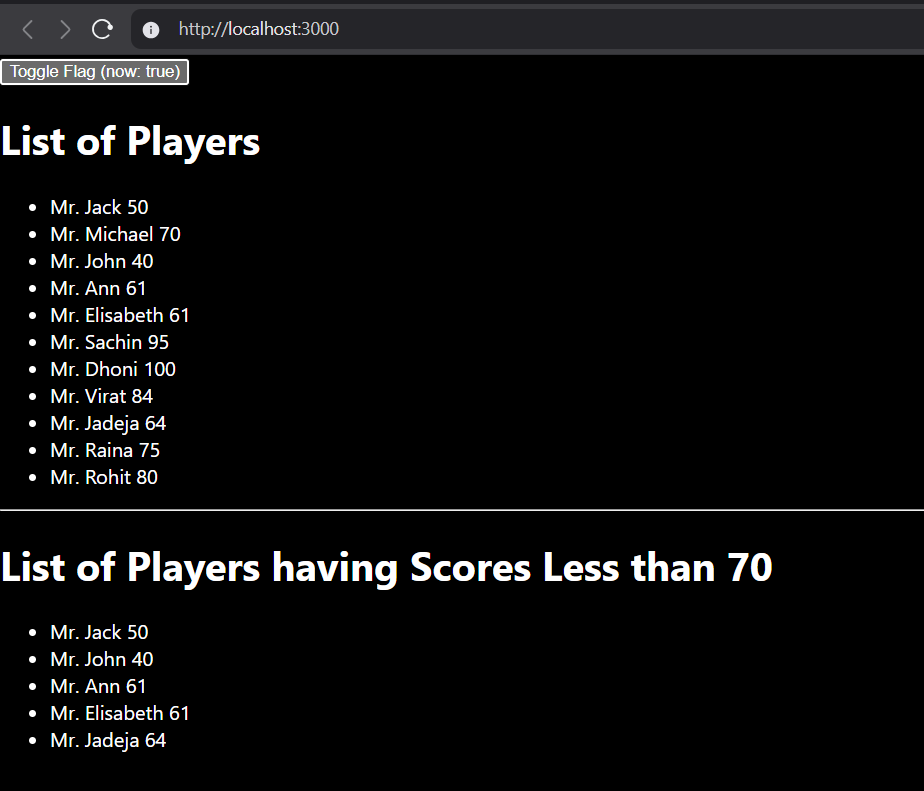
**ReactJS-HOL Week 7**

Superset ID: 6363229

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

Output:



Code:

App.js

import { useState } from 'react';

import ListofPlayers from './ListofPlayers';

import Scorebelow70 from './Scorebelow70';

import OddPlayers from './OddPlayers';

import EvenPlayers from './EvenPlayers';

import ListofIndianPlayers from './ListofIndianPlayers';

export default function App() {

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

  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 }

  ];

  const IndianTeam = ['Sachin1', 'Dhoni2', 'Virat3', 'Rohit4', 'Yuvaraj5', 'Raina6'];

  return (

    <>

      <button onClick={() => setFlag(!flag)}>

        Toggle Flag (now: {flag.toString()})

      </button>

      {flag ? (

        <div>

          <h1>List of Players</h1>

          <ListofPlayers players={players} />

          <hr />

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

          <Scorebelow70 players={players} />

        </div>

      ) : (

        <div>

          <h1>Indian Team</h1>

          <h2>Odd Players</h2>

          <OddPlayers IndianTeam={IndianTeam} />

          <hr />

          <h2>Even Players</h2>

          <EvenPlayers IndianTeam={IndianTeam} />

          <hr />

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

          <ListofIndianPlayers IndianPlayers={IndianTeam} />

        </div>

      )}

    </>

  );

}

ListofPlayers.js

export default function ListofPlayers({ players }) {

  return (

    <ul>

      {players.map((item, i) => (

        <li key={i}>

          Mr. {item.name} {item.score}

        </li>

      ))}

    </ul>

  );

}

IndianPlayers.js

const T20Players = ['First Player', 'Second Player', 'Third Player'];

const RanjiTrophyPlayers = ['Fourth Player', 'Fifth Player', 'Sixth Player'];

const merged = [...T20Players, ...RanjiTrophyPlayers];

export default function ListofIndianPlayers({ IndianPlayers }) {

  return (

    <ul>

      {merged.map((p, i) => (

        <li key={i}>Mr. {p}</li>

      ))}

    </ul>

  );

}

1. 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.

