1. **Write a program to configure routing with authentication with sharing information with and without login?**

App.js

import "./App.css";

import React from "react";

import { Link, Route, Routes } from "react-router-dom";

import Home from "../src/Pages/Home";

import Courses from "../src/Pages/Courses";

import Live from "../src/Pages/Live";

import Contact from "../src/Pages/Contact";

function App() {

  return (

    <div className="container">

<nav class="navbar bg-dark border-bottom border-body" data-bs-theme="dark">

<nav class="navbar navbar-expand-lg bg-body-tertiary">

  <div class="container-fluid">

    <a class="navbar-brand">AKASH SHETTY</a>

    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavDropdown" aria-controls="navbarNavDropdown" aria-expanded="false" aria-label="Toggle navigation">

      <span class="navbar-toggler-icon"></span>

    </button>

    <div class="collapse navbar-collapse" id="navbarNavDropdown">

      <ul class="navbar-nav">

        <li class="nav-item">

          <a class="nav-link active" aria-current="page">

            <Link to="/" class="list">

                Home

            </Link></a>

        </li>

        <li class="nav-item">

          <a class="nav-link">

              <Link to="/course" class="list">

                Courses

              </Link></a>

        </li>

        <li class="nav-item">

          <a class="nav-link">

              <Link to="/live" class="list">

                Live course

              </Link></a>

        </li>

        <li class="nav-item">

          <a class="nav-link">

              <Link to="/contact" class="list">

                Contact

              </Link></a>

        </li>

      </ul>

    </div>

  </div>

</nav>

</nav>

      {/\* Defining routes path and rendering components as element \*/}

      <Routes>

        <Route path="/" element={<Home />} />

        <Route path="/course" element={<Courses />} />

        <Route path="/live" element={<Live />} />

        <Route path="/contact" element={<Contact />} />

      </Routes>

    </div>

  );

}

export default App;

Home.js

import React from 'react';

const Home = () => {

  return <div className='container1'>

      <div class="box">

       This is the  <span color='red'>home page.</span> Public information can be displayed here.

      </div>

    </div>;

};

export default Home;

Cources.js

import React from 'react'

function Courses() {

  return (

    <div className='container1'>

    <div class="box">

     This is <span>COURCES</span> component

    </div>

  </div>

  )

}

export default Courses;

Live.js

import React from 'react'

function Live() {

  return (

    <div className='container1'>

    <div class="box">

      This is <span>LIVE</span> component

    </div>

  </div>

  )

}

export default Live;

Contact.js

import React from 'react'

function Contact() {

  return (

    <div className='container1'>

      <div class="box">

      This is <span>CONTACT</span> component

      </div>

    </div>

  )

}

export default Contact;

index.js

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import App from './App';

import reportWebVitals from './reportWebVitals';

import { BrowserRouter } from 'react-router-dom';

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

root.render(

  <BrowserRouter>

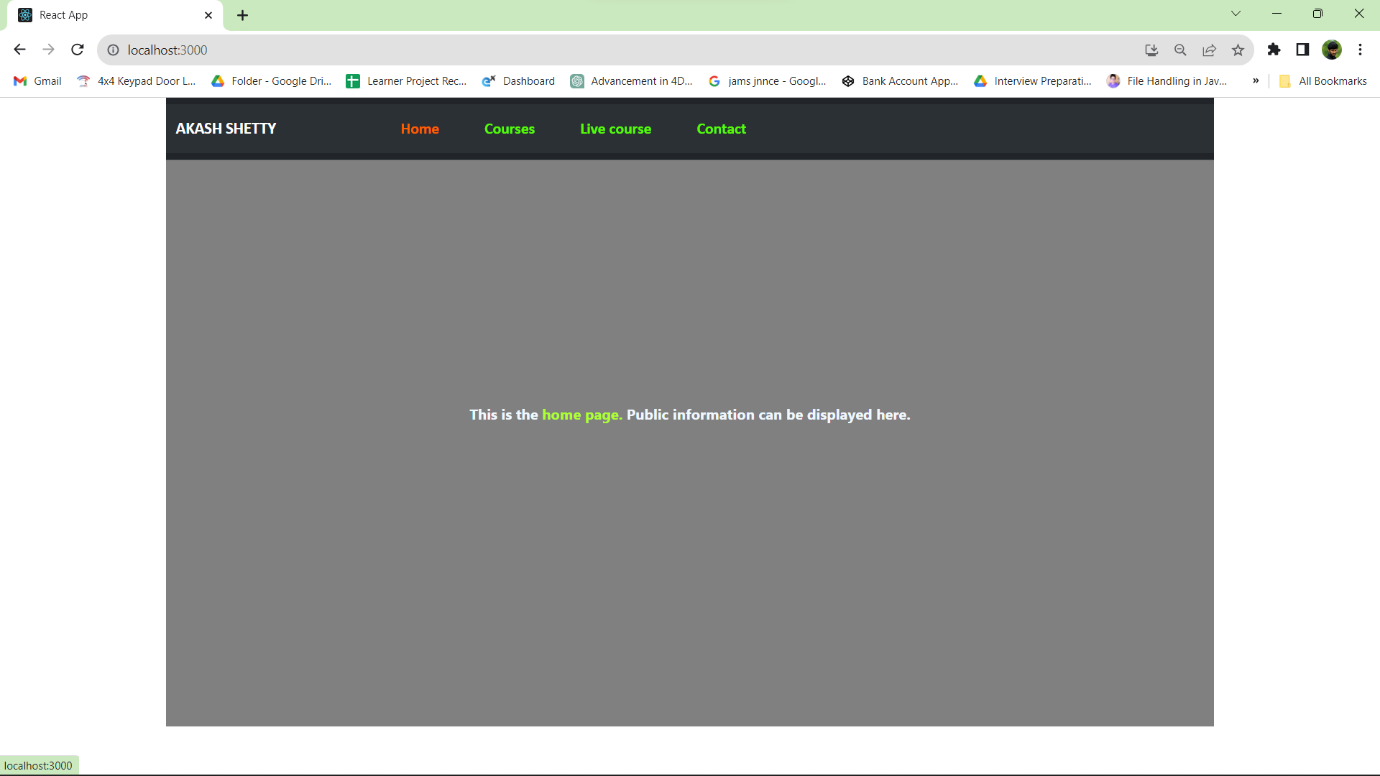
    <App />

  </BrowserRouter>

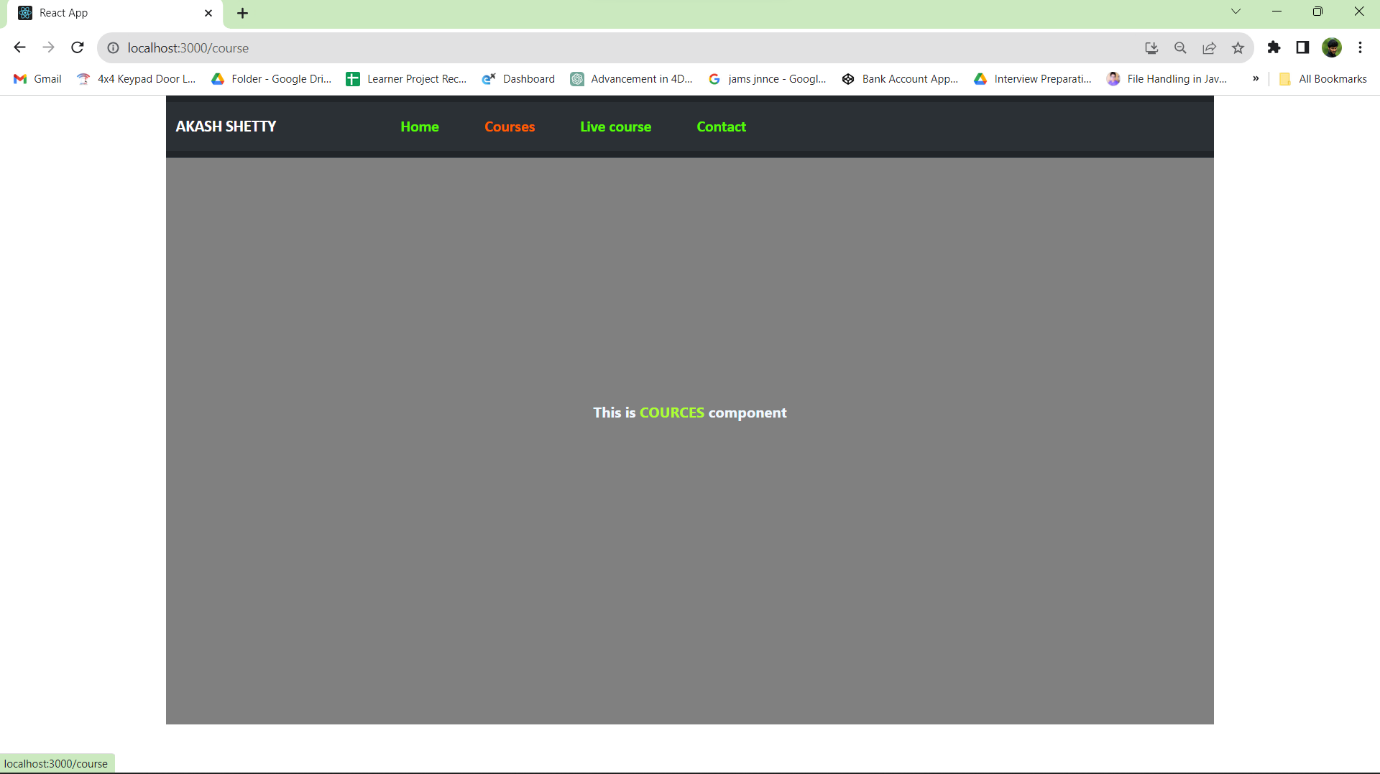
);

reportWebVitals();

