**1.MY FIRST REACT**

**CONTEXT OF THE EXAMPLE:**

In this hands-on, I create a simple Single Page Application using React. We set up the React environment using create-react-app, modify the main component to display a welcome heading, and run it locally on localhost:3000. This introduces basic React structure and usage.

**CODE**

**App.js:**

import React from 'react';

function App() {

  return (

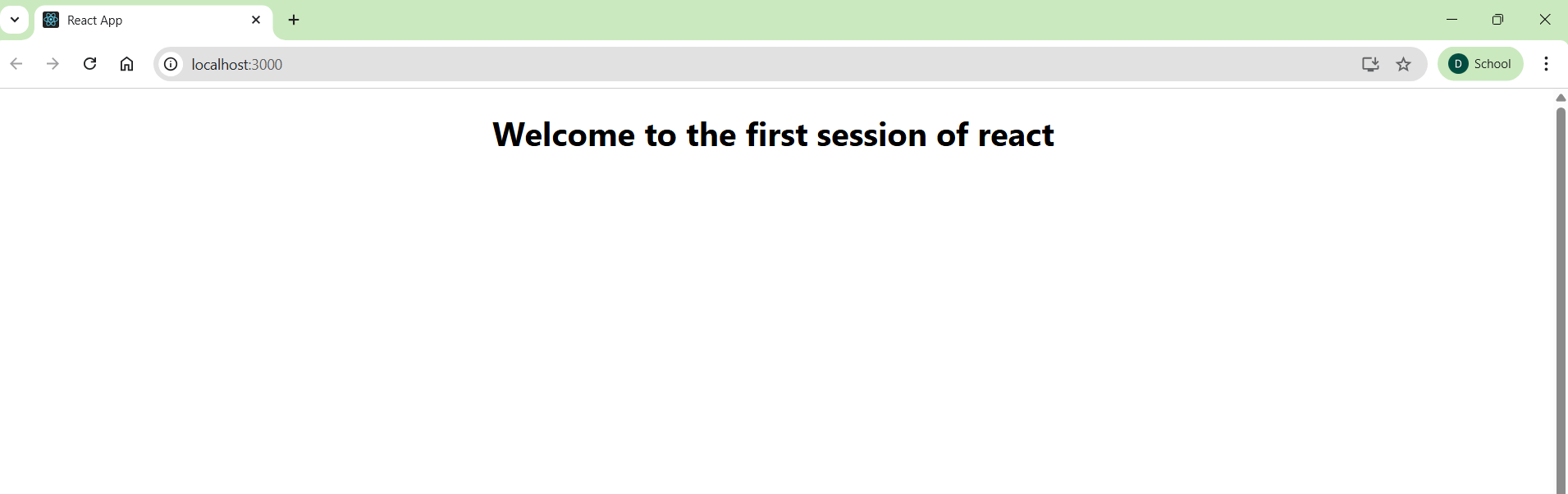
    <h1>Welcome to the first session of React</h1>

  );

}

export default App;

**OUTPUT:**



**2.STUDENT APP**

**CONTEXT OF THE EXAMPLE:**

In this lab, I create a React app named **StudentApp** using class components. I build and render three components — Home, About, and Contact — each showing a relevant welcome message for the Student Management Portal. This exercise helps me understand class components, their creation, and rendering in React.

**CODES:**

**Home.js:**

import React,{Component} from "react";

class Home extends Component {

    render() {

        return(

        <div>

        <h3>Welcome to the Home page of student Managment portal</h3>

        </div>

        );

    }

}

export default Home;

**About.js:**

import React,{Component} from "react";

class About extends Component {

    render() {

        return(

        <div>

        <h3>Welcome to the About page of student Managment portal</h3>

        </div>

        );

    }

}

export default About;

**Contact.js:**

import React,{Component} from "react";

class Contact extends Component {

    render() {

        return(

        <div>

        <h3>Welcome to the Contact page of student Managment portal</h3>

        </div>

        );