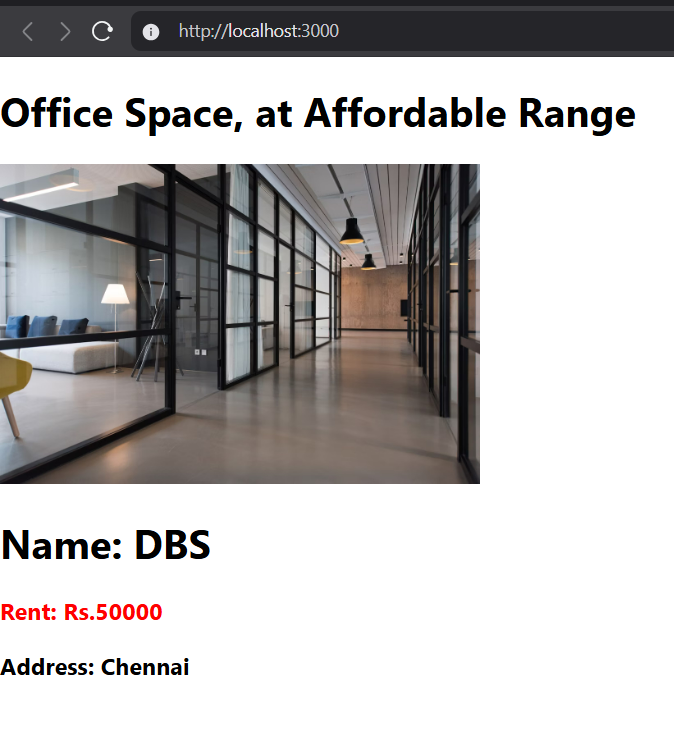
Attribute to display the image of the office space

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.

Output:



Code:

src/App.js

import React from 'react';

const offices = [

  { Name: 'DBS', Rent: 50000, Address: 'Chennai' }

];

export default function App() {

  return (

    <div>

      <h1>Office Space, at Affordable Range</h1>

      <img

        src="https://images.unsplash.com/photo-1497366754035-f200968a6e72?q=80&w=1169&auto=format&fit=crop"

        alt="Office"

        style={{ width: '25%' }}

      />

      {offices.map((item, i) => (

        <div key={i}>

          <h1>Name: {item.Name}</h1>

          <h3 style={{ color: item.Rent <= 60000 ? 'red' : 'green' }}>

            Rent: Rs.{item.Rent}

          </h3>

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

        </div>

      ))}

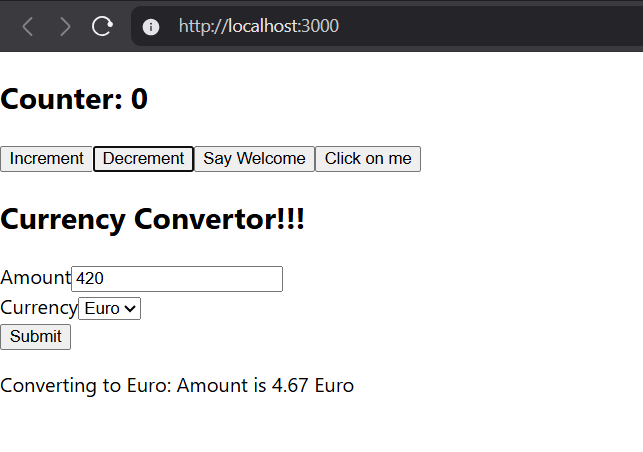
    </div>

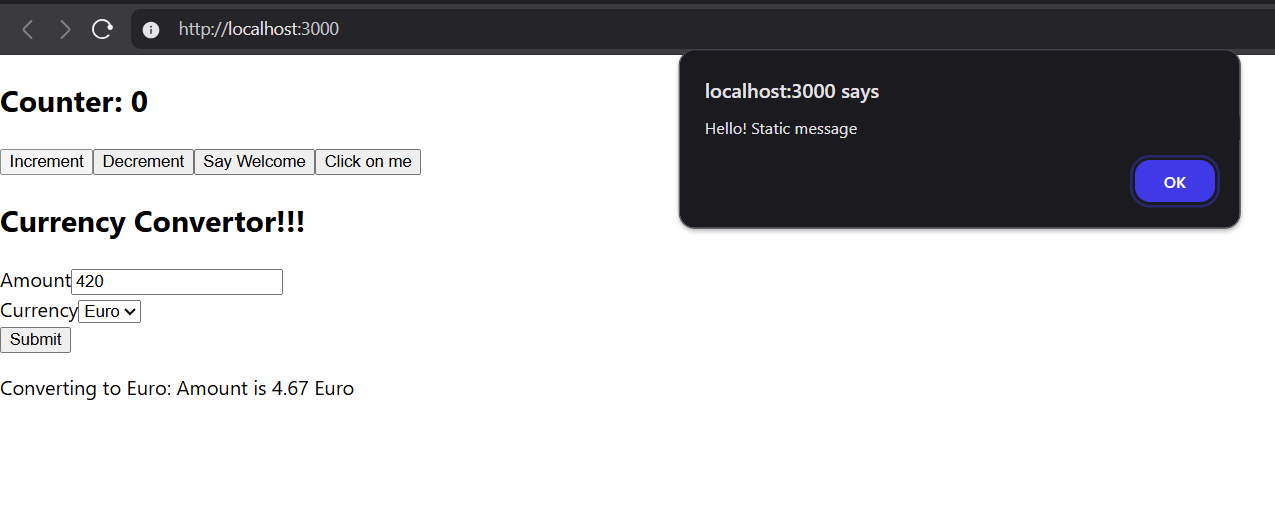
  );