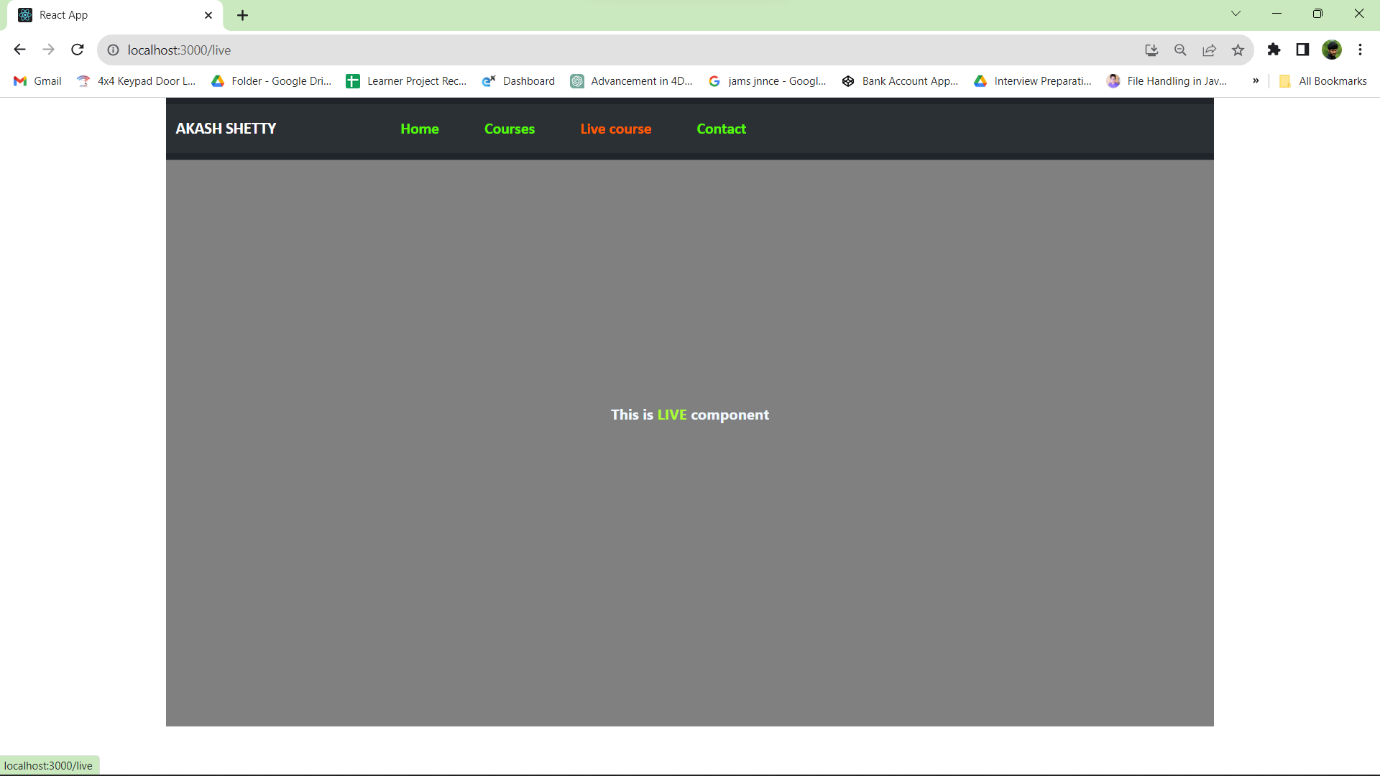
OUTPUT :



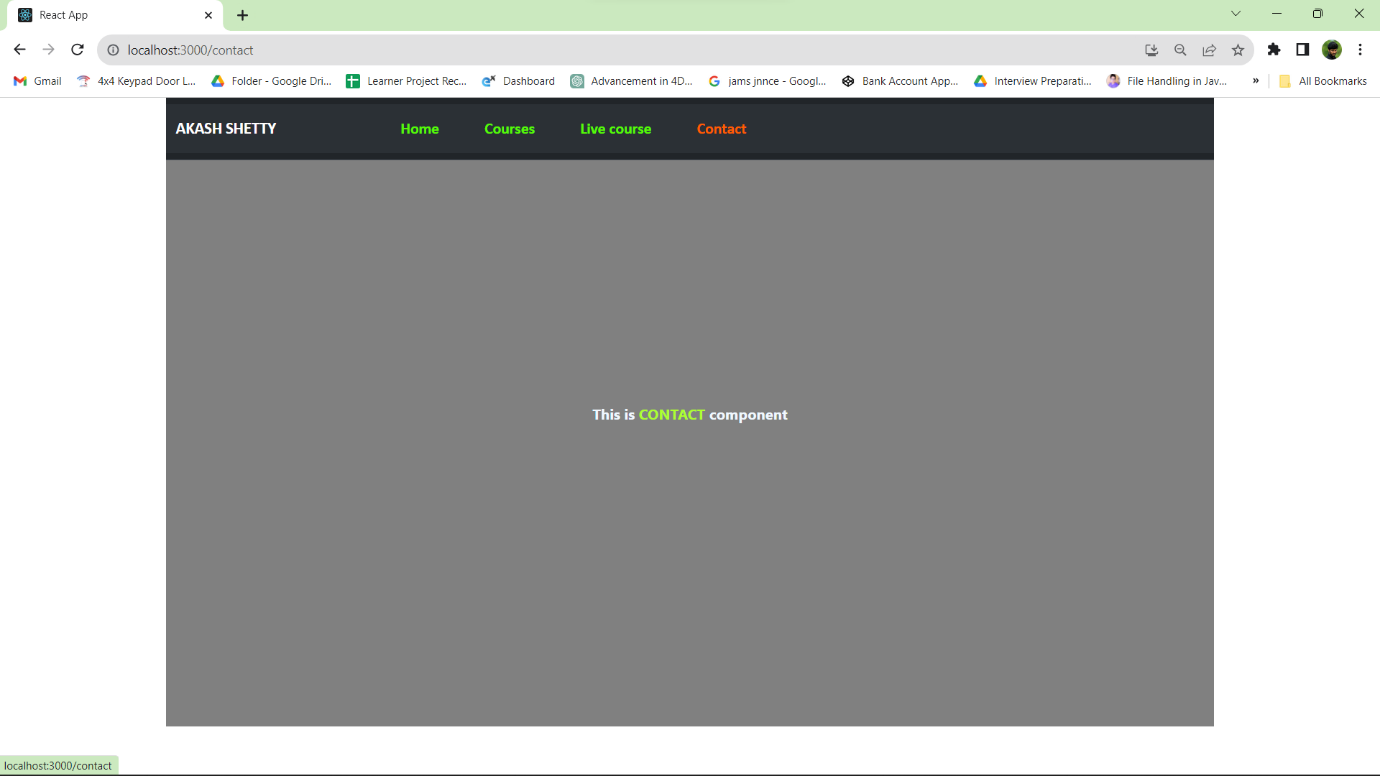
Home Page View (Default)



Courses page view



Live page view



Contact page view

2 ) **Write a program to create login and signup forms and on successful login need to show the dashboard with all logged in users?**

LoginForm.js

import React, { useState } from 'react';

// Simulated user data (replace with backend integration)

const usersData = [

    { id: 1, username: 'Akash\_Shetty', password: 'Akash22' },

    { id: 2, username: 'Akash', password: 'Akash33' }

    // Add more users as needed

  ];

const LoginForm = ({ onLogin }) => {

  const [username, setUsername] = useState('');

