**9.ReactJS-HOL**

**ListofPlayers.js:**

import React from 'react';

const ListofPlayers = () => {

const players = [

{ name: "Virat", score: 90 },

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

{ name: "Gill", score: 50 },

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

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

{ name: "Pant", score: 68 },

{ name: "Jadeja", score: 77 },

{ name: "Ashwin", score: 59 },

{ name: "Bumrah", score: 66 },

{ name: "Shami", score: 38 },

{ name: "Kuldeep", score: 82 },

];

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

return (

<div>

<h2>All Players</h2>

<ul>

{players.map((player, idx) => (

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

))}

</ul>

<h2>Players with Score &lt; 70</h2>

<ul>

{lowScorers.map((player, idx) => (

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

))}

</ul>

</div>

);

};

export default ListofPlayers;

**IndianPlayers.js:**

import React from 'react';

const IndianPlayers = () => {

const oddPlayers = [

{ position: "First", name: "Sachin1" },

{ position: "Third", name: "Virat3" },

{ position: "Fifth", name: "Yuvaraj5" },

];

const evenPlayers = [

{ position: "Second", name: "Dhoni2" },

{ position: "Fourth", name: "Rohit4" },

{ position: "Sixth", name: "Raina6" },

];

const firstPlayers = ["Mr. First Player", "Mr. Second Player", "Mr. Third Player"];

const secondPlayers = ["Mr. Fourth Player", "Mr. Fifth Player", "Mr. Sixth Player"];

const mergedPlayers = [...firstPlayers, ...secondPlayers];

return (

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

<h2>Odd Players</h2>

<ul>

{oddPlayers.map((player, idx) => (

<li key={idx}>

<strong>{player.position}</strong> : {player.name}

</li>

))}

</ul>

<h2>Even Players</h2>

<ul>

{evenPlayers.map((player, idx) => (

<li key={idx}>

<strong>{player.position}</strong> : {player.name}

</li>

))}

</ul>

<h2>List of Indian Players Merged:</h2>

<ul>

{mergedPlayers.map((player, idx) => (

<li key={idx}>{player}</li>

))}

</ul>

</div>

);

};

export default IndianPlayers;

**App.js:**

import React from 'react';

import IndianPlayers from './IndianPlayers';

function App() {

  return (

    <div style={{ padding: "20px" }}>

      <IndianPlayers />

    </div>

  );

}

export default App;

**index.js**

import React from 'react';

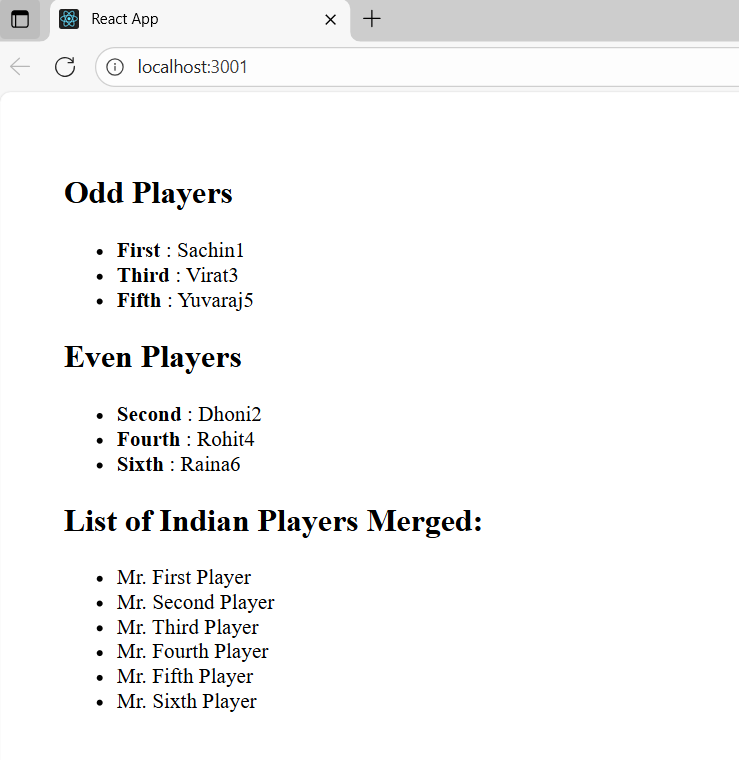
import ReactDOM from 'react-dom/client';

import App from './App';

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

root.render(<App />);

OUTPUT:



**10.ReactJS-HOL**

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

**App.js:**

import React from 'react';

import officeImage from './office.jpg'; // Place your image in src or public folder

const App = () => {

  const office = {

    name: "DBS",

    rent: 50000,

    address: "Chennai"

  };

  const headingStyle = {

    textAlign: 'center',

    color: '#111',

    fontWeight: 'bold',

    marginTop: '20px'

  };

  const imageStyle = {

    width: '300px',

    height: 'auto',

    display: 'block',

    margin: '20px auto'

  };

  const rentStyle = {

    color: 'red',

    fontWeight: 'bold'

  };

  const infoStyle = {

    textAlign: 'center',

    fontSize: '18px'

  };

  return (

    <div>

      <h1 style={headingStyle}>Office Space , at Affordable Range</h1>

      <img src="/office.jpg" alt="Office" style={imageStyle} />

      <div style={infoStyle}>

        <p><strong>Name:</strong> {office.name}</p>

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

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

      </div>

    </div>

  );

};

export default App;

**index.js:**

import React from 'react';