    }

}

export default Contact;

**App.js:**

import React from "react";

import './App.css';

import Home from "./Components/Home";

import About from "./Components/About";

import Contact from "./Components/Contact";

function App() {

  return (

    <div className="App">

      <Home />

      <About />

      <Contact />

    </div>

  );

}

export default App;

**App.css:**

h3 {

  display: flex;

  flex-direction: column;

  align-items: center;   /\* horizontal centering \*/

  justify-content: center; /\* vertical centering \*/

  min-height: 0vh;     /\* full screen height \*/

  text-align: center;    /\* center text \*/

}

.App-logo {

  height: 40vmin;

  pointer-events: none;

}

@media (prefers-reduced-motion: no-preference) {

  .App-logo {

    animation: App-logo-spin infinite 20s linear;

  }

}

.App-header {

  background-color: #282c34;

  min-height: 100vh;

  display: flex;

  flex-direction: column;

  align-items: center;

  justify-content: center;

  font-size: calc(10px + 2vmin);

  color: white;

}

.App-link {

  color: #61dafb;

}

@keyframes App-logo-spin {

  from {

    transform: rotate(0deg);

  }

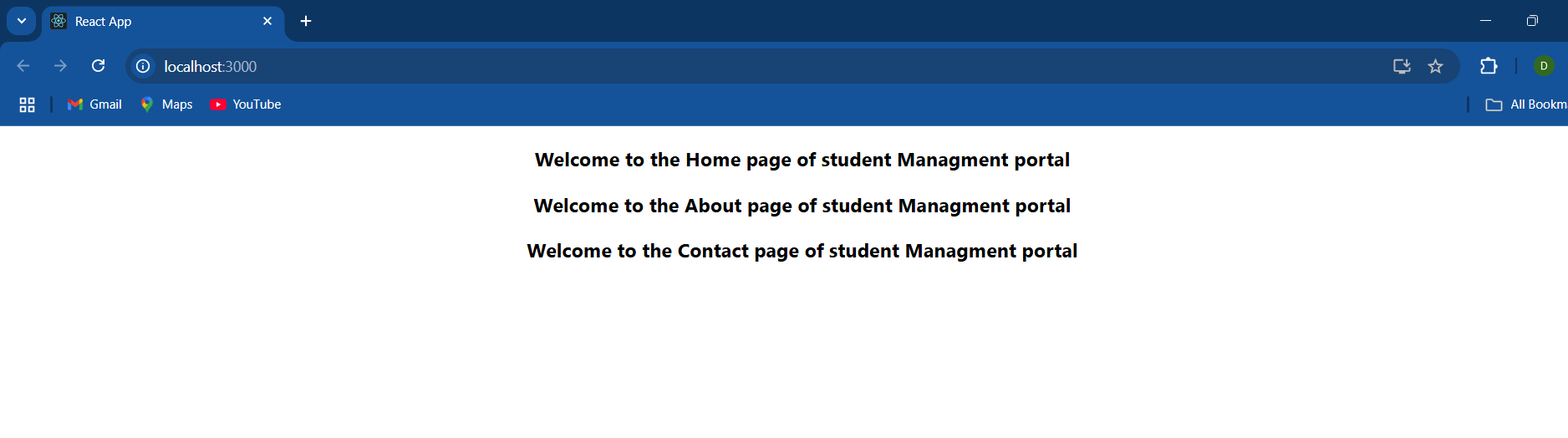
  to {

    transform: rotate(360deg);

  }

}

**OUTPUT:**

****

**3.SCORE CALCULATOR:**

**CONTEXT OF THE EXAMPLE:**

In this lab, I create a React app named scorecalculatorapp with a functional component called CalculateScore. The component accepts Name, School, Total, and Goal to compute and display a student’s average score. I also apply styles using an external CSS file to enhance the presentation.