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

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

  const [loggedInUsers, setLoggedInUsers] = useState([]);

  const handleLogin = () => {

    //We can perform our Authentication part here with backend

    //Here i check for a hard-coded username = Akash\_Shetty, Akash and password = Akash22, Akash33

    const user = usersData.find((u) => u.username === username && u.password === password);

    if (user) {

      onLogin(username);

      setIsLoggedIn(true);

      setLoggedInUsers([...loggedInUsers, user]);

      setUsername('');

      setPassword('');

      alert(`${username} you have logged in successfully..!!`)

    } else {

      alert('Invalid credentials. Please try again.');

    }

  };

  return (

    <div class="hero">

    <div class="form-box">

        <div class="btn-box">

            <div id="btn">

            </div>

            <button type="button" class="toggle-btn">Login</button>

            <button type="button" class="toggle-btn">Signup</button>

      </div>

      <input

        type="text"

        placeholder="Username"

        value={username}

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

      />

      <br></br>

      <br></br>

      <input

        type="password"

        placeholder="Password"

        value={password}

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

      />

      <br></br>

      <br></br>

      <button className='btn btn-success' onClick={handleLogin}>Login</button>

    </div>

    </div>

  );

};

export default LoginForm;

SignupForm.js

import React, { useState } from 'react';

const SignupForm = ({ onSignup }) => {

  const [username, setUsername] = useState('');

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

  const handleSignup = () => {

    //Here we can rerform our signup logic (e.g., send a request to a server)

    // Here we'll assume a successful signup

    alert(`${username} signed in successfully..!`)

    onSignup(username);

  };

  return (

    <div className='container1'>

<div class="btn-box1">

            <div id="btn1">

            </div>

            <button type="button" class="toggle-btn">Login</button>

            <button type="button" class="toggle-btn">Signup</button>

      </div>

      <input

        type="text"

        placeholder="Username"

        value={username}

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

      />

      <br></br>

      <br></br>

      <input

        type="password"

        placeholder="Password"

        value={password}

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

      />

      <br></br>

      <br></br>

      <button className='btn btn-warning' onClick={handleSignup}>Signup</button>

    </div>

  );

};

export default SignupForm;

Dashboard.js

import React from 'react';

const Dashboard = ({ username, loggedInUsers }) => {

  return (

    <div className='App'>

      <h2 text-align='center' color='green'>Welcome, {username}!</h2>

      <br></br>

      <br></br>

      <h3>Logged-in Users:</h3>

      <ul>

        {loggedInUsers.map((user) => (

          <li key={user}>{user}</li>

        ))}

      </ul>

    </div>

  );

};

export default Dashboard;

App.js

import React, { useState } from 'react';

import LoginForm from './LoginForm';

import SignupForm from './SignupForm';

import Dashboard from './Dashboard';

function App() {

  const [loggedIn, setLoggedIn] = useState(false);

  const [loggedInUsers, setLoggedInUsers] = useState([]);

  const [currentUser, setCurrentUser] = useState(null);

  const handleLogin = (username) => {

    setCurrentUser(username);

    setLoggedIn(true);

    setLoggedInUsers((prevUsers) => [...prevUsers, username]);

  };

  const handleSignup = (username) => {

    setCurrentUser(username);

    setLoggedIn(true);

    setLoggedInUsers((prevUsers) => [...prevUsers, username]);

  };

  const handleLogout = () => {

    setCurrentUser(null);

    setLoggedIn(false);

    setLoggedInUsers((prevUsers) => prevUsers.filter((user) => user !== currentUser));

    alert(`Logged out succefully..!!`)

  };

  return (

    <div className='container2'>

            {loggedIn ? (

        <Dashboard username={currentUser} loggedInUsers={loggedInUsers} />

      )  : (

        <div>

          <LoginForm onLogin={handleLogin} />

          <SignupForm onSignup={handleSignup} />

        </div>

      )}

      {loggedIn && <button className='btn btn-danger' onClick={handleLogout}>Logout</button>}

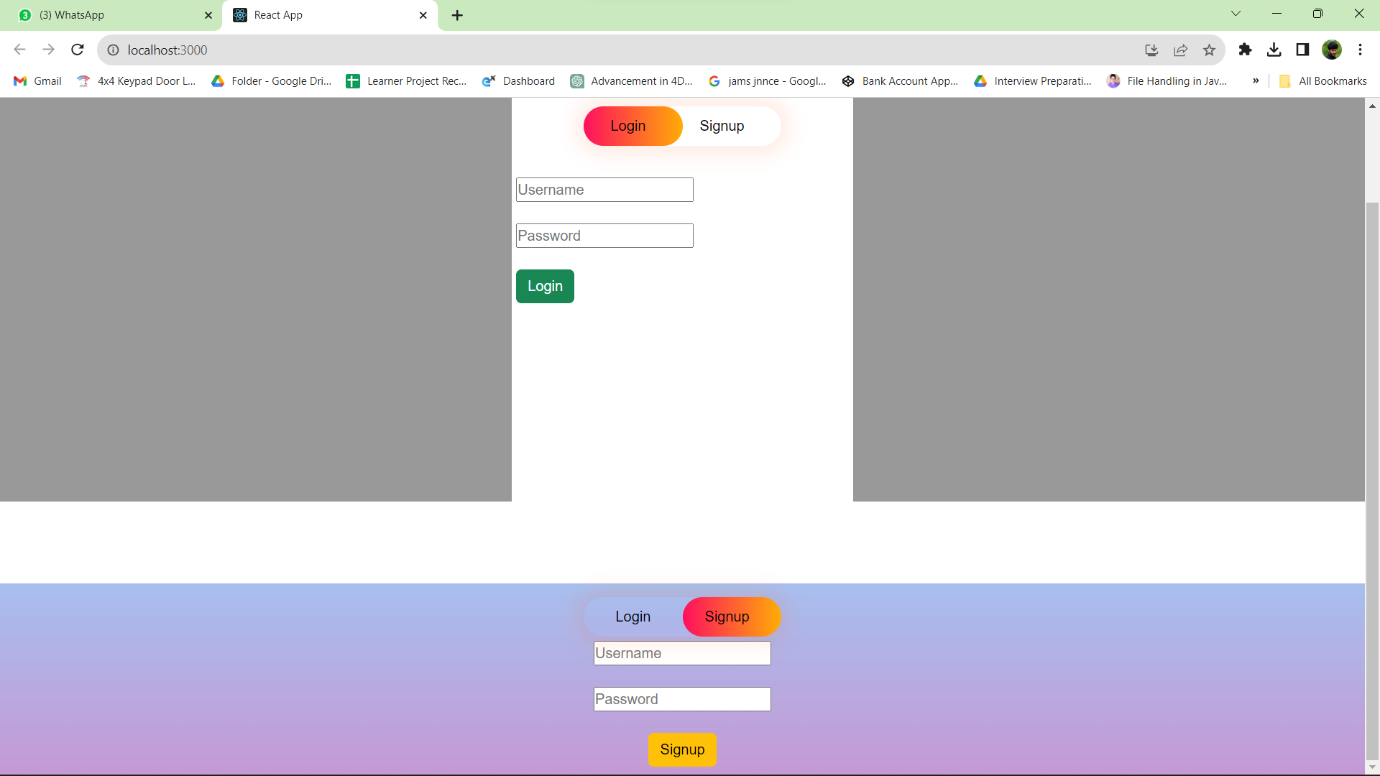
</div>

  );

