**WEEK 7**

***REACT***

**1.** 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.

1.Create react app using vite:

npm create vite@latest my-react-app

2. Create Components folder in src directory and create

* BlogDetails.jsx
* BookDetails.jsx
* CourseDetails.jsx

BlogDetails.jsx:

import React from 'react';

const blog = [{

  title: 'React Learning',

  author: 'Stephen Biz',

  content: "Wekonp to learning React.",

},{

    title: 'Installation',

    author: 'Schewzdenier',

    content: 'You can install React using npm.',

}

];

export default function BlogDetails({ show }) {

  return show ? ( // Conditional rendering using ternary operator

    <div>

      <h1>Blog Details</h1>

        {blog.map((blog) =>

        <div key={blog.title}>

      <h2>{blog.title}</h2>

      <h3>{blog.author}</h3>

      <p>{blog.content}</p>

      </div>

        )}

    </div>

  ) : null;

}

BookDetails.jsx:

import React from "react";

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 default function BookDetails({ show }) {

  if (!show) return null; // Conditional rendering using `if`

  const bookdet = books.map((book) => (

    <div key={book.id}>

      <h3>{book.bname}</h3>

      <h4>{book.price}</h4>

    </div>

  ));

  return (

    <div className="st2">

      <h1>Book Details</h1>

      {bookdet}

    </div>

  );

}

CourseDetails.jsx:

import React from 'react';

const courses = [

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

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

];

export default function CourseDetails({ show }) {

  return (

    <div>

      {show && ( // Conditional rendering using logical AND

        <>

          <h1>Course Details</h1>

          {courses.map((course, idx) => (

            <div key={idx}>

              <h3>{course.name}</h3>

              <p>{course.date}</p>

            </div>

          ))}

        </>

      )}

    </div>

  );

}

App.jsx:

import { useState } from 'react'

import reactLogo from './assets/react.svg'

import viteLogo from '/vite.svg'

import './App.css'

import CourseDetails from './components/CourseDetails'

import BookDetails from './components/BookDetails'

import BlogDetails from './components/BlogDetails'

function App() {

  const [count, setCount] = useState(0)

