**WEEK-6 REACT**

1. **ReactJS-HOL:**

**App.js**

import logo from './logo.svg';

import './App.css';

function App() {

  return (

    <div className="App">

      <header className="App-header">

        <h1>Welcome the first session of the React</h1>

      </header>

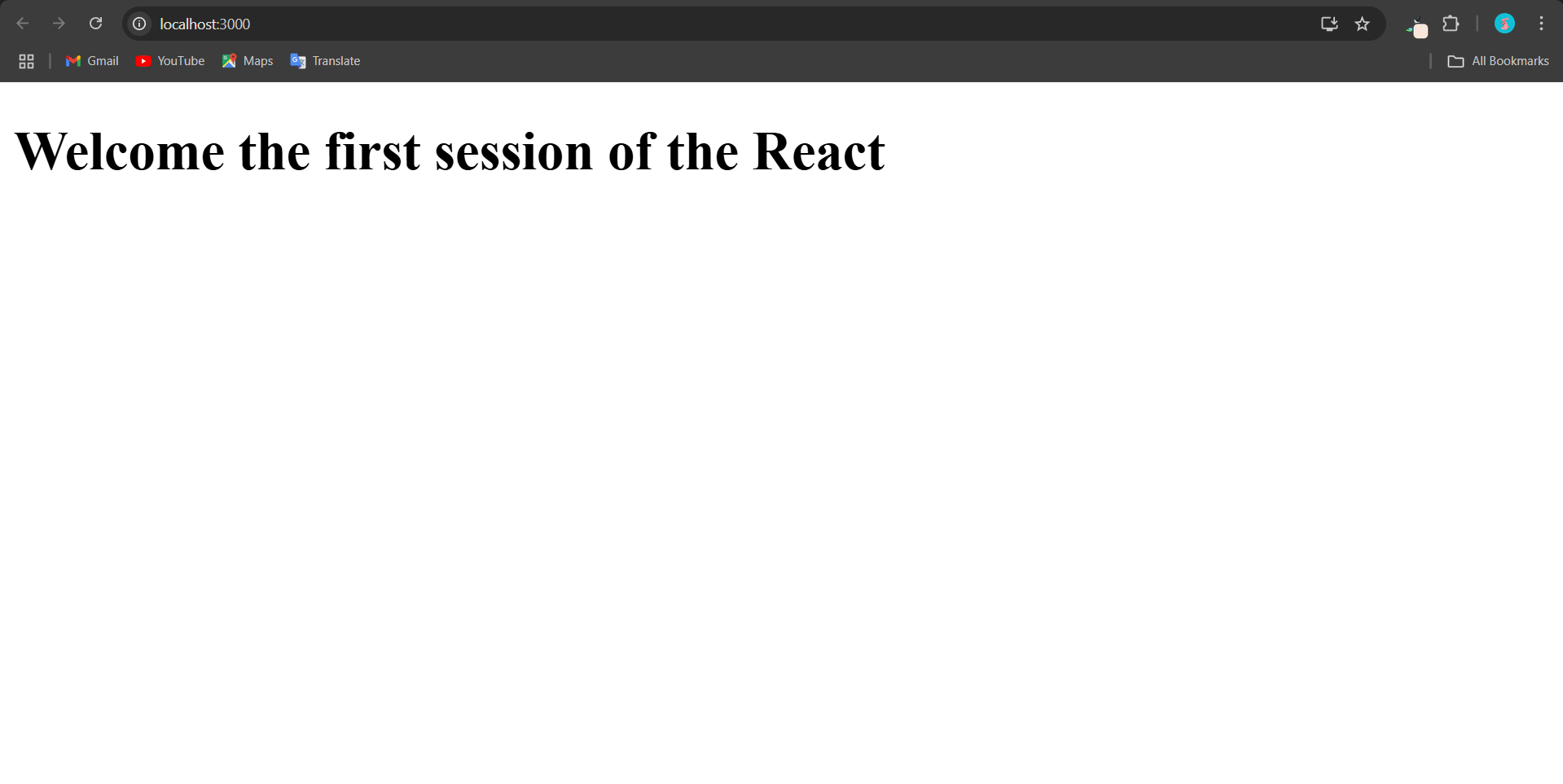
    </div>

  );