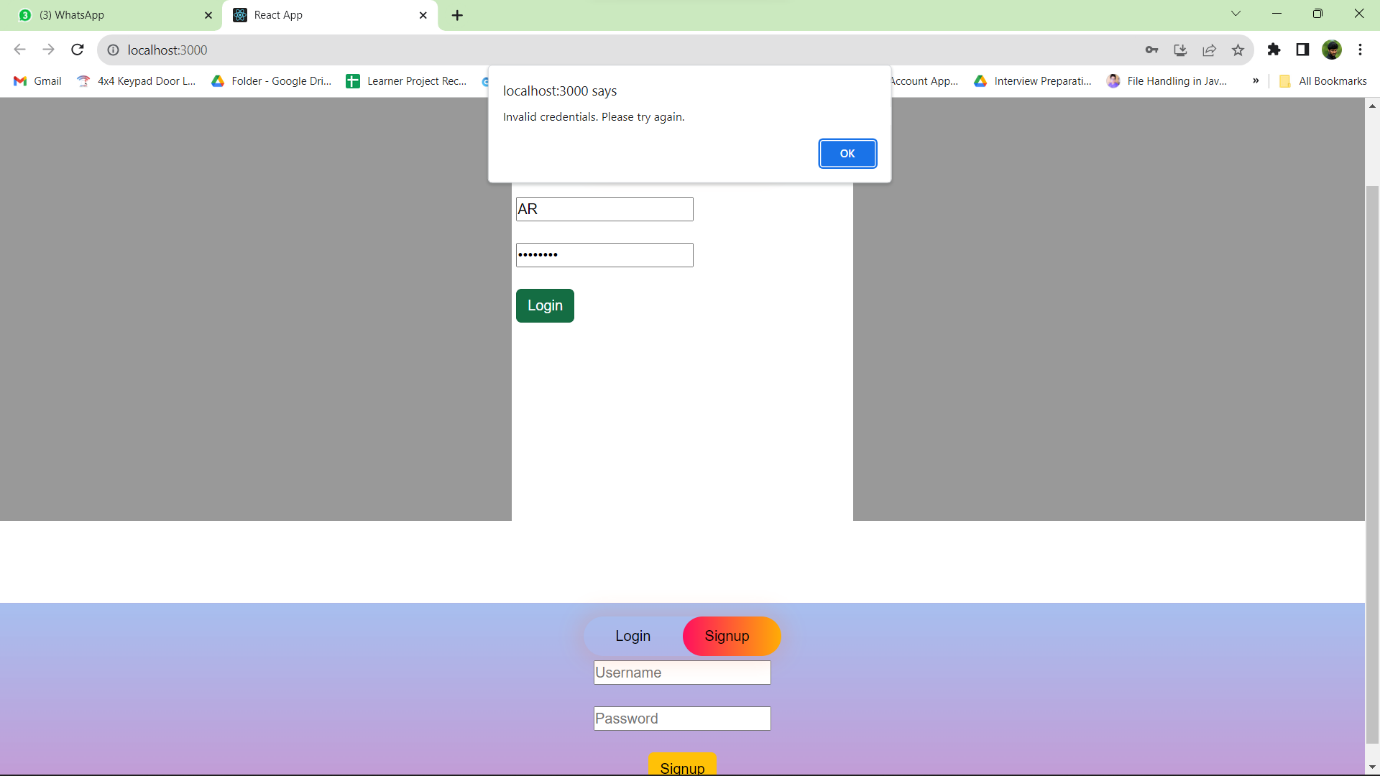
}

export default App;

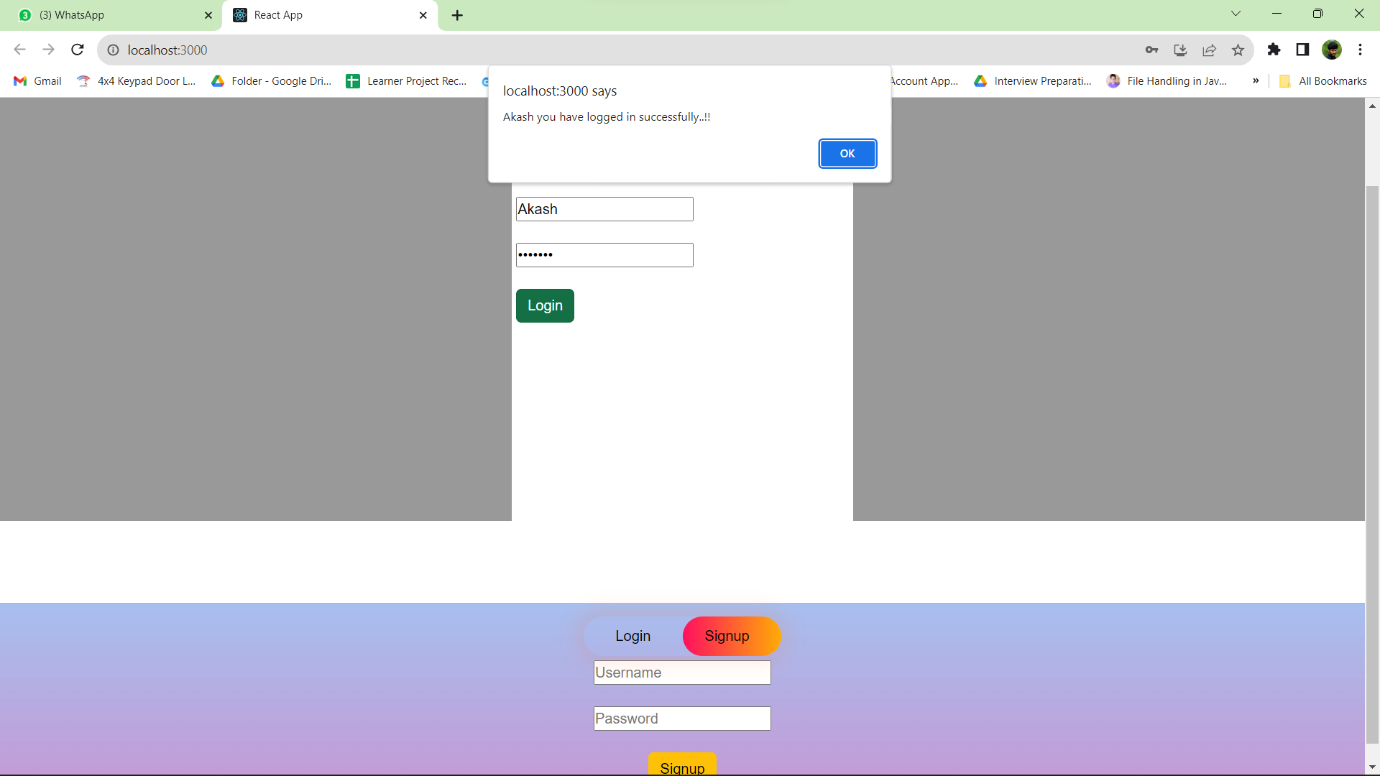
OUTPUT :