  return (

    <>

    <div className="App">

      <CourseDetails show={true} />

      <hr style={{width:"5px",backgroundColor:"green",border:"none"}}/>

      <BookDetails show={true} />

      <hr style={{width:"5px",backgroundColor:"green",border:"none"}}/>

      <BlogDetails show={true} />

    </div>

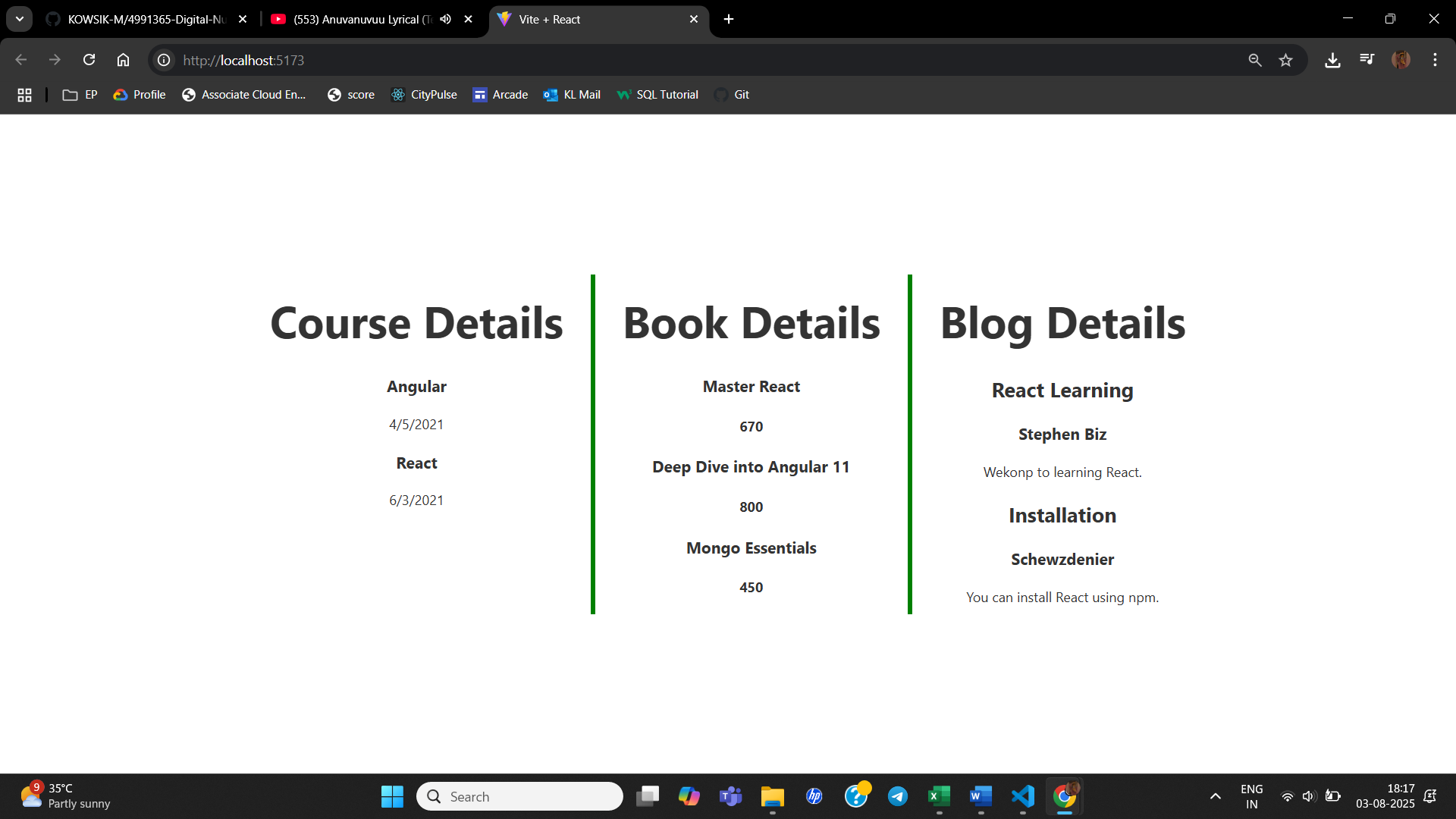
    </>

  )

}

export default App

Output:



2. Developers of Apps Centric Solutions have created an employee management application which supports light and dark themes for the buttons. The current solution uses the react state and props to provide the theme name to be used from App component to Employee List component and from there to Employee Card component. Quality assurance team analyzed the solutions and found the technique being used to be a substandard one. React architect suggested to use the react context API to share the theme name with nested child components instead of passing them down using props from the parent component.

1. Unzip the employeesapp.zip and open it using VS Code
2. Go to terminal and execute *npm install* command to restore all the node modules
3. Create ThemeContext.js

import { createContext } from 'react';

const ThemeContext = createContext('light'); // default value

export default ThemeContext;

1. Modify EmployeeCard.js

// EmployeeCard.js

import React, { useContext } from 'react';

import ThemeContext from './ThemeContext';

import Styles from './EmployeeCard.module.css';

function EmployeeCard(props) {

  const theme = useContext(ThemeContext); // ✅ Get theme from context

  return (

    <div className={Styles.Card}>

      <h3>{props.employee.name}</h3>

      <p>{props.employee.email}</p>

      <p>{props.employee.phone}</p>

      <p>

        <a href="#" className={theme}>Edit</a>

        <a href="#" className={theme}>Delete</a>

      </p>

    </div>

  );

}

export default EmployeeCard;