**CODES:**

**App.js:**

import React from 'react';

import './App.css';

import CalculateScore from './Components/CalculateScore';

function App() {

  return (

    <div className="App">

      <CalculateScore />

    </div>

  );

}

export default App;

**CalculateScore.js:**

import React from 'react';

import '../Stylesheets/mystyle.css';

function CalculateScore() {

    const name = "Stevee";

    const school = "DNV Public School";

    const total = 488;

    const goal = 500;

    const average = (total / goal) \* 100;

    return (

        <div className="score-box">

            <h1 >Student Details:</h1>

            <h2 className='blue'> <span>Name: </span> {name}</h2>

            <h3 className='red'><span>School: </span>{school}</h3>

            <h3 className='indigo'><span>Total Score: </span>{total} Marks</h3>

            <h3 className='green'><span> Score: </span>{average.toFixed(2)}%</h3>

        </div>

    );

}

export default CalculateScore;

**mystyle.css:**

.score-box {

    margin: 20px auto;

    width: 50%;

    background-color: #f9f9f9;

    font-family: Arial, sans-serif;

    box-shadow: 0 0 10px rgba(46, 207, 65, 0.1);

    text-align: center;

}

h1{

    margin-bottom: 40px;

}

h3{

    margin-top: -20px;

}

.score-box h1{

    color: #740c0c;

}

span{

    font-weight: bolder;

}

.blue  {

    color: #0000FF;

}

.red {

    color: red;

}.green  {

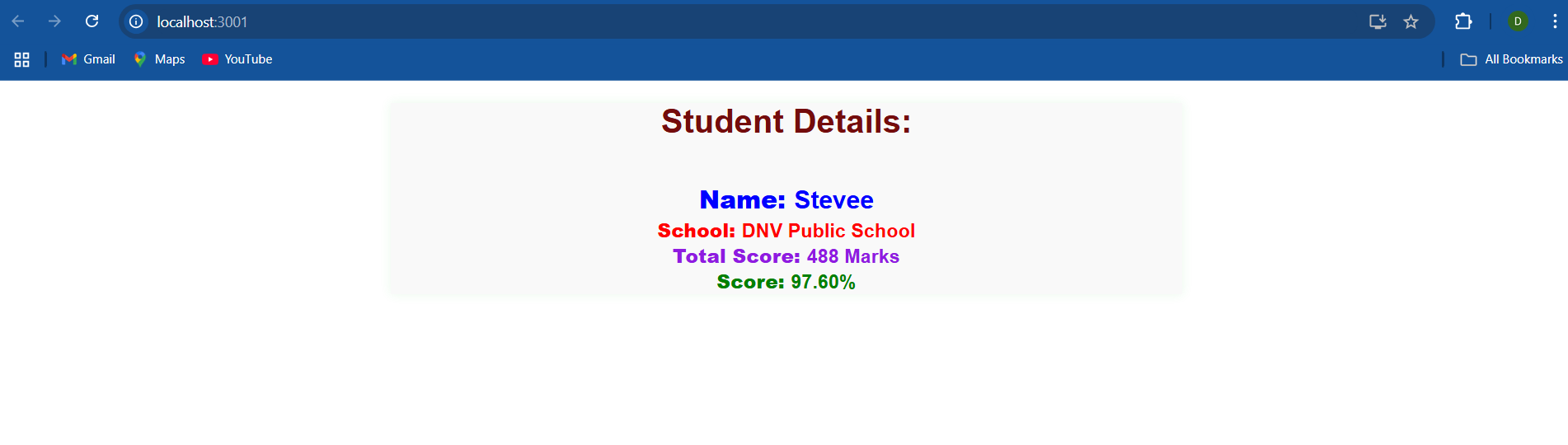
    color: green;

}.indigo {

    color: rgb(143, 27, 225);

}

**OUTPUT:**

****

**4.CREATE-REACT-APP**

**CONTEXT OF THE EXAMPLE:**

In this exercise, I build a basic React app called **StudentApp** using class components. We create and render three components — Home, About, and Contact — each displaying a welcome message for the Student Management Portal. This helps in understanding component structure, creation, and rendering in React.