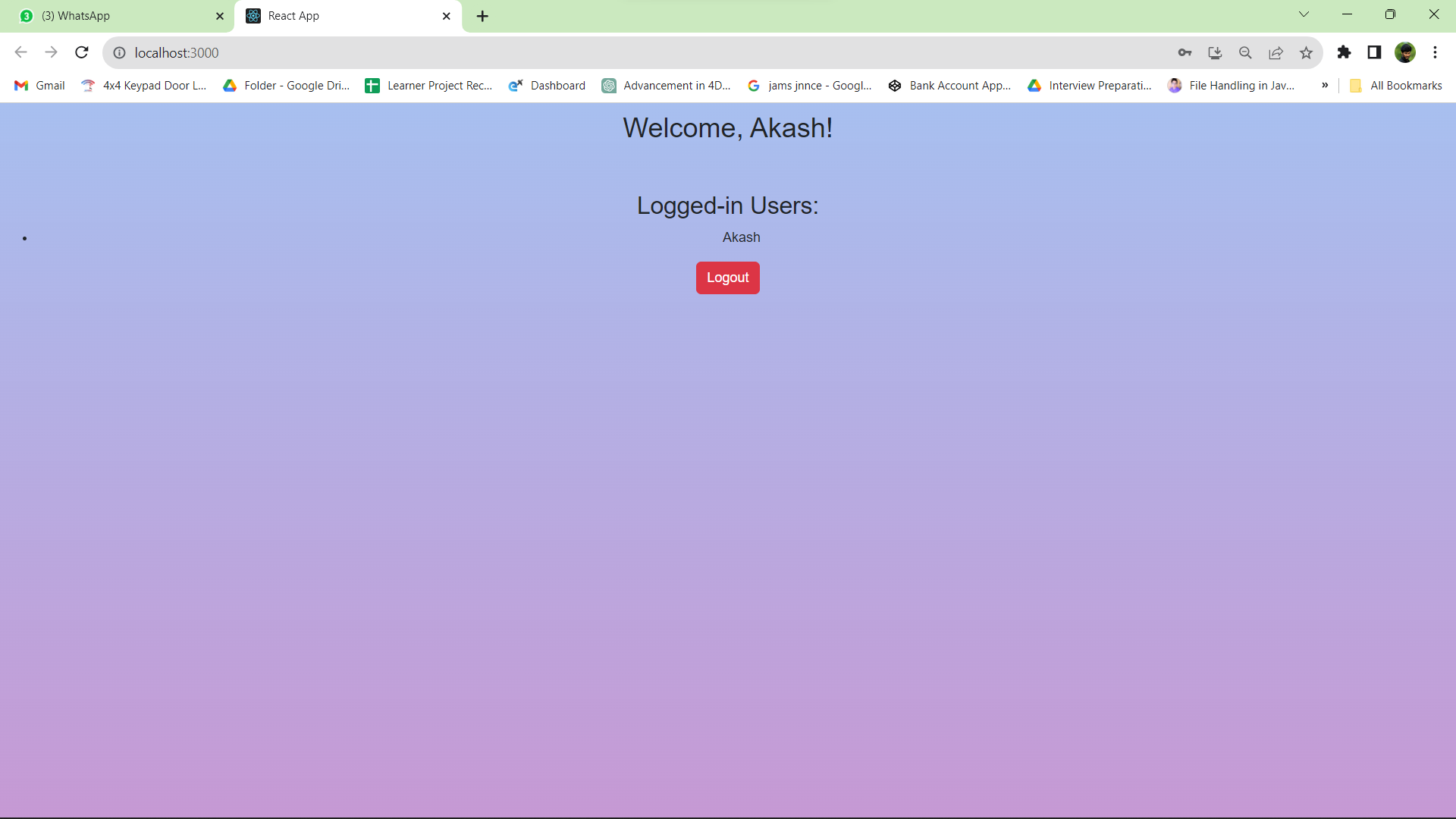
Front view



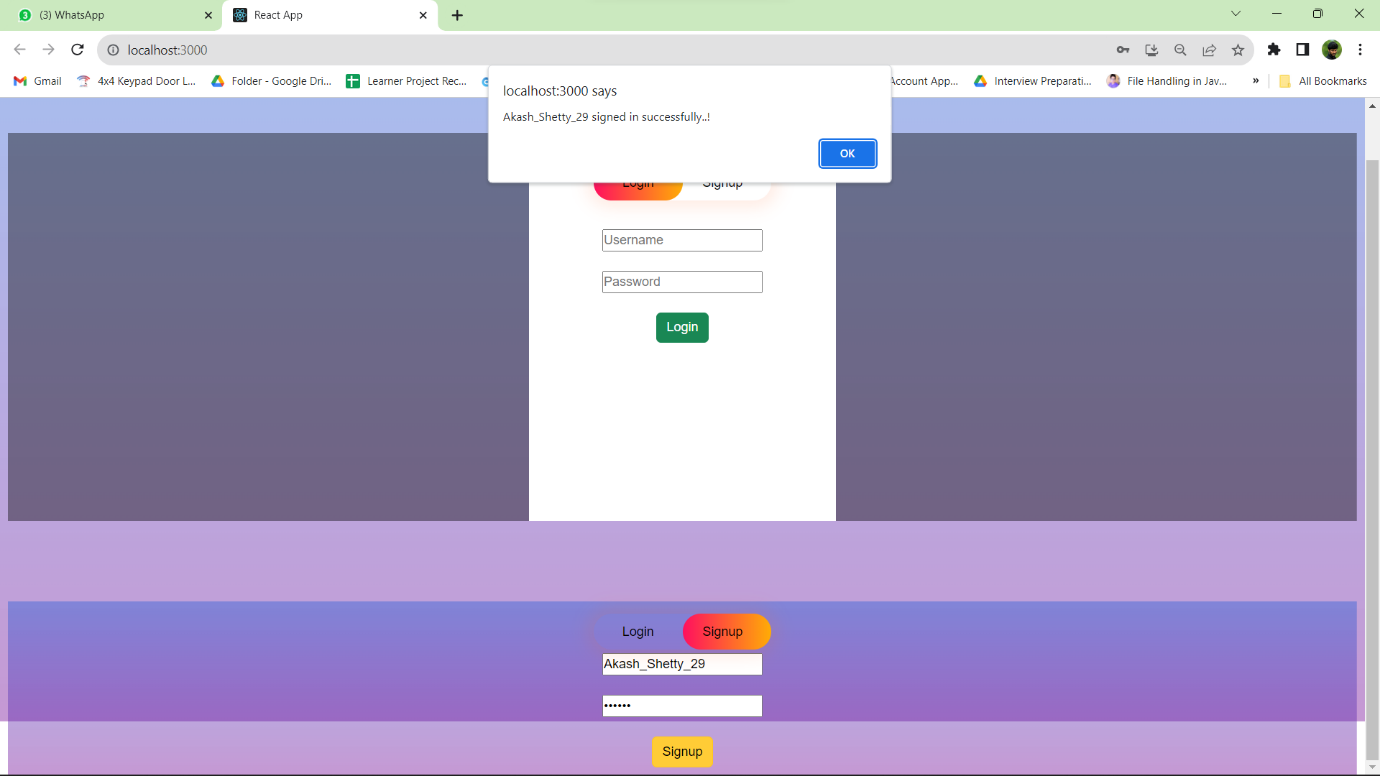
Denied entry of Invalid user



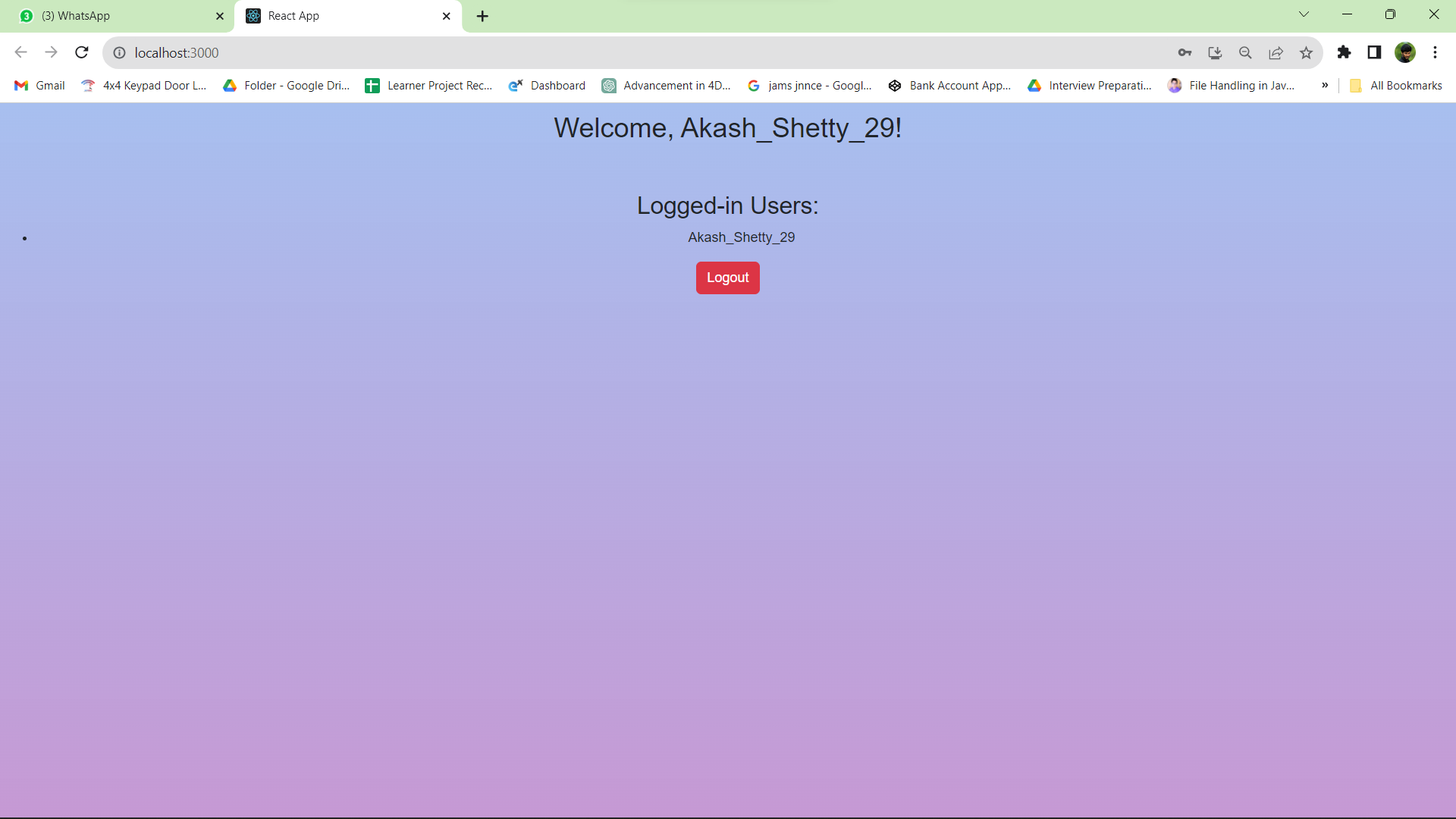
Successful Login



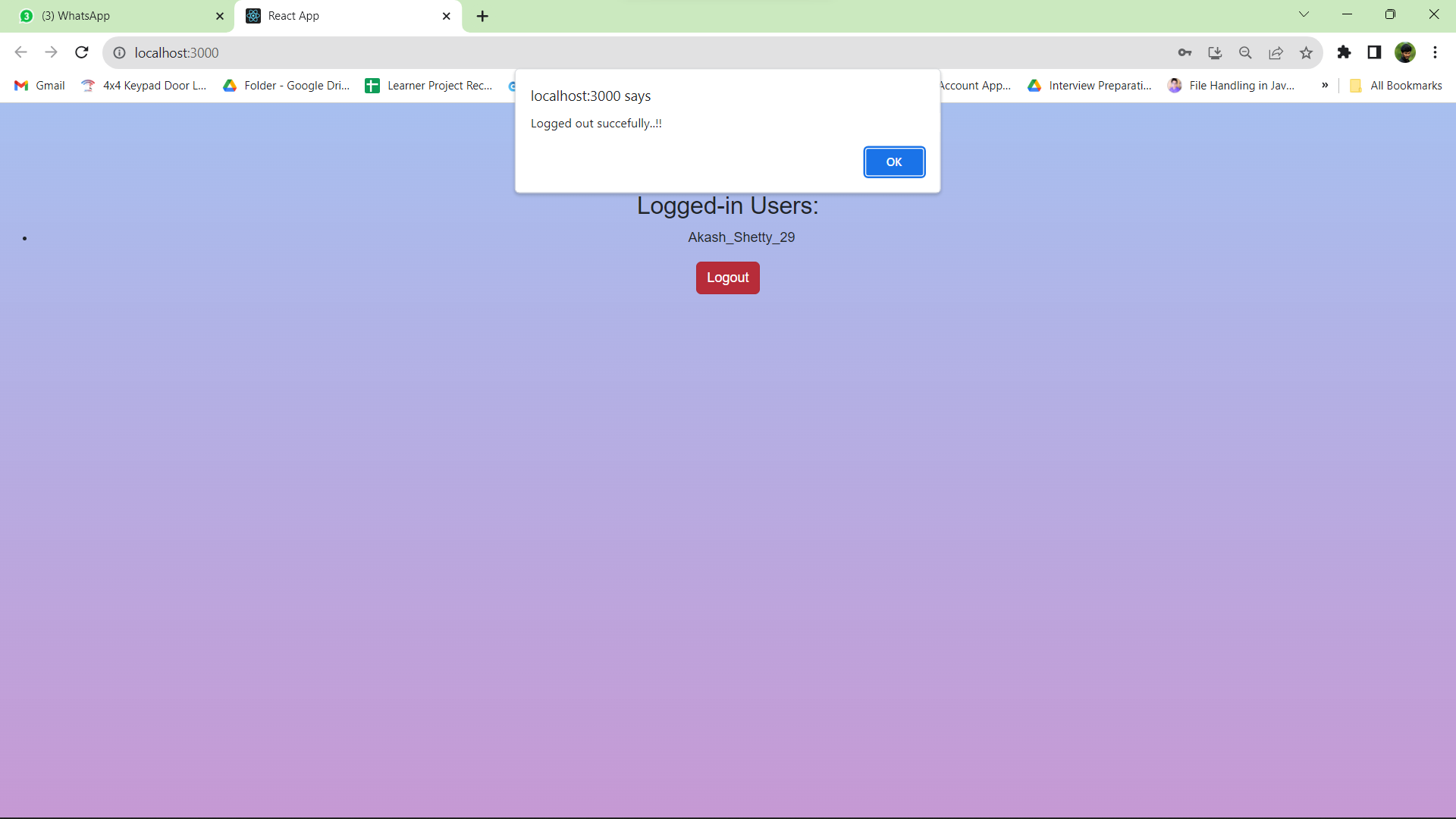
Welcome Screen and user name



Successful signup



Welcome screen and user name



Logging Out Successful

1. Write a program to execute react life cycle hooks?

LifecycleDemo.js

import React, { Component } from "react";

class LifecycleDemo extends Component {

  constructor(props) {

    super(props);

    this.state = {

      count: 0,

    };

    console.log("Constructor called");

  }

  componentDidMount() {

    console.log("Component Did Mount");

  }

  componentDidUpdate(prevProps, prevState) {

    console.log("Component Did Update");

  }

  componentWillUnmount() {

    console.log("Component Will Unmount");

  }

  incrementCount = () => {

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

  };

  render() {

    console.log("Render method called");

    return (

      <div>

        <h2>React Lifecycle Demo</h2>

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

        <button onClick={this.incrementCount}>Increment Count</button>

      </div>

    );

  }

}

export default LifecycleDemo;

App.js

import React from "react";

import LifecycleDemo from "./LifecycleDemo";

function App() {

  return (

    <div className="App">

      <LifecycleDemo />

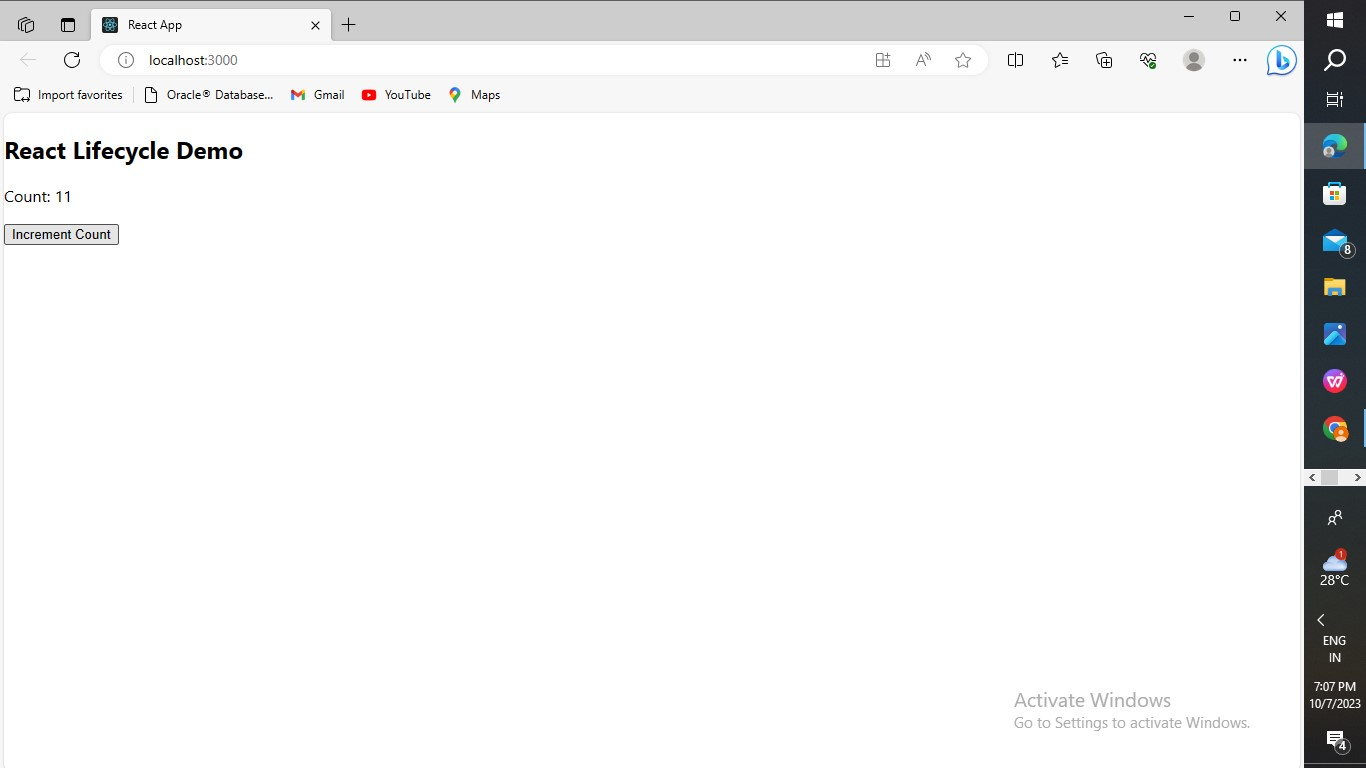