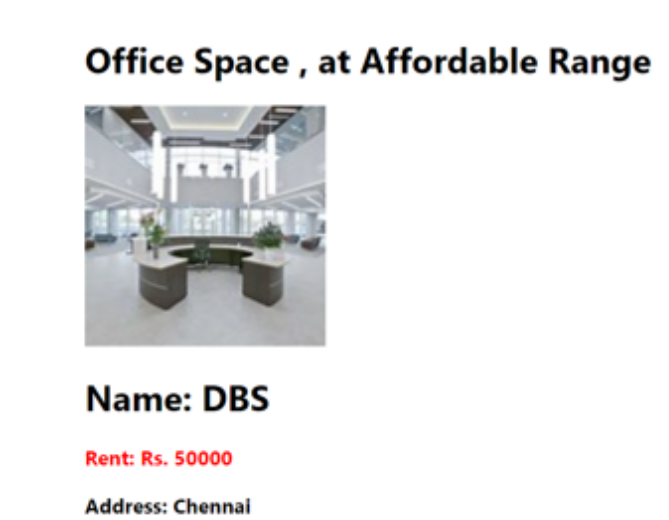
import ReactDOM from 'react-dom/client';

import App from './App';

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

root.render(<App />);

**OUTPUT**:



**11.ReactJS-HOL**

**App.js:**

import React, { Component } from 'react';

import CurrencyConvertor from './CurrencyConvertor';

class App extends Component {

constructor(props) {

super(props);

this.state = {

count: 1

};

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

this.handleClick = this.handleClick.bind(this);

}

increment = () => {

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

this.sayHello();

};

sayHello = () => {

alert("Hello! Counter was increased.");

};

decrement = () => {

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

};

handleWelcome(message) {

alert(`Say ${message}`);

}

handleClick(event) {

alert("I was clicked");

console.log("Synthetic Event:", event);

}

render() {

return (

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

<h1>{this.state.count}</h1>

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

<br /><br />

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

<br /><br />

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

<br /><br />

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

<hr />

<CurrencyConvertor />

</div>

);

}

}

export default App;

**CurrencyConvertor.js:**

import React, { Component } from 'react';

class CurrencyConvertor extends Component {

constructor(props) {

super(props);

this.state = {

amount: '',

currency: 'Euro'

};

this.handleSubmit = this.handleSubmit.bind(this);

this.handleChange = this.handleChange.bind(this);

}

handleChange(event) {

const { name, value } = event.target;

this.setState({ [name]: value });

}

handleSubmit(event) {

event.preventDefault();

const rupeeValue = parseFloat(this.state.amount);

let convertedValue = 0;

if (this.state.currency === "Euro") {

convertedValue = rupeeValue \* 80;

}

alert(`Converting to ${this.state.currency} Amount is ${convertedValue}`);

}

render() {

return (

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

<h2 style={{ color: "green" }}>Currency Convertor!!!</h2>

<form onSubmit={this.handleSubmit}>

<div>

<label>Amount: </label>

<input

type="text"

name="amount"

value={this.state.amount}

onChange={this.handleChange}

/>

</div>

<br />

<div>

<label>Currency: </label>

<select

name="currency"

value={this.state.currency}

onChange={this.handleChange}

>

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

{/\* You can add more options like USD etc. \*/}

</select>

</div>

<br />

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

</form>

</div>

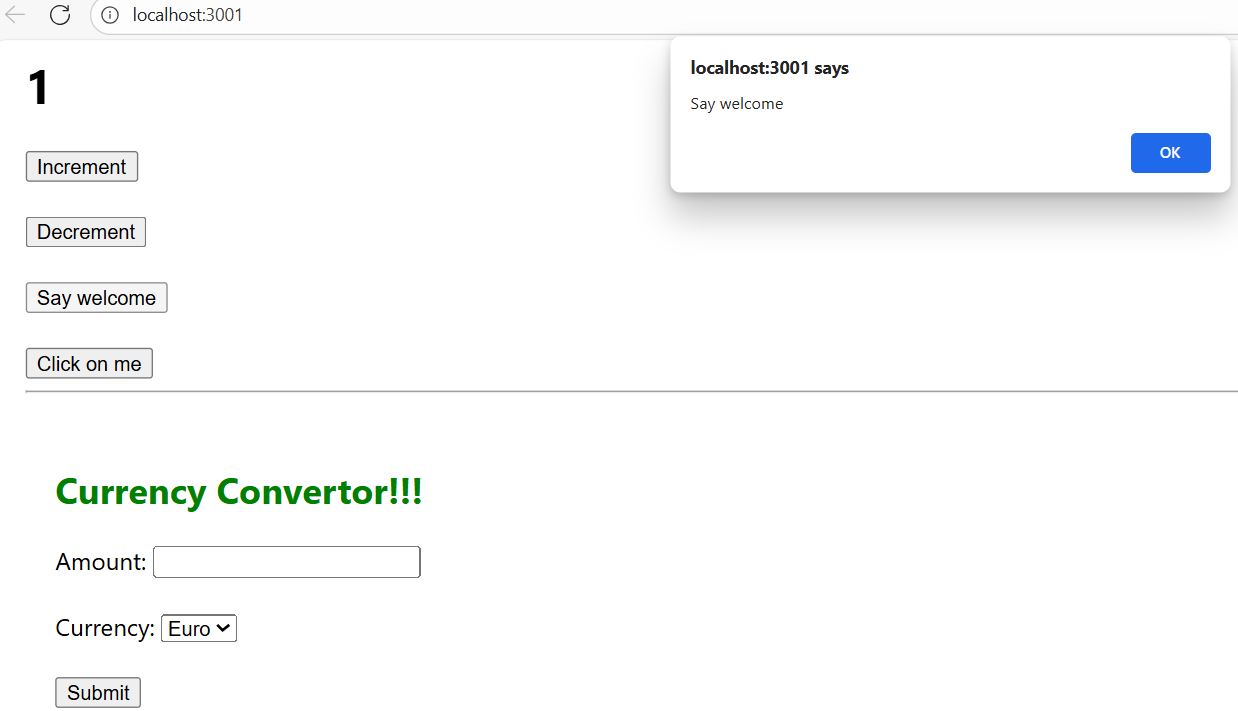
);