**CODE:**

**App.js:**

import React from 'react';

import './App.css';

import Posts from './Posts';

function App() {

  return (

    <div className="App">

      <Posts />

    </div>

  );

}

export default App;

**Post.js:**

class Post {

  constructor(id, title, body) {

    this.id = id;

    this.title = title;

    this.body = body;

  }

}

export default app;

**Posts.js:**

// JS Posts.js UX

import React from 'react';

class Posts extends React.Component {

  constructor(props) {

    super(props);

    this.state = {

      posts: []

    };

  }

  loadPosts() {

    fetch('https://jsonplaceholder.typicode.com/posts')

      .then((response) => response.json())

      .then((data) => {

        this.setState({ posts: data });

      })

      .catch((error) => {

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

      });

  }

  componentDidMount() {

    this.loadPosts();

  }

  render() {

    return (

      <div>

        <h1>Posts</h1>

        {this.state.posts.map(post => (

          <div key={post.id}>

            <h2>{post.title}</h2>

            <p>{post.body}</p>

            <hr />

          </div>

        ))}

      </div>

    );

  }

  componentDidCatch(error, info) {

    alert("An error occurred: " + error.message);

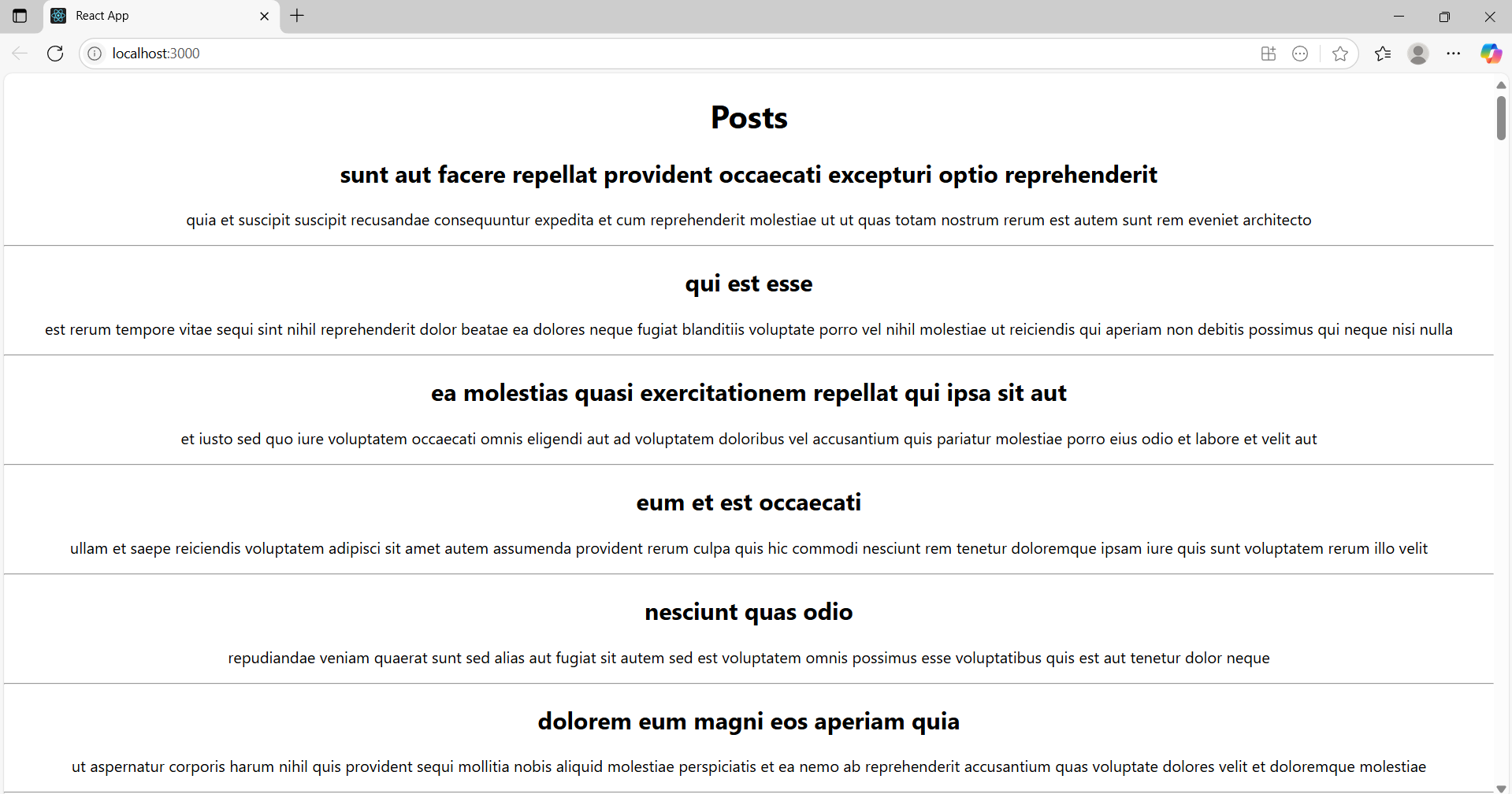
    console.error("Error details:", info);

  }

}

export default Posts;

**OUTPUT:**



**5.COHORT DETAILS**

**Context of the Example:**

In this task, I style a React application dashboard for MyAcademy at Cognizant by using CSS Modules and inline styles. I define reusable styles in CohortDetails.module.css and apply conditional formatting to elements based on cohort status, enhancing the visual structure and clarity of the component.

**CODE:**

**App.js:**

import React from 'react';

import Dashboard from './Dashboard';

function App() {

  return (

    <div>

      <Dashboard />

    </div>

  );

}

export default App;

**Dashboard.jsx:**

import React from 'react';

import CohortDetails from './components/CohortDetails';

const cohortData = [

  {

    title: "INTADMDF10-.NET FSD",

    startDate: "22-Feb-2022",

    status: "Scheduled",

    coach: "Aathma",

    trainer: "Jojo Jase"

  },

  {

    title: "ADM21JF014 - Java FSD",

    startDate: "10-Sep-2021",

    status: "Ongoing",

    coach: "Apoorv",

    trainer: "Elisa Smith"

  },

  {

    title: "CDBJF21025-Java FSD",

    startDate: "24-Dec-2021",

    status: "Ongoing",

    coach: "Aathma",

    trainer: "John Doe"

  }

];

const Dashboard = () => {

  return (

    <div>

      <h2>Cohorts Details</h2>

      {cohortData.map((cohort, index) => (

        <CohortDetails key={index} cohort={cohort} />

      ))}

    </div>

  );

};

export default Dashboard;

**CohortDetails.jsx:**

import React from 'react';

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

const CohortDetails = ({ cohort }) => {

  const isGreenTitle = cohort.title.startsWith('ADM') || cohort.title.startsWith('CDB');

  return (

    <div className={styles.box}>

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

        {cohort.title}

      </h3>

      <dl>

        <dt>Started On</dt>

        <dd>{cohort.startDate}</dd>

        <dt>Current Status</dt>

        <dd>{cohort.status}</dd>

        <dt>Coach</dt>

        <dd>{cohort.coach}</dd>

        <dt>Trainer</dt>

        <dd>{cohort.trainer}</dd>

      </dl>

    </div>

  );

};

export default CohortDetails;

**CohortDetails.module.css:**

.box {

  width: 300px;

  display: inline-block;

  margin: 10px;

  padding: 10px 20px;

  border: 1px solid black;

  border-radius: 10px;

}

dt {

  font-weight: 500;

}

**OUTPUT:**