    </div>

  );

}

export default App;

output



1. Write a program to exchange product information from parent to child components in react?

App.js

import React, { Component } from 'react';

import './App.css';

import ProductList from './ProductList'; // Import the ProductList component

class App extends Component {

  constructor(props) {

    super(props);

    this.state = {

      products: [

        { id: 1, name: 'wheat', price: 40 },

        { id: 2, name: 'Chicken', price: 100},

        { id: 3, name: 'cookies', price: 50 },

        { id: 4, name: 'rice', price: 200},

        { id: 5, name: 'oats', price: 90},

      ],

    };

  }

  render() {

    return (

      <div className="App">

        <h1>Food Product List</h1>

        <ProductList products={this.state.products} />

      </div>

    );

  }

}

export default App;

Product.js

import React from 'react';

const Product = (props) => {

  const { name, price } = props.product;

  return (

    <li>

      <h3>{name}</h3>

      <p>Price: ${price}</p>

    </li>

  );

};

export default Product;

ProductList.js

import React from 'react';

import Product from './Product'; // Import the Product component

const ProductList = (props) => {

  const { products } = props;

  return (

    <div>

      <h2>Product List</h2>

      <ul>

        {products.map((product) => (

          <Product key={product.id} product={product} />

        ))}

      </ul>

    </div>

  );