1. Modify EmployeesList.js

// EmployeesList.js

import EmployeeCard from './EmployeeCard';

function EmployeesList(props) {

  return (

    <div>

      <h1>Employees List</h1>

      {props.employees.map(employee => (

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

      ))}

    </div>

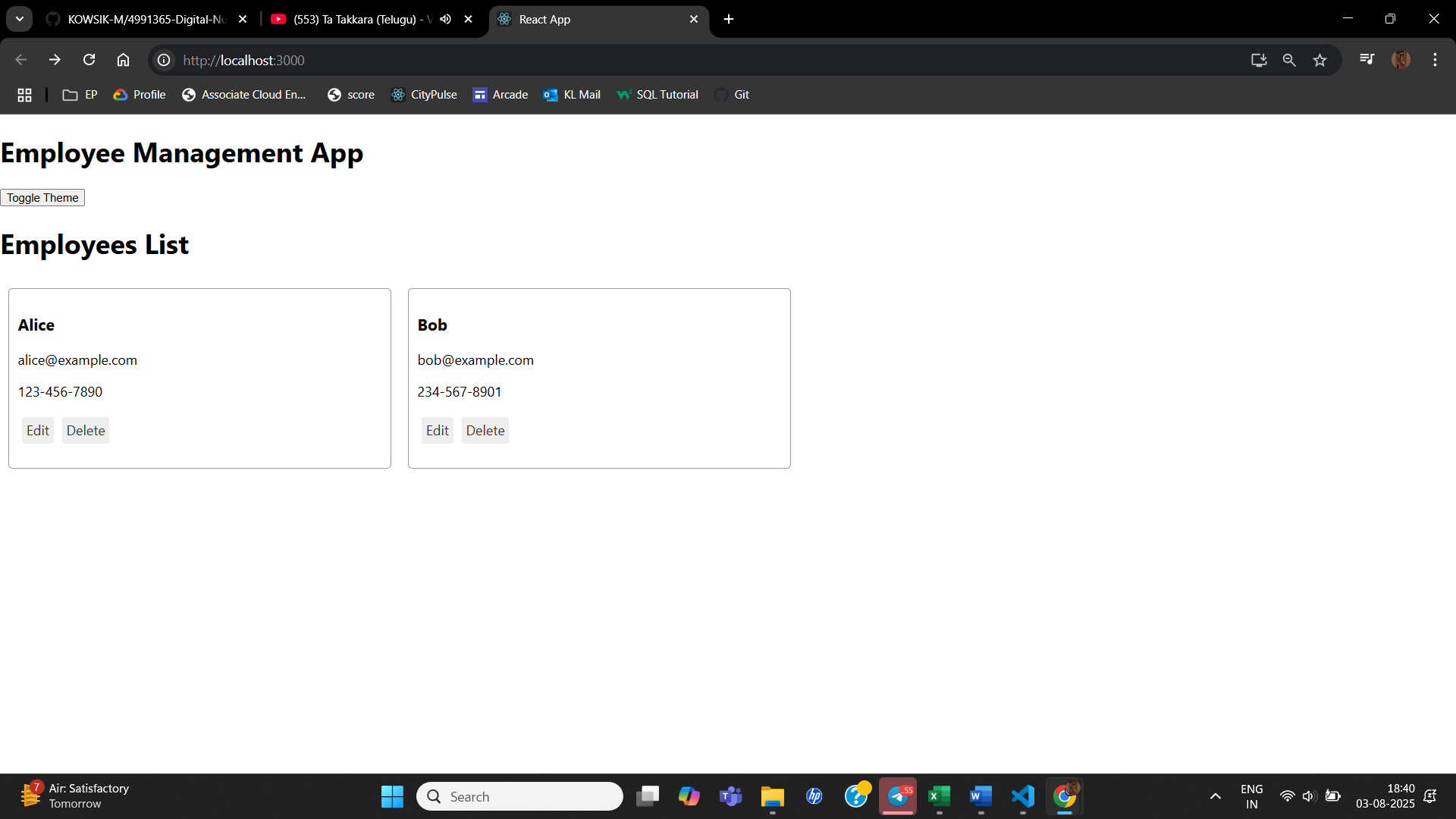
  );

}

export default EmployeesList;

Output:

Light mode:



Dark mode:

A screenshot of a computer

AI-generated content may be incorrect.

3. 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.