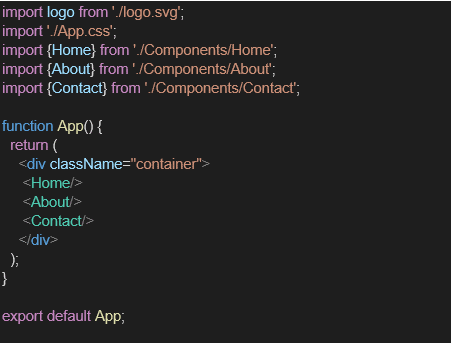
}

export default App;

**Output:**



1. **ReactJS-HOL**



import React, { Component } from 'react';

class About extends Component {

  render() {

    return (

      <div>

        <h3>Welcome to the Home Page of Student Management Portal</h3>

      </div>

    );

  }

}

export default About;

import React, { Component } from 'react';

class About extends Component {

  render() {

    return (

      <div>

        <h3>Welcome to the About Page of Student Management Portal</h3>

      </div>

    );

  }

}

export default About;

import React, { Component } from 'react';

class About extends Component {

  render() {

    return (

      <div>

        <h3>Welcome to the Contact Page of Student Management Portal</h3>

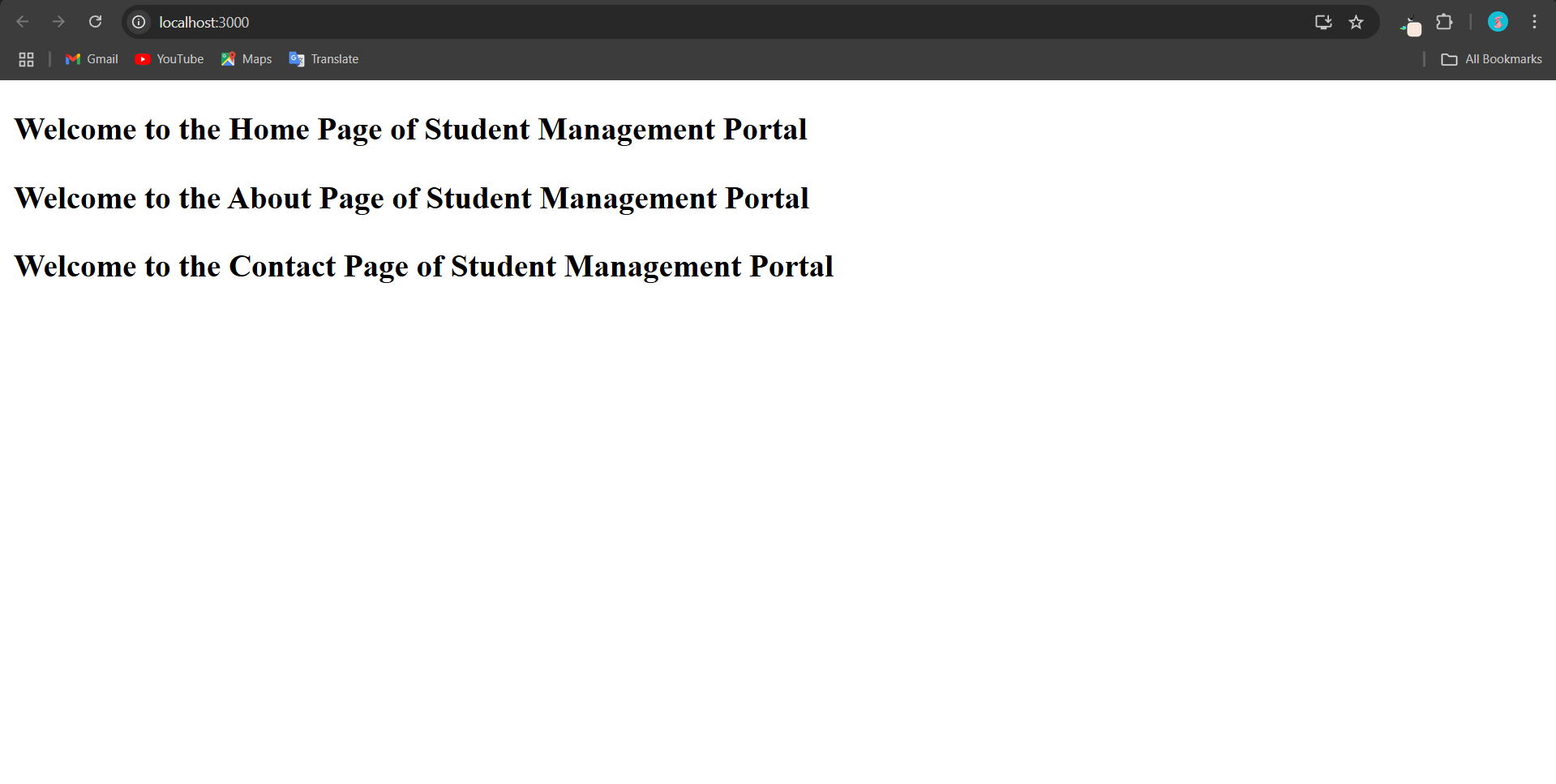
      </div>

    );