};

export default ProductList;

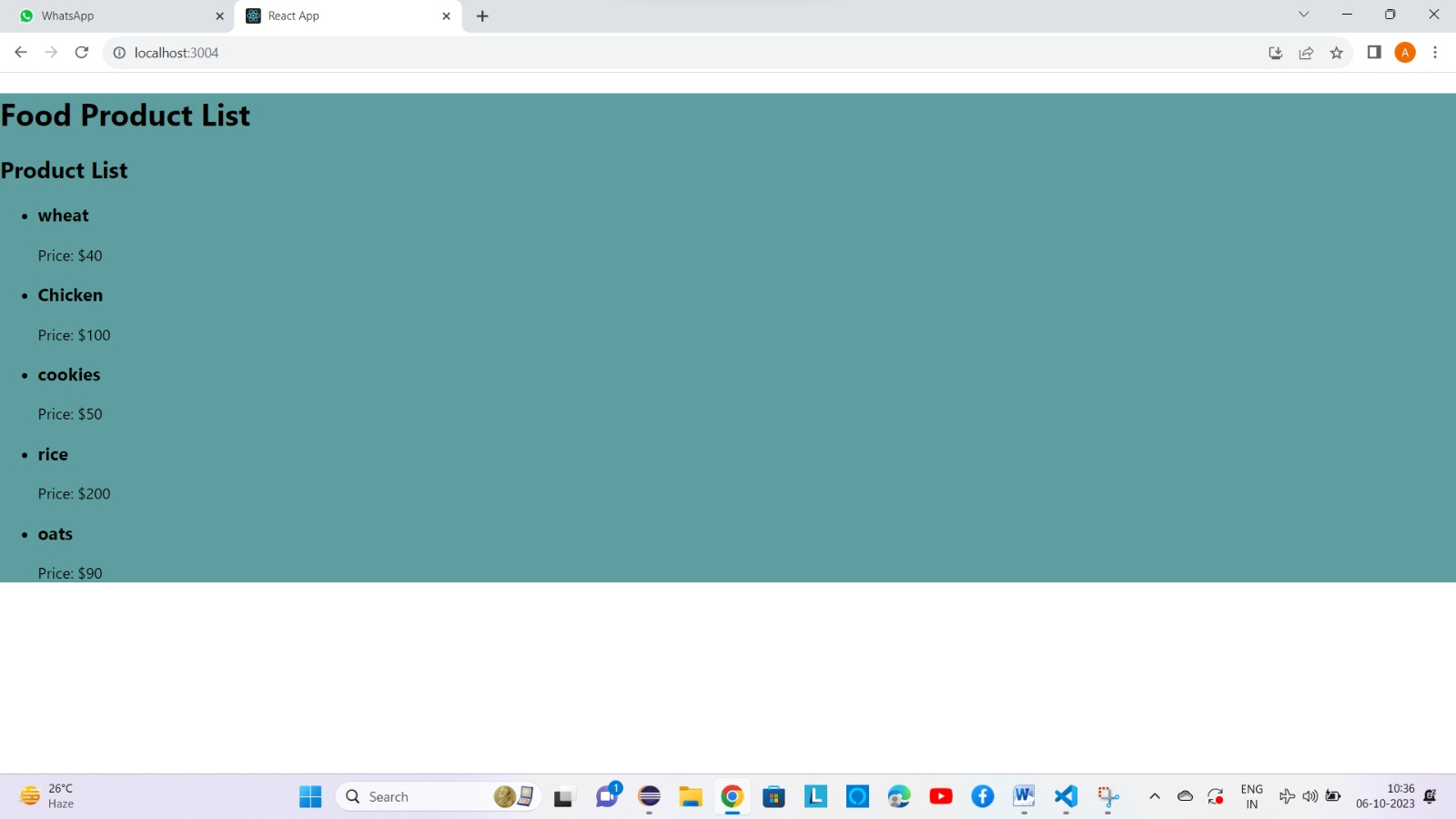
App.css

.App {

  text-align: left;

  background-color: cadetblue;

}

Output

5) Write a program to do post request in Axios and Fetch methods?

Index.html

<!DOCTYPE html>

<html>

<head>

    <title>POST Request Example</title>

    <style>

        body{

            background-color: lightgrey;

        }

        h1{

            color: brown;

        }

    </style>

</head>

<body>

    <h1>POST Request Example</h1>

    <button id="axiosButton">Make POST Request with Axios</button>

    <button id="fetchButton">Make POST Request with Fetch</button>

    <pre id="response"></pre>

    <script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>

    <script src="script.js"></script>

</body>

</html>

Script.html

// Function to make a POST request using Axios

document.getElementById('axiosButton').addEventListener('click', function () {

    const postData = {

        key1: 'Akash B shetty',

        key2: '8217256357'

    };

    axios.post('https://jsonplaceholder.typicode.com/posts', postData)

        .then(response => {

            displayResponse(response.data);

        })

        .catch(error => {

            displayError('Axios Error: ' + error);

        });

});

// Function to make a POST request using Fetch

document.getElementById('fetchButton').addEventListener('click', function () {

    const postData = {

        key1: 'akash.shetty.ec@gmail.com',

        key2: 'Karnataka'

    };

    fetch('https://jsonplaceholder.typicode.com/posts', {

        method: 'POST',

        headers: {

            'Content-Type': 'application/json'

        },

        body: JSON.stringify(postData)

    })

        .then(response => {

            if (response.ok) {

                return response.json();

            } else {

                throw new Error('Fetch Error: ' + response.status);

            }

        })

        .then(data => {

            displayResponse(data);

        })

        .catch(error => {

            displayError('Fetch Error: ' + error);

        });

});

// Function to display the response or error

function displayResponse(data) {

    document.getElementById('response').textContent = JSON.stringify(data, null, 2);

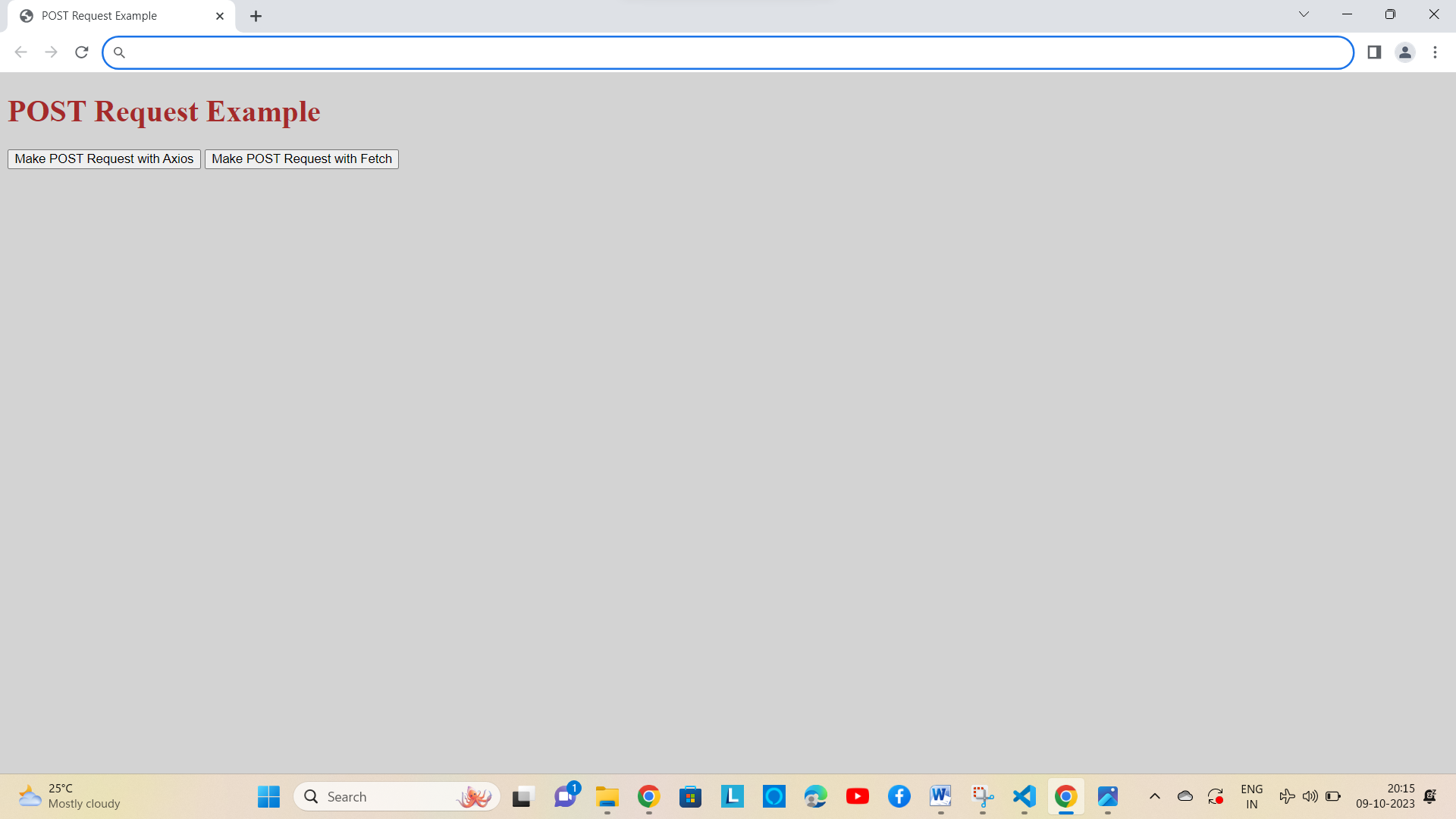
}

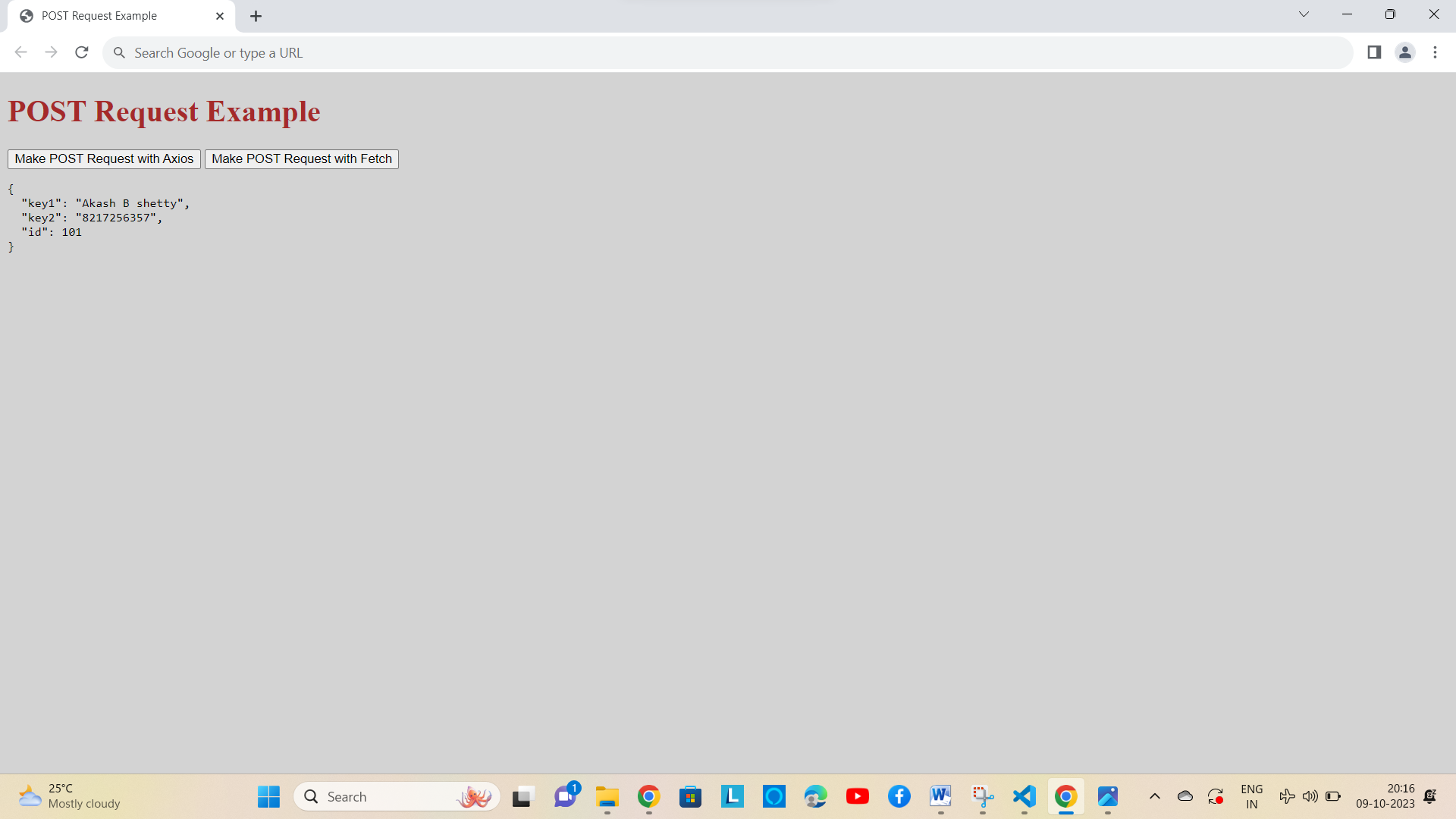
function displayError(errorMessage) {

    document.getElementById('response').textContent = errorMessage;

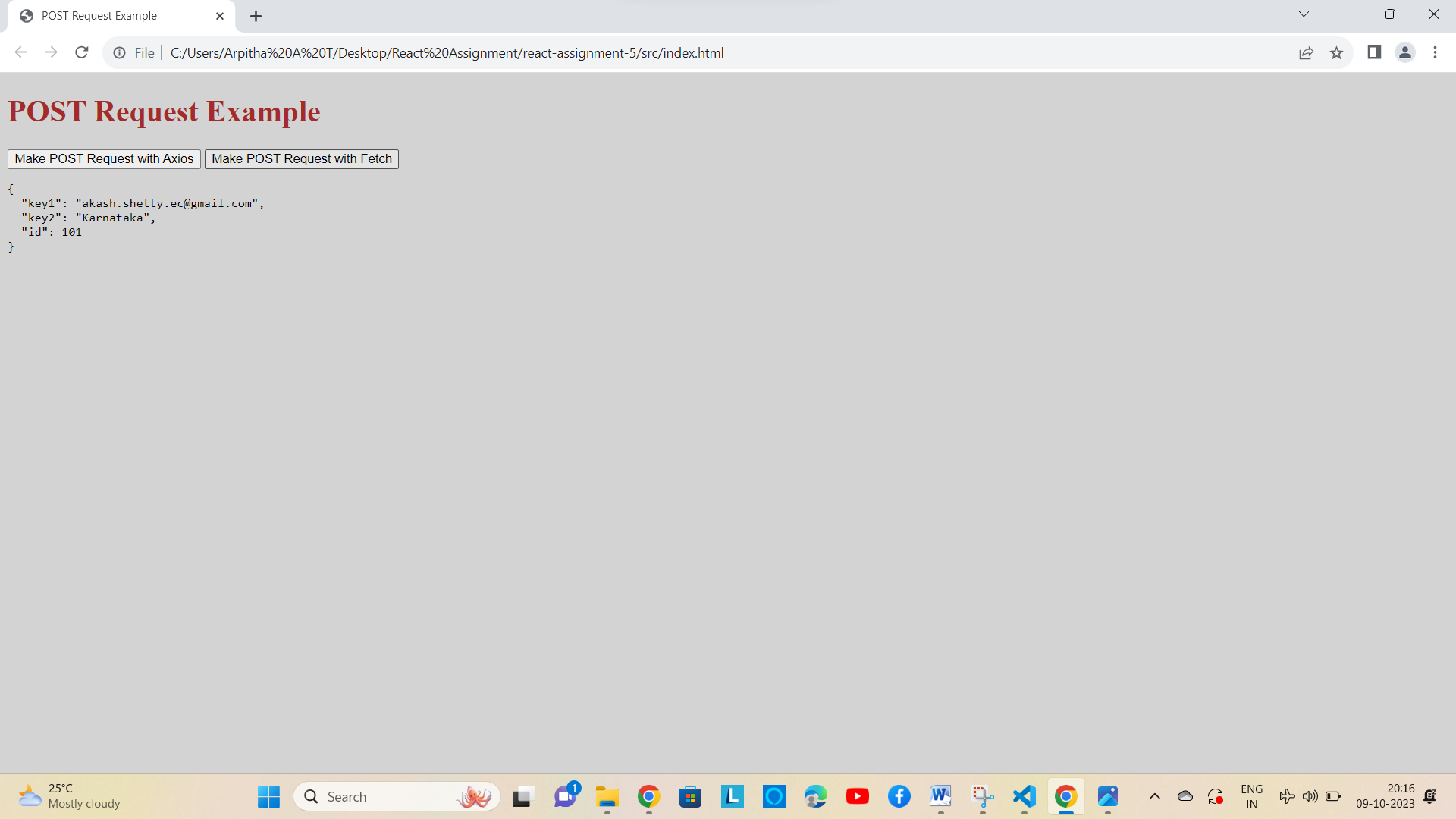
}

Output





Request with Axios



Request with Fetch