1. Create the App:

npx create-react-app ticketraisingapp

cd ticketraisingapp

1. Create ComplaintRegister.js in src

// src/ComplaintRegister.js

import React, { Component } from 'react';

class ComplaintRegister extends Component {

  constructor(props) {

    super(props);

    this.state = {

      ename: '',

      complaint: '',

      NumberHolder: Math.floor(Math.random() \* 100) + 1, // Generates random ID

    };

  }

  handleChange = (event) => {

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

  };

  handleSubmit = (event) => {

    const msg = `Thanks ${this.state.ename}

Your Complaint was Submitted.

Transaction ID is: ${this.state.NumberHolder}`;

    alert(msg);

    event.preventDefault();

  };

  render() {

    return (

      <div style={{ textAlign: 'center', marginTop: '40px' }}>

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

        <form onSubmit={this.handleSubmit}>

          <div>

            <label>Name: </label>

            <input

              type="text"

              name="ename"

              value={this.state.ename}

              onChange={this.handleChange}

              required

            />

          </div>

          <br />

          <div>

            <label>Complaint: </label>

            <textarea

              name="complaint"

              value={this.state.complaint}

              onChange={this.handleChange}

              required

            />

          </div>

          <br />

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

        </form>

      </div>

    );

  }

}

export default ComplaintRegister;

1. Modify App.js:

// src/App.js

import React from 'react';

import ComplaintRegister from './ComplaintRegister';

function App() {

  return (

    <div className="App">

      <ComplaintRegister />

    </div>

  );

}

export default App;

Output:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

4. 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.

1. Create react app:

npx create-react-app@latest mailregisterapp

cd mailregisterapp

1. Create Register.js:

import React, { Component } from 'react';

class Register extends Component {

  constructor(props) {

    super(props);

    this.state = {

      name: '',

      email: '',

      password: '',

    };

  }

  handleChange = (event) => {

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

  };

  validate = () => {

    const { name, email, password } = this.state;

    if (name.length < 5) {

      alert('Full Name must be 5 characters long!');

      return false;

    }

    const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

    if (!emailRegex.test(email)) {

      alert('Email is not valid!');

      return false;

    }

    if (password.length < 8) {

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

      return false;

    }

    return true;

  };

  handleSubmit = (event) => {

    event.preventDefault();

    if (this.validate()) {

      alert(`Thanks ${this.state.name}, you have been registered successfully!`);

    }

  };

  render() {

    return (

      <div style={{ textAlign: 'center', marginTop: '40px' }}>

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

        <form onSubmit={this.handleSubmit}>

          <div>

            <label>Name: </label>

            <input

              type="text"

              name="name"

              value={this.state.name}

              onChange={this.handleChange}

              required

            />

          </div>

          <br />

          <div>

            <label>Email: </label>

            <input

              type="email"

              name="email"

              value={this.state.email}

              onChange={this.handleChange}

              required

            />

          </div>

          <br />

          <div>

            <label>Password: </label>

            <input

              type="password"

              name="password"

              value={this.state.password}

              onChange={this.handleChange}

              required

            />

          </div>

          <br />

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

        </form>

      </div>

    );