  }

}

export default About;

Output:

1. **ReactJS-HOL:**

**App.js:**

**App.css**

.container {

  text-align: center;

  margin-top: 50px;

  font-family: Arial, sans-serif;

}

.heading {

  color: brown;

  font-size: 2em;

}

.label-blue {

  color: blue;

}

.label-red {

  color: crimson;

}

.label-purple {

  color: purple;

}

.label-green {

  color: green;

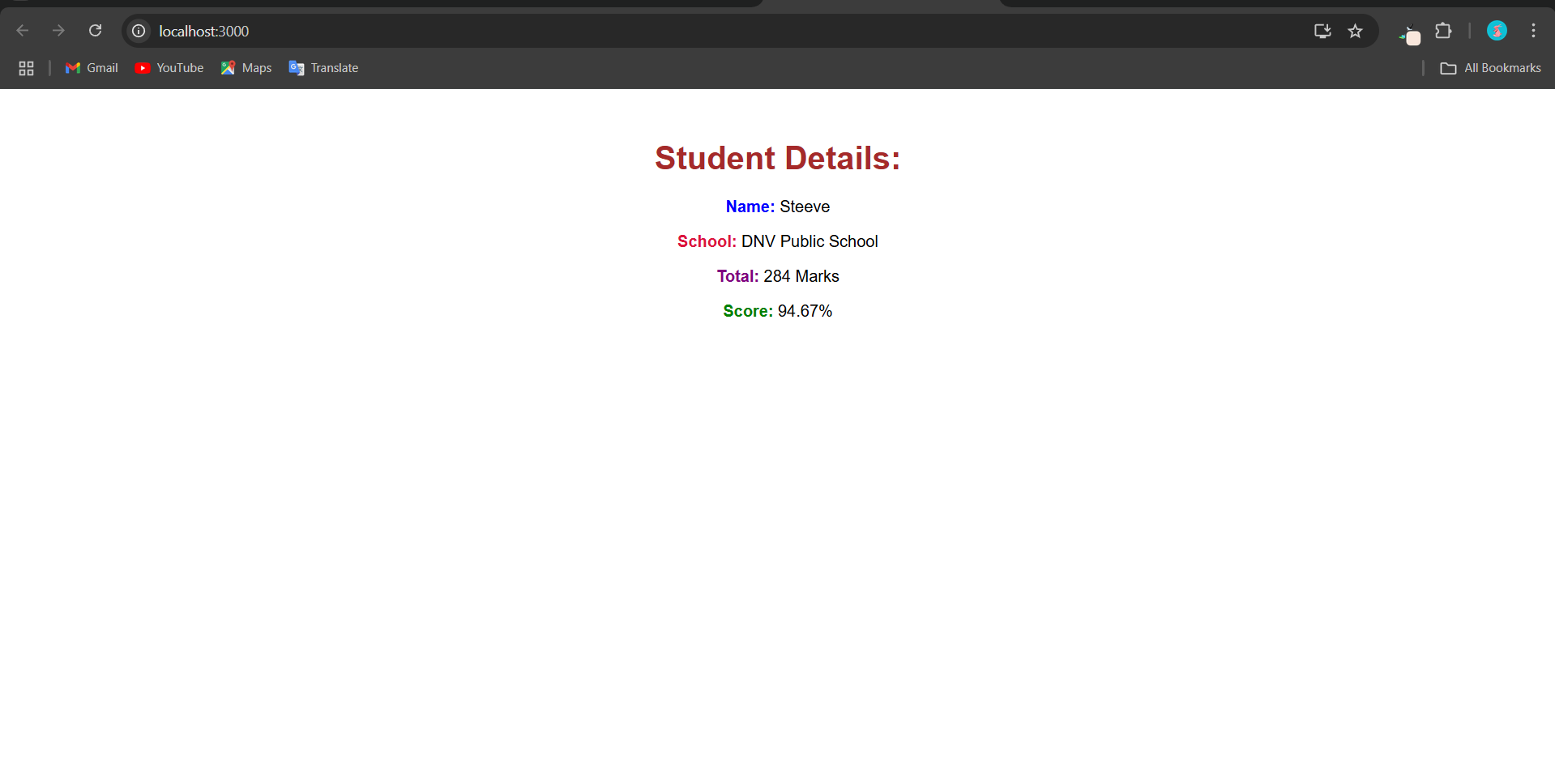
}

.value {

  color: #000;

}

**Output:**

****

1. **ReactJS-HOL**

**Post.js:**

// src/Post.js

class Post {

  constructor(userId, id, title, body) {

    this.userId = userId;

    this.id = id;

    this.title = title;

    this.body = body;

  }

}

export default Post;

// src/Posts.js

import React, { Component } from 'react';

import Post from './Post';

class Posts extends Component {

  constructor(props) {

    super(props);

    this.state = {

      posts: [],

      hasError: false,

    };

  }

  // Fetch posts from API and set state

  loadPosts = async () => {

    try {

      const response = await fetch('https://jsonplaceholder.typicode.com/posts');

      const data = await response.json();

      const postObjects = data.map(

        post => new Post(post.userId, post.id, post.title, post.body)

      );

      this.setState({ posts: postObjects });

    } catch (error) {

      this.setState({ hasError: true });

    }

  };

  // Lifecycle hook to trigger post loading

  componentDidMount() {

    this.loadPosts();

  }

  // Catch any rendering errors

  componentDidCatch(error, info) {

    alert("An error occurred while rendering posts.");

    console.error(error, info);

  }

  render() {

    const { posts, hasError } = this.state;

    if (hasError) {

      return <h2 style={{ color: "red" }}>Something went wrong!</h2>;

    }

    return (

      <div>

        <h1 style={{ textAlign: "center", color: "brown" }}>Blog Posts</h1>

        {posts.map(post => (

          <div key={post.id} style={{ marginBottom: "20px", padding: "10px", borderBottom: "1px solid #ccc" }}>

            <h3>{post.title}</h3>

            <p>{post.body}</p>

          </div>

        ))}

      </div>

    );

  }

}

export default Posts;

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

import React from 'react';

import Posts from './Posts';