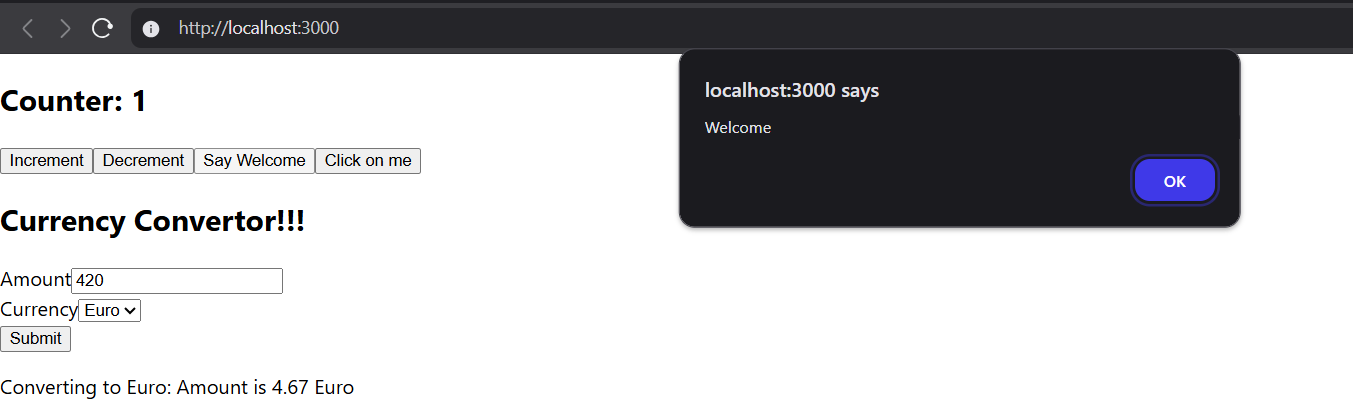
}

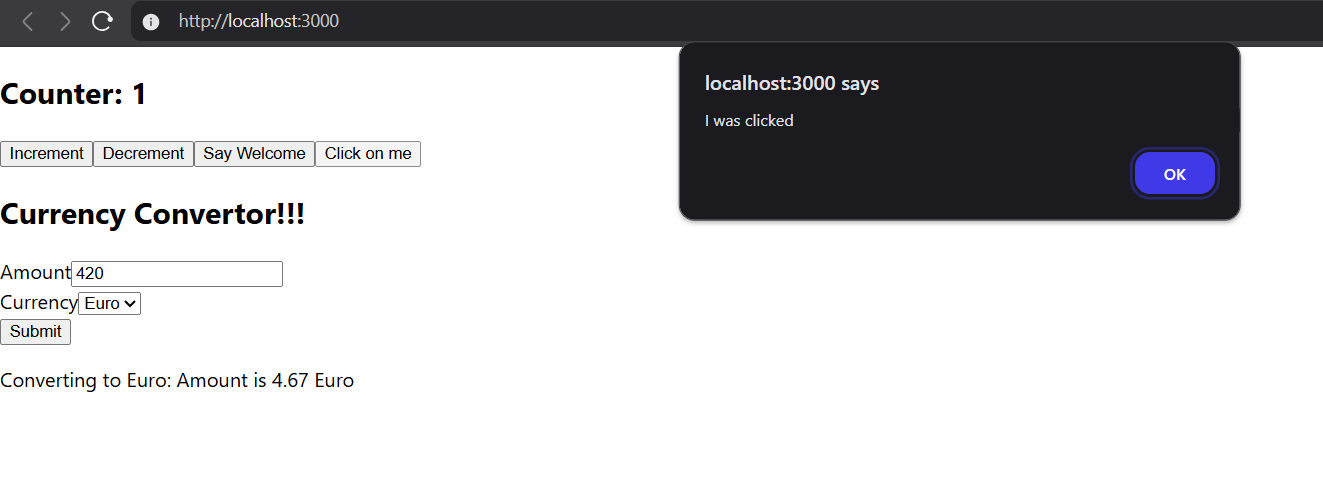
1. Create a React Application “eventexamplesapp” to handle various events of the form elements in HTML.
2. Create “Increment” button to increase the value of the counter and “Decrement” button to decrease the value of the counter. The “Increase” button should invoke multiple methods.
   1. To increment the value
   2. Say Hello followed by a static message.

Output:









Code:

### **src/App.js**

export { default } from './EventExamples';

Src/Counter.js

import React, { Component } from 'react';

export default class Counter extends Component {

  state = { count: 0 };

  increment = () => {

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

    this.sayHello();

  };

  decrement = () => this.setState(({ count }) => ({ count: count - 1 }));

  sayHello = () => alert('Hello! Static message');

  render() {

    return (

      <>

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

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

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

      </>

    );

  }

}

Src/Saywelcome.js

import React from 'react';

export default function SayWelcome() {

  const say = (msg) => alert(msg);

  return <button onClick={() => say('Welcome')}>Say Welcome</button>;

}

Src/clickme.js

import React from 'react';

export default function ClickMe() {

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

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

}

Src/currencyconverter.js

import React, { useState } from 'react';

export default function CurrencyConverter() {

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

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

  const [result, setResult]   = useState('');

  const handleSubmit = (e) => {

    e.preventDefault();

    const rate = 90; // fixed INR → EUR

    setResult((amount / rate).toFixed(2));

  };

  return (

    <>

      <h2>Currency Convertor!!!</h2>

      <form onSubmit={handleSubmit}>

        <label>

          Amount

          <input value={amount} onChange={(e) => setAmount(e.target.value)} />

        </label>

        <br />

        <label>

          Currency

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

            <option>Euro</option>

          </select>

        </label>

        <br />

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

      </form>

      {result && (

        <p>Converting to Euro: Amount is {result} Euro</p>

      )}

    </>

  );

}

src/EventExamples.js

import React, { Component } from 'react';

import Counter from './Counter';

import SayWelcome from './SayWelcome';

import ClickMe from './ClickMe';

import CurrencyConverter from './CurrencyConverter';

export default class EventExamples extends Component {

  render() {

    return (

      <>

        <Counter />

        <SayWelcome />

        <ClickMe />

        <CurrencyConverter />

      </>

    );

  }

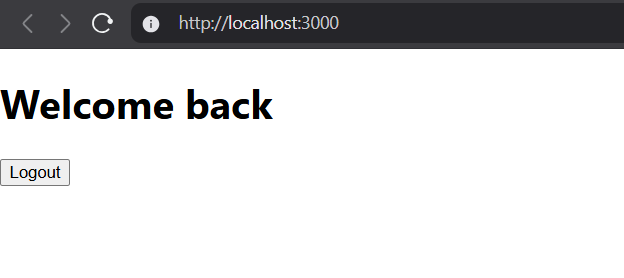
}

1. 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.

Output:





Codes:

Src/index.js

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import App from './App';

import reportWebVitals from './reportWebVitals';

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

root.render(<App />);

root.render(

  <React.StrictMode>

    <App />

  </React.StrictMode>

);

reportWebVitals();

Src/App.js

import React, { useState } from 'react';

import LoginButton  from './LoginButton';

import LogoutButton from './LogoutButton';

import Greeting     from './Greeting';

export default function App() {

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

  return (

    <div>

      {isLoggedIn ? (

        <>

          <Greeting isLoggedIn={isLoggedIn} />

          <LogoutButton onClick={() => setIsLoggedIn(false)} />

        </>

      ) : (

        <>

          <Greeting isLoggedIn={isLoggedIn} />

          <LoginButton onClick={() => setIsLoggedIn(true)} />

        </>

      )}

    </div>

  );

}

Src/LogoutButton.js

export default function LogoutButton({ onClick }) {

  return <button onClick={onClick}>Logout</button>;

}

Src/Greeting.js

import UserGreeting  from './UserGreeting';

import GuestGreeting from './GuestGreeting';

export default function Greeting({ isLoggedIn }) {

  return isLoggedIn ? <UserGreeting /> : <GuestGreeting />;

}

Src/UserGreeting.js

export default function UserGreeting() {

  return <h1>Welcome back</h1>;

}

Src/GuestGreeting.js

export default function GuestGreeting() {

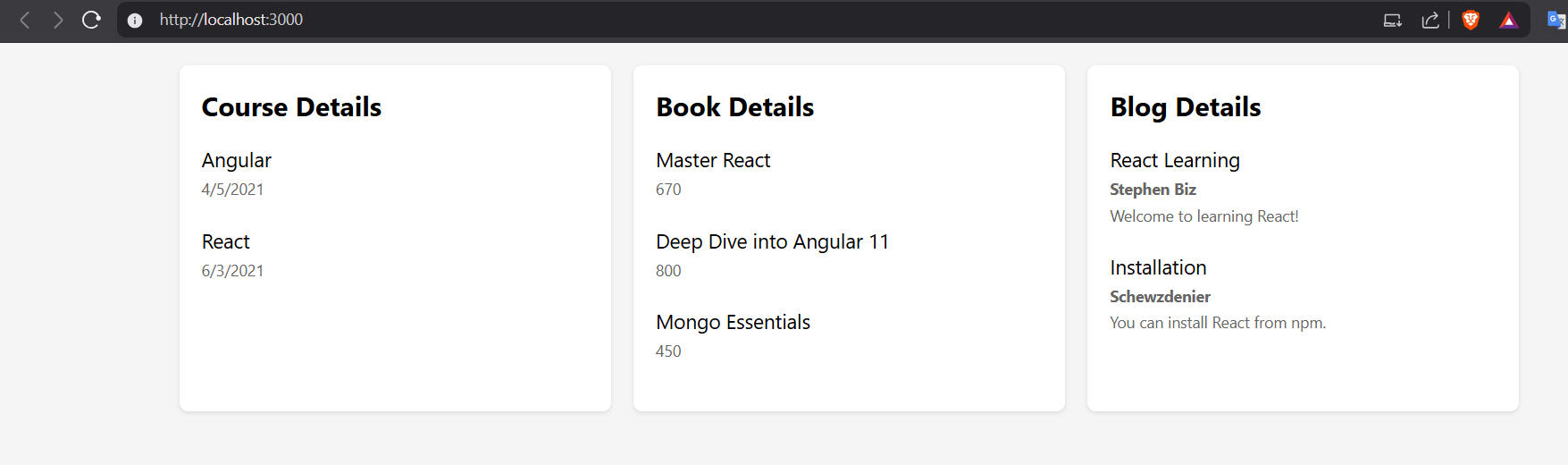
  return <h1>Please sign up.</h1>;

}

1. Create a React App named “bloggerapp” in with 3 components.
2. Book Details
3. Blog Details
4. Course Details

Implement this with as many ways possible of Conditional Rendering.

Output:



Codes:

Src/App.js

import React, { useState } from 'react';

import './App.css';

// Data

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 }

];

const courses = [

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

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

];

const blogs = [

  {

    title: 'React Learning',

    author: 'Stephen Biz',

    content: 'Welcome to learning React!'

  },

  {

    title: 'Installation',

    author: 'Schewzdenier',

    content: 'You can install React from npm.'

  }

];

// Components

const BookDetails = ({ books }) => (

  <div className="column">

    <h2>Book Details</h2>

    {books.map(book => (

      <div key={book.id} className="item">

        <h3>{book.bname}</h3>

        <p>{book.price}</p>

      </div>

    ))}

  </div>

);

const CourseDetails = ({ courses }) => (

  <div className="column">

    <h2>Course Details</h2>

    {courses.map((course, index) => (

      <div key={index} className="item">

        <h3>{course.name}</h3>

        <p>{course.date}</p>

      </div>

    ))}

  </div>

);

const BlogDetails = ({ blogs }) => (

  <div className="column">

    <h2>Blog Details</h2>

    {blogs.map((blog, index) => (

      <div key={index} className="item">

        <h3>{blog.title}</h3>

        <p className="author">{blog.author}</p>

        <p className="content">{blog.content}</p>

      </div>

    ))}

  </div>

);

// Main App

function App() {

  const [showContent, setShowContent] = useState(true);

  return (

    <div className="App">

      {showContent && (

        <div className="container">

          <CourseDetails courses={courses} />

          <BookDetails books={books} />

          <BlogDetails blogs={blogs} />

        </div>

      )}

    </div>

  );

}

export default App;