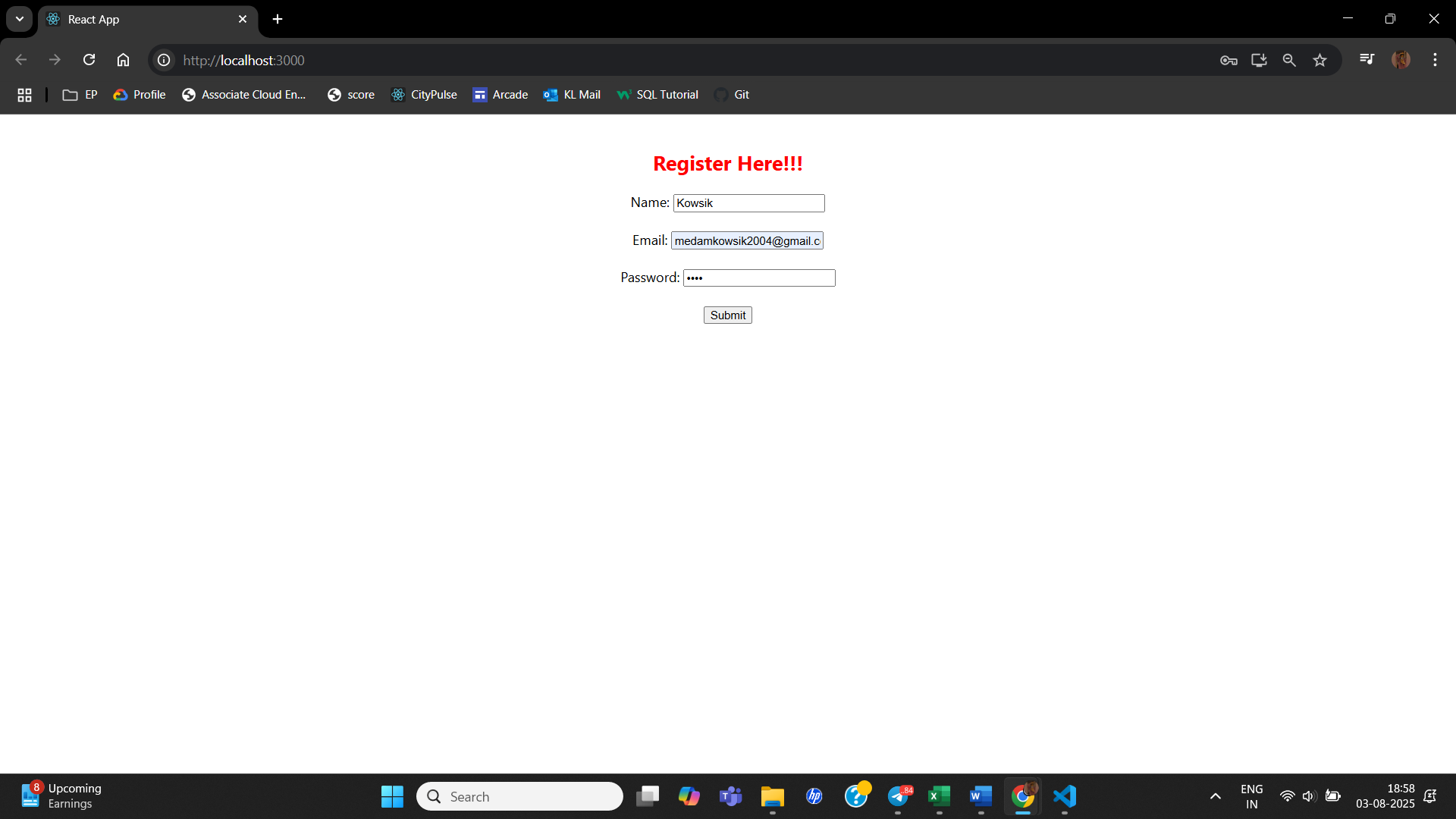
  }

}

export default Register;