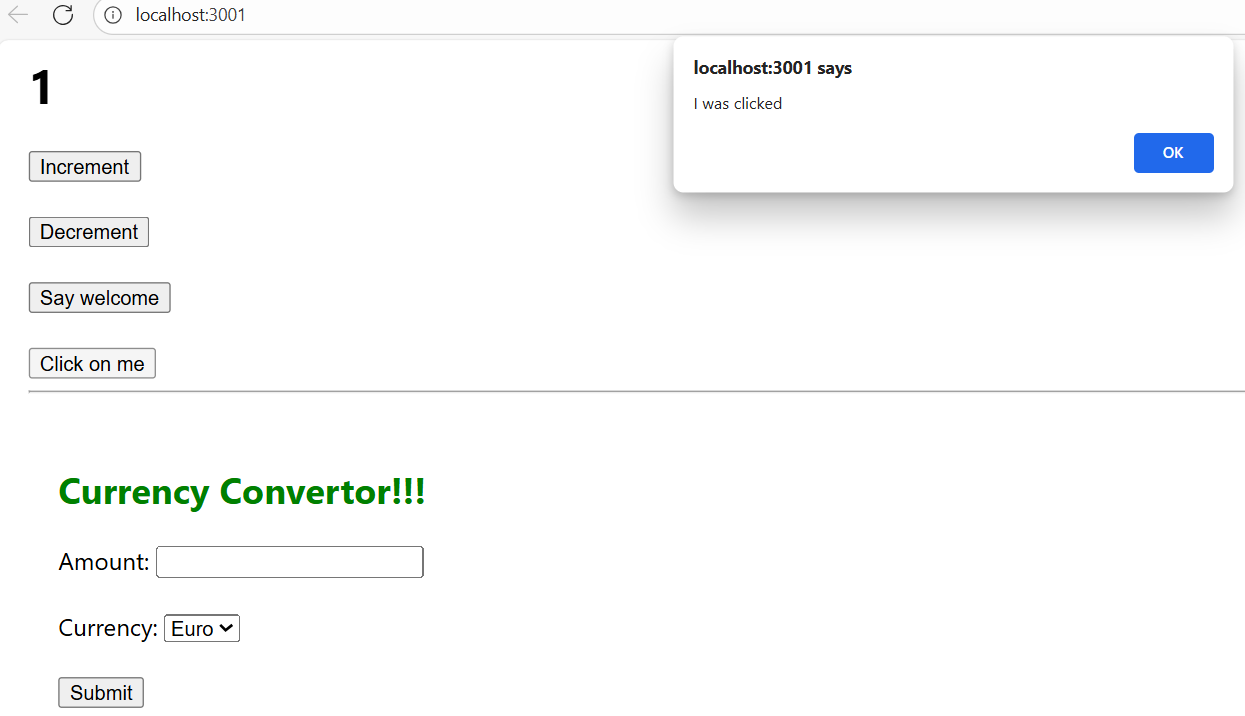
}

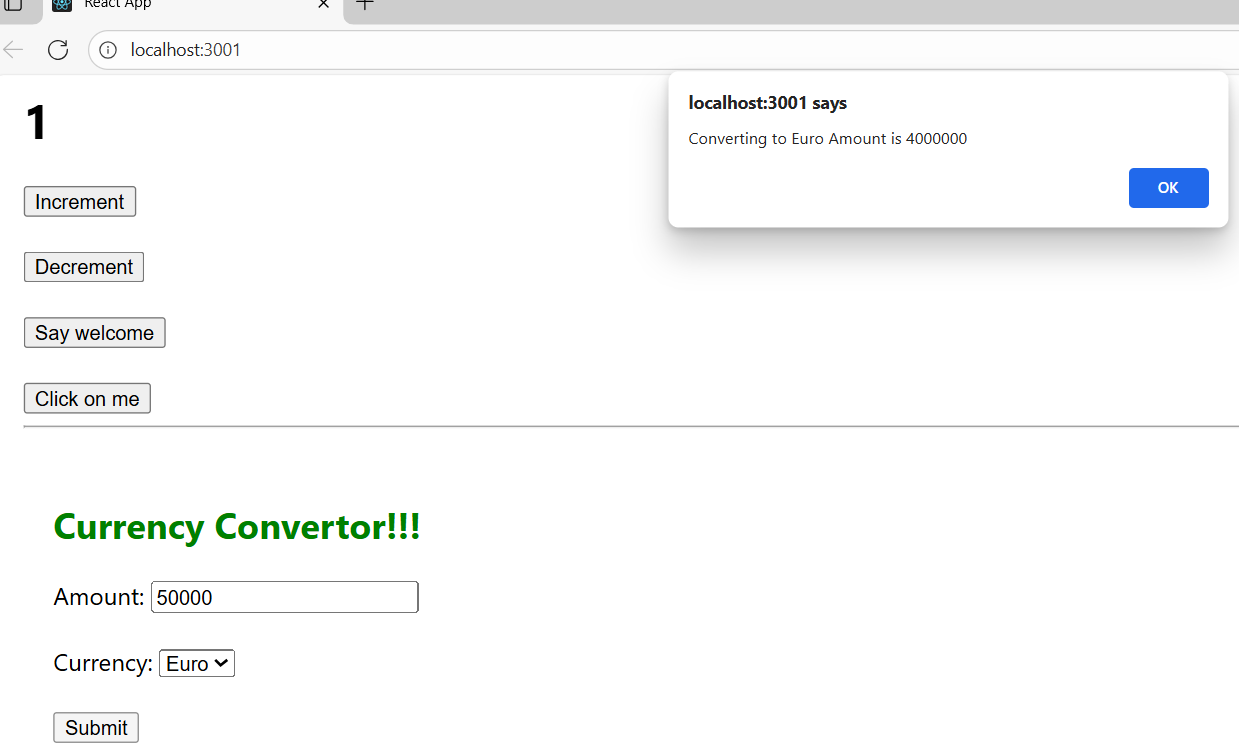
}

export default CurrencyConvertor;

**OUTPUT:**







12-ReactJS-HOL

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.

**App.js:**

import React, { Component } from 'react';

import GuestPage from './GuestPage';

import UserPage from './UserPage';

class App extends Component {

constructor(props) {

super(props);

this.state = {

isLoggedIn: false

};

}

handleLogin = () => {

this.setState({ isLoggedIn: true });

};

handleLogout = () => {

this.setState({ isLoggedIn: false });

};

render() {

let page;

if (this.state.isLoggedIn) {

page = <UserPage onLogout={this.handleLogout} />;

} else {

page = <GuestPage onLogin={this.handleLogin} />;

}

return (

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

{page}

</div>

);

}

}

export default App;

**GuestPage.js:**

import React from 'react';

const GuestPage = ({ onLogin }) => {

return (

<div>

<h1>Welcome Guest</h1>

<p>Please login to book tickets.</p>

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

</div>

);

};

export default GuestPage;

**UserPage.js:**

import React from 'react';

const UserPage = ({ onLogout }) => {

return (

<div>

<h1>Welcome back</h1>

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

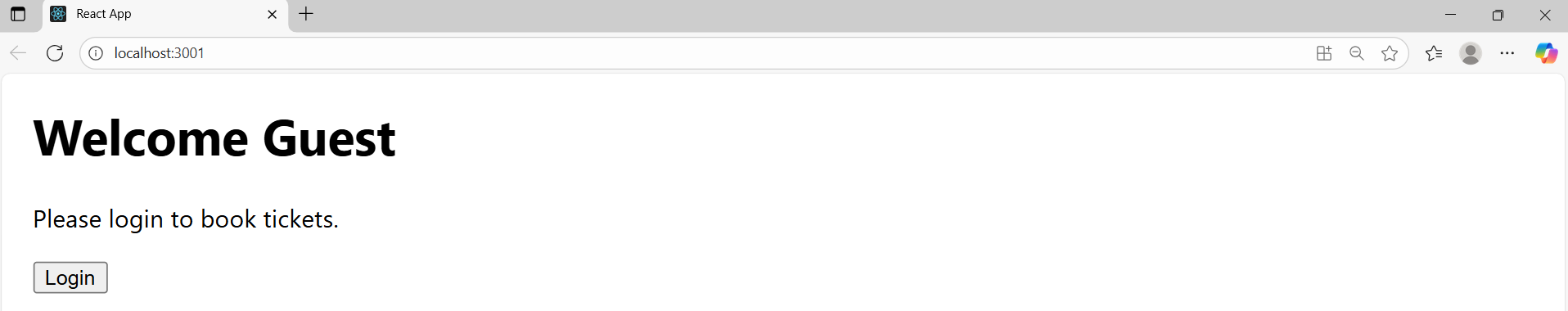
</div>

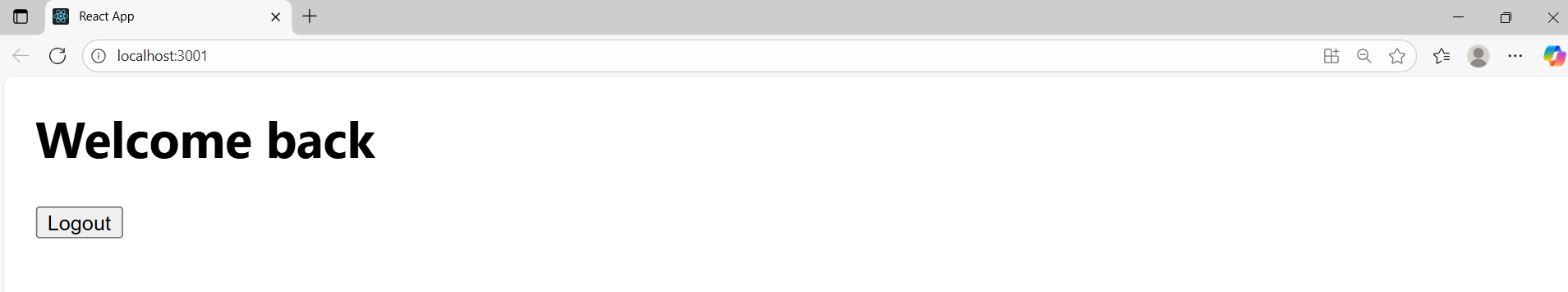
);

};

export default UserPage;

**OUTPUT:**





**13.ReactJS-HOL**

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 from 'react';

import CourseDetails from './CourseDetails';

import BookDetails from './BookDetails';

import BlogDetails from './BlogDetails';

import './App.css';

function App() {

return (

<div className="container">

<CourseDetails />

<BookDetails />

<BlogDetails />

</div>

);

}

export default App;

**CourseDetails.js:**

import React from 'react';

const courses = [

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

{ name: 'React', date: '6/3/2021' }

];

const CourseDetails = () => {

return (

<div className="column">

<h2>Course Details</h2>

{courses.length > 0 ? (

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

<div key={index}>

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

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

</div>

))

) : (

<p>No courses available</p>

)}

</div>

);

};

export default CourseDetails;

**BookDetails.js:**

import React from 'react';

const books = [

{ name: 'Master React', price: 670 },

{ name: 'Deep Dive into Angular 11', price: 800 },

{ name: 'Mongo Essentials', price: 450 }

];

const BookDetails = () => {

return (

<div className="column">

<h2>Book Details</h2>

{books.length > 0 &&

books.map((book, index) => (

<div key={index}>

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

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

</div>

))}

</div>

);

};

export default BookDetails;

**BlogDetails.js:**

import React from 'react';

const blogs = [

{

title: 'React Learning',

author: 'Stephen Biz',

content: 'Welcome to learning React!'

},

{

title: 'Installation',

author: 'Schwezdenier',

content: 'You can install React from npm.'

}

];

const BlogDetails = () => {

return (

<div className="column">

<h2>Blog Details</h2>

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

<div key={index}>

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

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

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

</div>

))}

</div>

);

};

export default BlogDetails;

**App.css:**

.container {

display: flex;

justify-content: space-around;

padding: 30px;

}

.column {

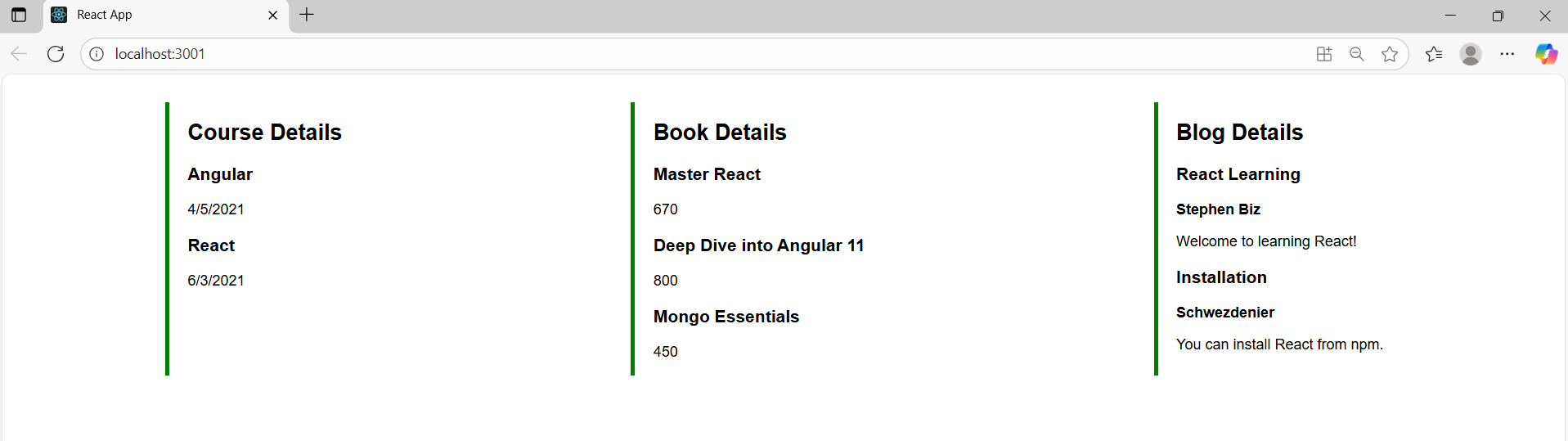
border-left: 5px solid green;

padding: 0 20px;

font-family: sans-serif;

}

**OUTPUT:**



**14.ReactJS-HOL:**

**ThemeContent.js:**

import { createContext } from 'react';

const ThemeContext = createContext('light');

export default ThemeContext;

**App.js:**

import React, { useState } from 'react';

import EmployeeList from './EmployeeList';

import ThemeContext from './ThemeContext';

function App() {

const [theme, setTheme] = useState('light');

return (

<ThemeContext.Provider value={theme}>

<div className="App">

<h1>Employee Management</h1>

<EmployeeList />

</div>

</ThemeContext.Provider>

);

}

export default App;

**EmployeeList.js:**

import React from 'react';

import EmployeeCard from './EmployeeCard';

function EmployeeList() {

const employees = [

{ id: 1, name: 'Alice', position: 'Developer' },

{ id: 2, name: 'Bob', position: 'Designer' },

];

return (

<div>

<h2>Employee List</h2>

{employees.map((emp) => (

<EmployeeCard key={emp.id} employee={emp} />

))}

</div>

);

}

export default EmployeeList;

**EmployeeCard.js:**

import React, { useContext } from 'react';

import ThemeContext from './ThemeContext';

function EmployeeCard({ employee }) {

const theme = useContext(ThemeContext);

return (

<div style={{ margin: '10px 0' }}>

<h3>{employee.name}</h3>

<p>{employee.position}</p>

<button className={theme === 'dark' ? 'btn-dark' : 'btn-light'}>

View Profile

</button>

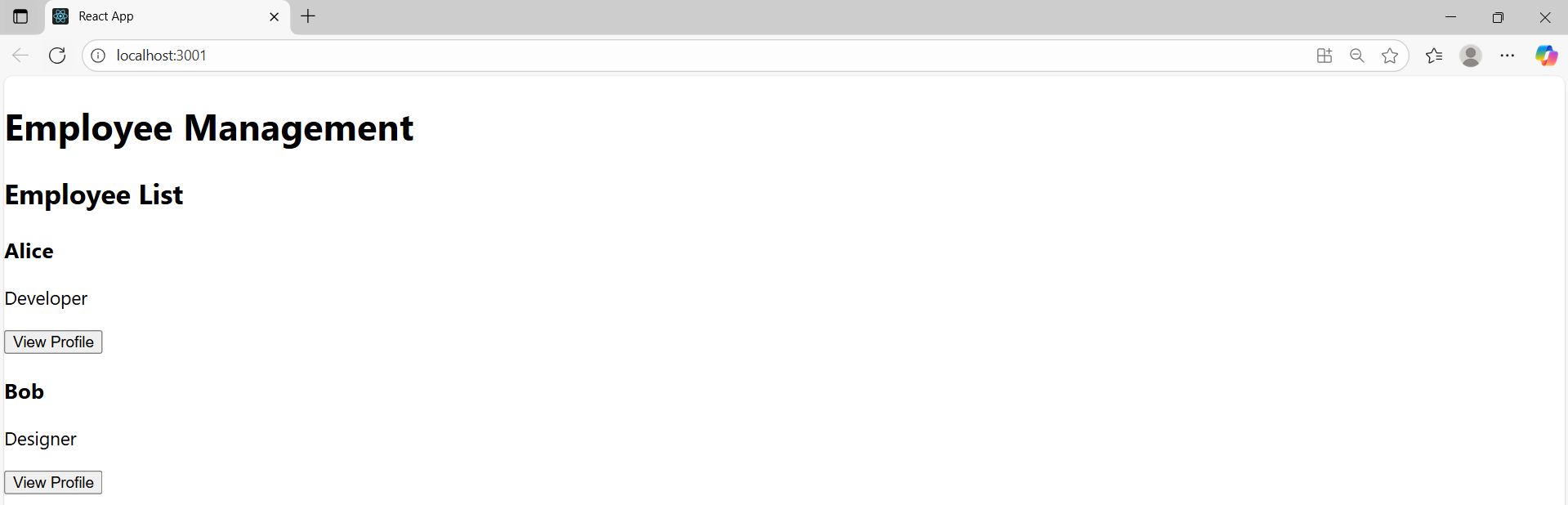
</div>

);

}

export default EmployeeCard;

**OUTPUT:**



**15.ReactJS-HOL**

Create a React App named “ticketraisingapp” which will help to raise a complaint and get it resolved.

Create a component named “ComplaintRegister” with a form containing a textbox to enter the employee name and a textarea to enter the complaint. Use “handleSubmit” event of the button to submit the complaint and generate a Reference number for further follow ups in the alert box.

**App.js:**

import React from 'react';

import ComplaintRegister from './ComplaintRegister';

function App() {

return (

<div>

<ComplaintRegister />

</div>

);

}

export default App;

**ComplaintRegister.js:**

import React, { useState } from 'react';

function ComplaintRegister() {

const [name, setName] = useState('');

const [complaint, setComplaint] = useState('');

const handleSubmit = (e) => {

e.preventDefault();

const txnId = Math.floor(Math.random() \* 1000);

alert(`Thanks ${name}\nYour Complaint was Submitted.\nTransaction ID is: ${txnId}`);

setName('');

setComplaint('');

};

return (

<div style={{ textAlign: 'center', marginTop: '50px' }}>

<h2 style={{ color: 'red' }}>Register your complaints here!!!</h2>

<form onSubmit={handleSubmit}>

<div>

<label>Name: </label>

<input

type="text"

value={name}

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

required

/>

</div>

<br />

<div>

<label>Complaint: </label>

<textarea

value={complaint}

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

required

/>

</div>

<br />

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

</form>

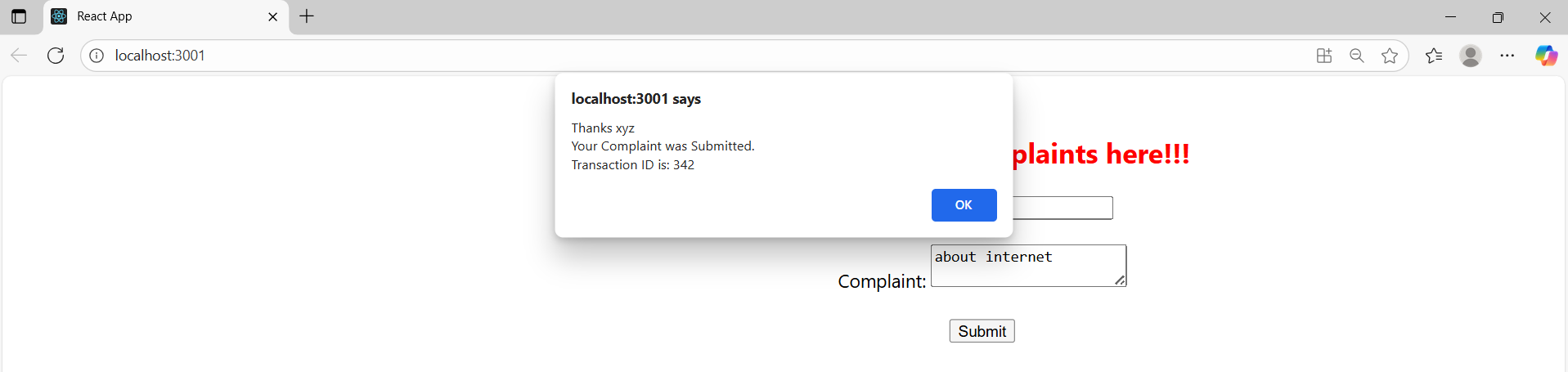
</div>

);

}

export default ComplaintRegister;

**OUTPUT:**



16.ReactJS-HOL:

Create a React App named “mailregisterapp” which will have a component named “register.js”. Create a form which accepts the name, email and password and validate the fields as per the following:

1. Name should have atleast 5 characters
2. Email should have @ and .
3. Password should have atleast 8 characters.

Ensure that validations are implemented through eventhandle and eventsubmit of a form.

**App.js:**

import React from 'react';

import Register from './Register';

function App() {

return (

<div>

<Register />

</div>

);

}

export default App;

**Register.js:**

import React, { useState } from 'react';

function Register() {

const [name, setName] = useState('');

const [email, setEmail] = useState('');

const [password, setPassword] = useState('');

const handleSubmit = (e) => {

e.preventDefault();

if (name.length < 5) {

alert("Name must be at least 5 characters!");

return;

}

if (!email.includes('@') || !email.includes('.')) {

alert("Email is not valid!");

return;

}

if (password.length < 8) {

alert("Password must be at least 8 characters!");

return;

}

alert("Form submitted successfully!");

setName('');

setEmail('');

setPassword('');

};

return (

<div style={{ textAlign: 'center', marginTop: '50px' }}>

<h2 style={{ color: 'red' }}>Register Here!!!</h2>

<form onSubmit={handleSubmit}>

<div>

<label>Name: </label>

<input

type="text"

value={name}

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

required

/>

</div>

<br />

<div>

<label>Email: </label>

<input

type="text"

value={email}

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

required

/>

</div>

<br />

<div>

<label>Password: </label>

<input

type="password"

value={password}

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

required

/>

</div>

<br />

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

</form>

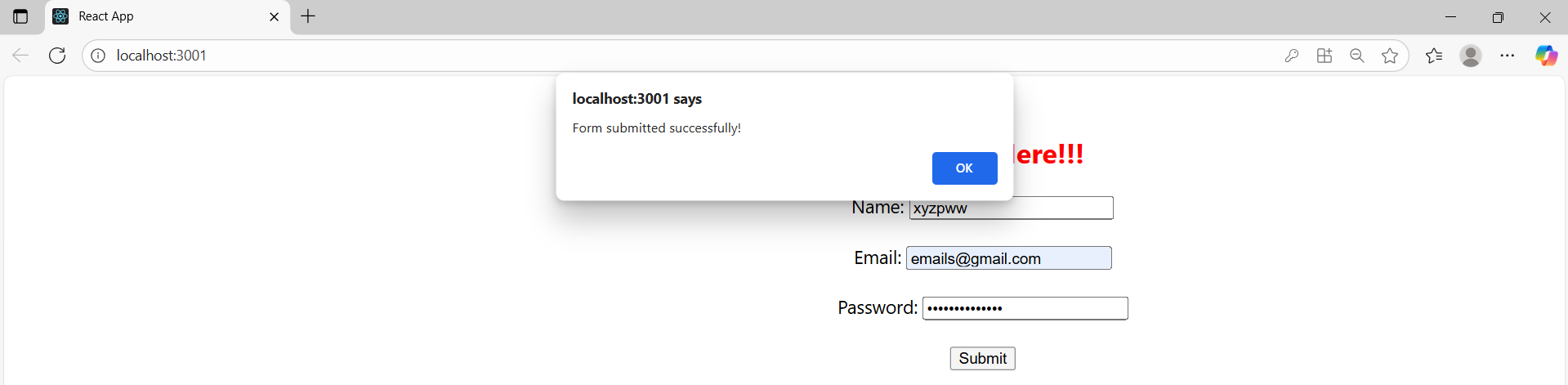
</div>

);

}

export default Register;

**OUTPUT:**



**17.React-HOL:**

Create a React Application “fetchuserapp” which will retrieve the user details from <https://api.randomuser.me/> and display the title, firstname and image of a user.

Create a component named “Getuser” and in the asynchronous method “ComponentDidMount ()” invoke the URL using fetch method and the response can be displayed in the render method of the component.

**App.js:**

import React from 'react';

import Getuser from './Getuser';

function App() {

return (

<div>

<Getuser />

</div>

);

}

export default App;

**Getuser.js:**

import React from 'react';

class Getuser extends React.Component {

constructor() {

super();

this.state = {

user: null

};

}

async componentDidMount() {

try {

const response = await fetch('https://api.randomuser.me/');

const data = await response.json();

const userData = data.results[0];

this.setState({

user: {

title: userData.name.title,

firstName: userData.name.first,

lastName: userData.name.last,

picture: userData.picture.medium

}

});

} catch (error) {

console.error('Error fetching user:', error);

}

}

render() {

const { user } = this.state;

return (

<div style={{ textAlign: 'center', marginTop: '50px' }}>

{user ? (

<>

<h2>

{user.title} {user.firstName} {user.lastName}

</h2>

<img src={user.picture} alt="User" />

</>

) : (

<p>Loading user...</p>

)}

</div>

);

}

}

export default Getuser;

**index.js:**

import React from 'react';

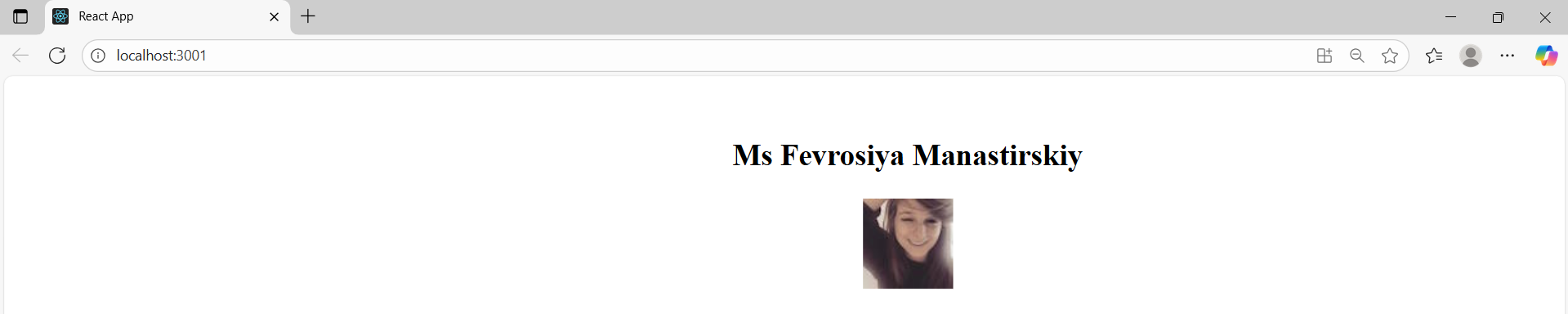
import ReactDOM from 'react-dom/client';

import App from './App';

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

root.render(<App />);

**OUTPUT:**



**18-ReactJS-HOL:**

My Academy team at Cognizant want to create a dashboard containing the details of ongoing and completed cohorts. A react application is created which displays the detail of the cohorts using React component. You are assigned the task of unit testing the component to ensure it’s free of bugs.

**setupTests.js:**

import { configure } from 'enzyme';

import Adapter from 'enzyme-adapter-react-16';

configure({ adapter: new Adapter() });

**CohortDetails.test.js**

import React from 'react';

import { shallow, mount } from 'enzyme';

import CohortDetails from './CohortDetails'; // adjust the path based on your project

import { CohortData } from './Cohort'; // adjust the path based on your project

import toJson from 'enzyme-to-json';

describe('Cohort Details Component', () => {

test('should create the component', () => {

const wrapper = shallow(<CohortDetails />);

expect(wrapper.exists()).toBeTruthy();

});

test('should initialize the props', () => {

const wrapper = mount(<CohortDetails cohort={CohortData[0]} />);

expect(wrapper.props().cohort).toEqual(CohortData[0]);

});

test('should display cohort code in h3', () => {

const wrapper = mount(<CohortDetails cohort={CohortData[0]} />);

const h3 = wrapper.find('h3');

expect(h3.text()).toBe(CohortData[0].cohortCode);

});

test('should always render same html', () => {

const wrapper = shallow(<CohortDetails cohort={CohortData[0]} />);

expect(toJson(wrapper)).toMatchSnapshot();

});

});

**19.ReactJS-HOL:**

As an intern at OpenAI you are assigned the task of creating and testing a React application which will fetch and display a list of repository names for a given user.

**GitClient.js:**

import axios from 'axios';

class GitClient {

static async getRepositories(username) {

const url = `https://api.github.com/users/${username}/repos`;

const response = await axios.get(url);

return response.data.map(repo => repo.name);

}

}

export default GitClient;

**App.js:**

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

import GitClient from './GitClient';

function App() {

const [repos, setRepos] = useState([]);

useEffect(() => {

GitClient.getRepositories('TechieSyed')

.then(data => setRepos(data))

.catch(err => console.error(err));

}, []);

return (

<div>

<h2>Git repositories of User - TechieSyed</h2>

<ul>

{repos.map((repo, index) => (

<li key={index}>{repo}</li>

))}

</ul>

</div>

);

}

export default App;

**GitClient.test.js:**

import axios from 'axios';

import GitClient from './GitClient';

jest.mock('axios');

describe('Git Client Tests', () => {

test('should return repository names for TechieSyed', async () => {

const mockRepos = [

{ name: 'Repo1' },

{ name: 'Repo2' },

{ name: 'Repo3' },

];

axios.get.mockResolvedValue({ data: mockRepos });

const result = await GitClient.getRepositories('TechieSyed');

expect(result).toEqual(['Repo1', 'Repo2', 'Repo3']);

expect(axios.get).toHaveBeenCalledWith('https://api.github.com/users/TechieSyed/repos');

});

});

**GitClient.test.js:**

import axios from 'axios';

import GitClient from './GitClient';

jest.mock('axios');

describe('Git Client Tests', () => {

test('should return repository names for TechieSyed', async () => {

const mockRepos = [

{ name: 'Repo1' },

{ name: 'Repo2' },

{ name: 'Repo3' },

];

axios.get.mockResolvedValue({ data: mockRepos });

const result = await GitClient.getRepositories('TechieSyed');

expect(result).toEqual(['Repo1', 'Repo2', 'Repo3']);

expect(axios.get).toHaveBeenCalledWith('https://api.github.com/users/TechieSyed/repos');

});

});

**OUTPUT:**