function App() {

  return (

    <div className="App">

      <Posts />

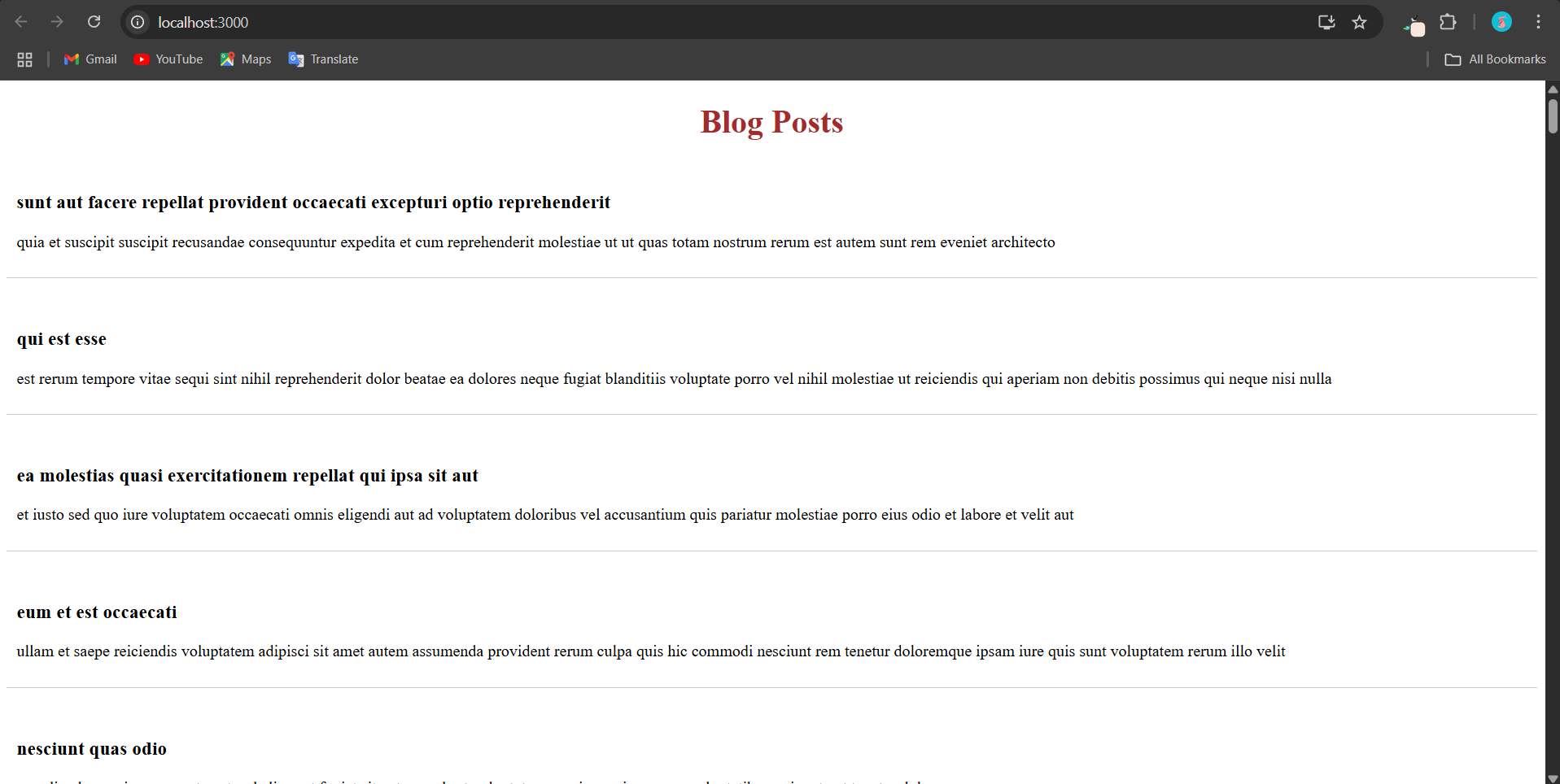
    </div>

  );

}

export default App;

**Output:**



**5.ReactJs-HOL:**import React from 'react';

import styles from './CohortDetails.module.css';

const CohortDetails = ({ cohort }) => {

  const { name, startDate, endDate, status } = cohort;

  return (

    <div className={styles.box}>

      <h3 style={{ color: status === 'ongoing' ? 'green' : 'blue' }}>

        {status.toUpperCase()}

      </h3>

      <dl>

        <dt>Name:</dt>

        <dd>{name}</dd>

        <dt>Start Date:</dt>

        <dd>{startDate}</dd>

        <dt>End Date:</dt>

        <dd>{endDate}</dd>

      </dl>

    </div>

  );

};

export default CohortDetails;

.ongoing {

  color: green;

}

.completed {

  color: blue;

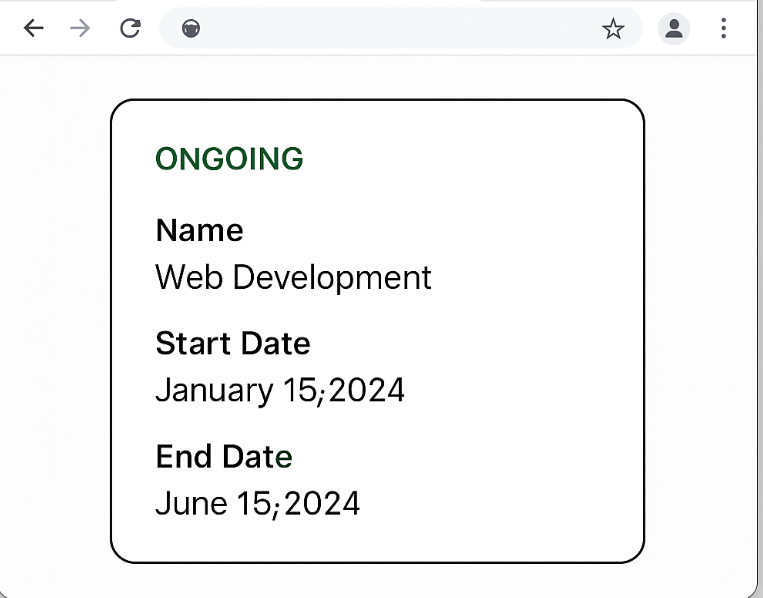
}

<h3 style={{ color: status === 'ongoing' ? 'green' : 'blue' }}>

  {status.toUpperCase()}

</h3>

**Output:**

****