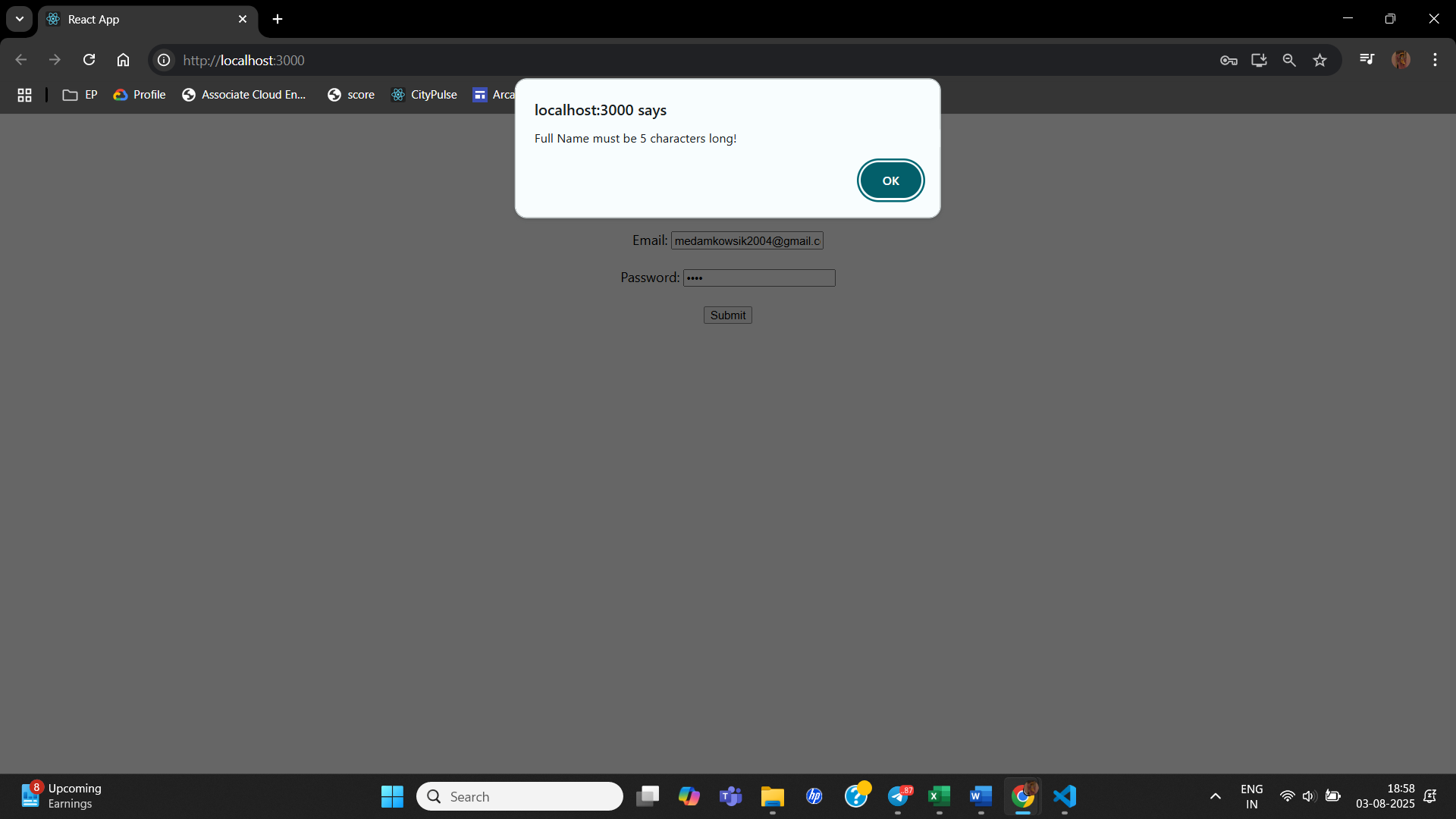
1. Modify App.js:
2. import React from 'react';
3. import Register from './Register';
4. function App() {
5. return (
6. <div className="App">
7. <Register />
8. </div>
9. );
10. }
11. export default App;

Output:



A screenshot of a computer

AI-generated content may be incorrect.

 A screenshot of a computer

AI-generated content may be incorrect.

5. 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.

Code Snippet in Getuser Component:

1. Create react app:

npx create-react-app@latest fetchuserapp

cd fetchuserapp

1. Create GetUser.js in src:

import React, { Component } from "react";

class GetUser extends Component {

  constructor(props) {

    super(props);

    this.state = {

      person: null,

      loading: true,

    };

  }

  async componentDidMount() {

    const url = "https://api.randomuser.me/";

    const response = await fetch(url);

    const data = await response.json();

    this.setState({

      person: data.results[0],

      loading: false,

    });

    console.log(data.results[0]);

  }

  render() {

    const { person, loading } = this.state;

    if (loading) {

      return <h2>Loading...</h2>;

    }

    return (

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

        <h1>{`${person.name.title} ${person.name.first} ${person.name.last}`}</h1>

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

      </div>

    );

  }

}

export default GetUser;

1. Modify App.js:
2. import React from 'react';
3. import GetUser from './GetUser';
4. function App() {
5. return (
6. <div className="App">
7. <GetUser />
8. </div>
9. );
10. }
11. export default App;

Output:

