ReactJS-HOL

## Exercise1: “myfirstreact” App

Step: 1: Open the terminal in VS Code using Ctrl+J

Step: 2: Create a new react project using the command

npx create-react-app myfirstreactapp

Step: 3: Change the directory of the terminal using the command

cd myfirstreactapp

Step: 4: Edit the App.js file

import React from 'react';

function App() {

  return (

    <div>

      <h1>Welcome to the first session of React, Abi Ayshwariya S</h1>

    </div>

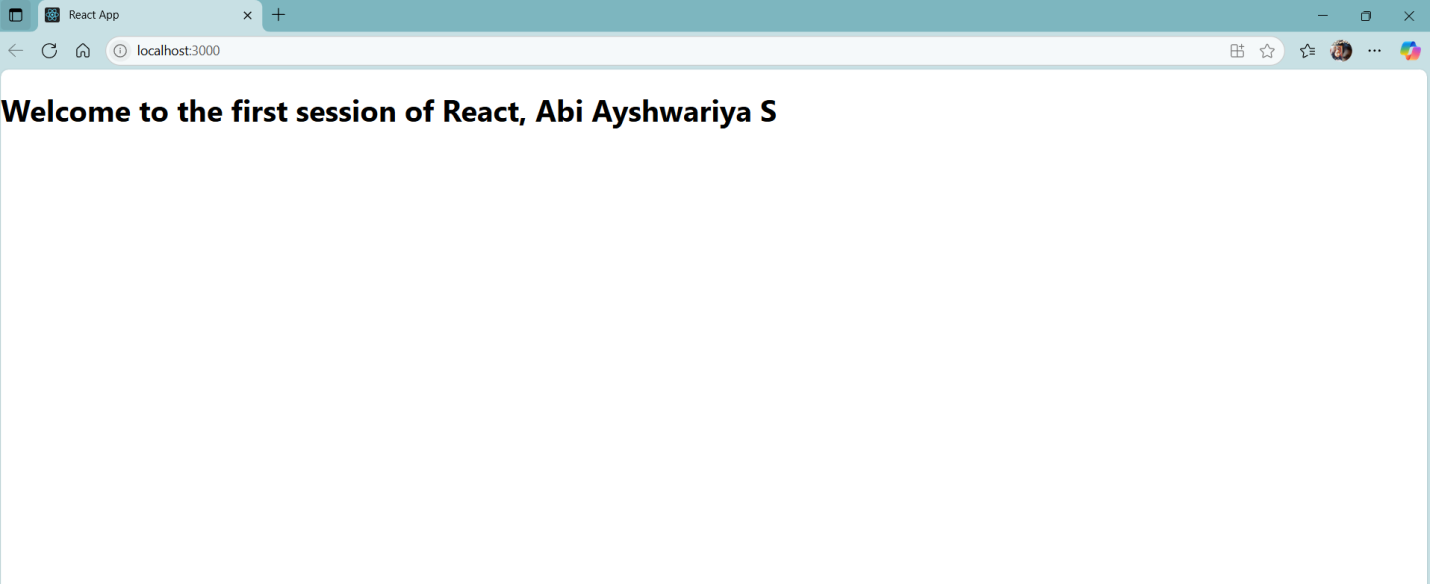
  );

}

export default App;

Step: 5: Run the command npm start

Open <http://localhost:3000>



## Exercise: 2: “StudentApp” – Multiple Components

### Steps:

1. Create App:

**npx create**-react-app StudentApp

**2.** Change the directory

cd StudentApp

3. Create multiple Components as follows

Mkdir src/Components

Components/Home.js

import React from 'react';

function Home() {

  return <h2>Welcome to the Home page of Student Management Portal</h2>;

}

export default Home;

Components/Contact.js

import React from 'react';

function About() {

  return <h2>Welcome to the About page of the Student Management Portal</h2>;

}

export default About;

Components/About.js

import React from 'react';

function About() {

  return <h2>Welcome to the About page of the Student Management Portal</h2>;

}

export default About;

4: Edit the App.js file

App.js

import React from 'react';

import Home from './Components/Home';

import About from './Components/About';

import Contact from './Components/Contact';

function App() {

  return (

    <div>

      <Home />

      <About />

      <Contact />

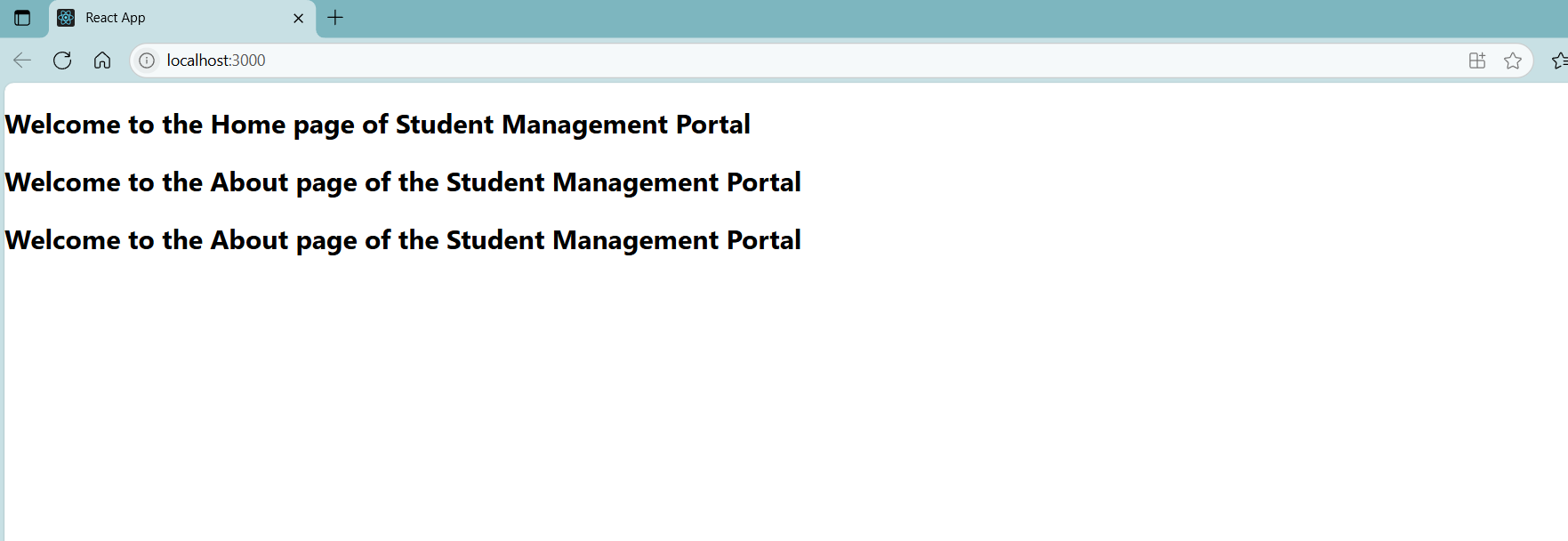
    </div>

  );

}

export default App;

Output



## Exercise: 3: “scorecalculatorapp” with Styling

1. Create App:

**npx create**-react-app scorecalculatorapp

**2.** Change the directory

cd scorecalculatorapp

3. Create Component CalculateScore.js as follows

mkdir src/Components/CalculateScore.js

CalculateScore.js

import React from 'react';

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

function CalculateScore({ name, school, total, goal }) {

  const average = total / goal;

  return (

    <div className="score-box">

      <h2>Student Score Details</h2>

      <p>Name: {name}</p>

      <p>School: {school}</p>

      <p>Total Score: {total}</p>

      <p>Goal: {goal}</p>

      <p>Average: {average.toFixed(2)}</p>

    </div>

  );

}

export default CalculateScore;

Edit App.js

import React from 'react';

import CalculateScore from './Components/CalculateScore';

function App() {

  return (

    <center>

    <div>

      <CalculateScore name="Abi Ayshwariya S" school="Vel Tech University" total={450} goal={5} />

    </div>

    </center>

  );

}

export default App;

Create custom stylesheet using the command

mkdir src/Stylesheets/mystyle.css

mystyle.css

.score-box {

  border: 1px solid #333;

  padding: 20px;

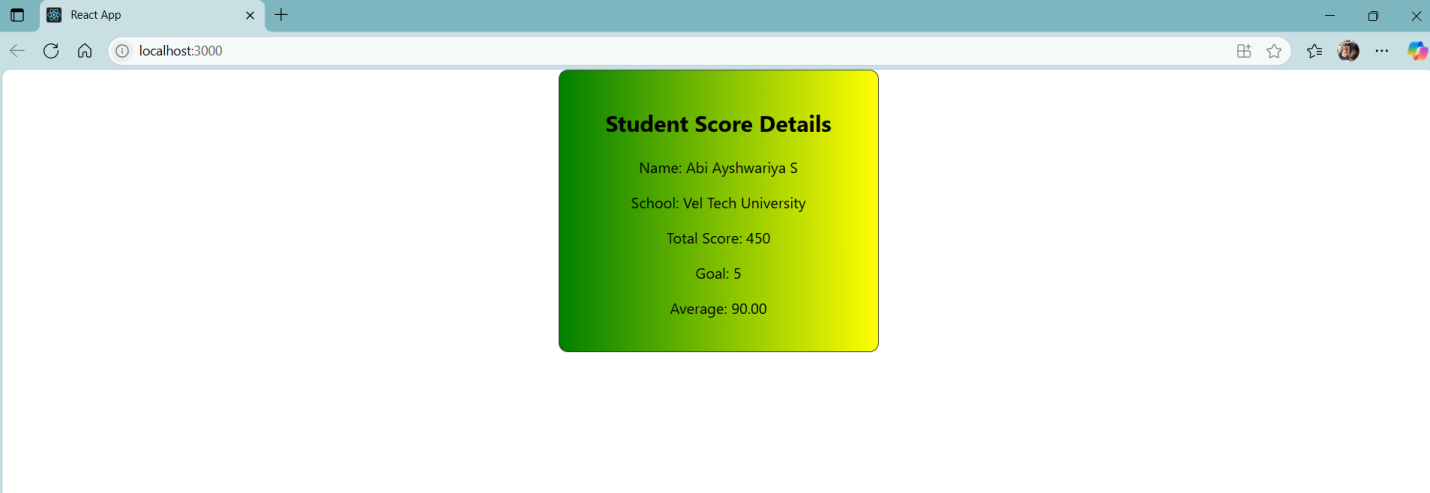
  border-radius: 10px;

  width: 300px;

  background:linear-gradient(90deg,green,yellow);

}

## Output



## Exercise: 4: “blogapp” with Lifecycle Hooks

1. Create App:

**npx create**-react-app blogapp

**2.** Change the directory

cd blogapp

3. Create Posts.js

Posts.js

import React, { Component } from 'react';

class Posts extends Component {

  constructor(props) {

    super(props);

    this.state = {

      posts: [],

      hasError: false,

    };

  }

  componentDidMount() {

    this.loadPosts();

  }

  loadPosts() {

    const customPosts = [

      {

        id: 1,

        title: 'Getting Started with React',

        body: 'React is a JavaScript library for building user interfaces. Start with Create React App to get up and running quickly.',

      },

      {

        id: 2,

        title: 'Understanding Component Lifecycle',

        body: 'React components go through lifecycle stages. Learn how componentDidMount and componentWillUnmount work.',

      },

      {

        id: 3,

        title: 'JavaScript ES6 Features Every Developer Should Know',

        body: 'Let, const, arrow functions, template literals, destructuring, and spread/rest operators are essential in modern JS.',

      },

      {

        id: 4,

        title: 'Why Use Virtual DOM in React?',

        body: 'Virtual DOM improves performance by reducing actual DOM manipulations. React compares virtual DOM with real DOM efficiently.',

      },

      {

        id: 5,

        title: 'How to Manage State in React',

        body: 'Learn the difference between local, global, and lifted state. Understand useState, useReducer, and context API.',

      }

    ];

    // Simulate async fetch

    setTimeout(() => {

      this.setState({ posts: customPosts });

    }, 1000);

  }

  componentDidCatch(error, info) {

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

    this.setState({ hasError: true });

  }

  render() {

    const containerStyle = {

      fontFamily: 'Segoe UI, sans-serif',

      padding: '20px',

      maxWidth: '700px',

      margin: 'auto',

    };

    const titleStyle = {

      color: '#2c3e50',

      textAlign: 'center',

      marginBottom: '30px',

    };

    const postTitleStyle = {

      color: '#2980b9',

      fontSize: '20px',

      marginBottom: '5px',

    };

    const postBodyStyle = {

      fontSize: '16px',

      lineHeight: '1.6',

      marginBottom: '15px',

    };

    const dividerStyle = {

      borderTop: '1px solid #ccc',

      marginBottom: '20px',

    };

    return (

      <div style={containerStyle}>

        <h2 style={titleStyle}>Tech Blog Posts</h2>

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

          <div key={post.id}>

            <h3 style={postTitleStyle}>{post.title}</h3>

            <p style={postBodyStyle}>{post.body}</p>

            <div style={dividerStyle}></div>

          </div>

        ))}

      </div>

    );

  }

}

export default Posts;

App.js

import React from 'react';

import Posts from './Posts';

function App() {

  return (

    <div>

      <Posts />

    </div>

  );

}

export default App;

App.css

.App {

  text-align: center;

}

.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);

  }

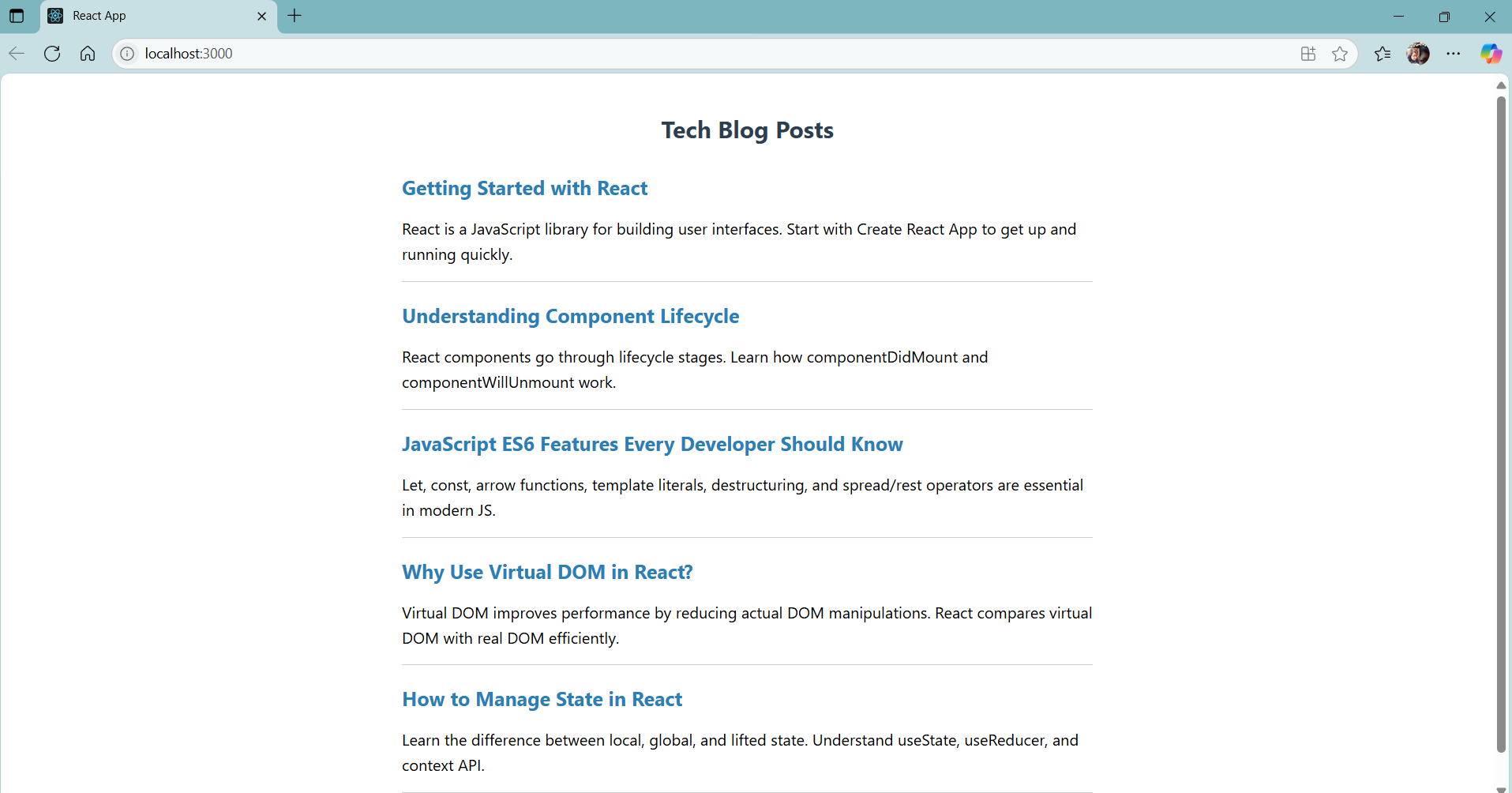
  to {

    transform: rotate(360deg);

  }

}

Output:



## Exercise: 5: Styling Components (Cohort Dashboard)

CohortDetails.js

import React from 'react';

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

function CohortDetails({ cohort }) {

  const titleStyle = {

    color: cohort.status === 'ongoing' ? 'green' : 'blue'

  };

  return (

    <div className={styles.box}>

      <h3 style={titleStyle}>{cohort.name}</h3>

      <dl>

        <dt>Status:</dt>

        <dd>{cohort.status}</dd>

        <dt>Start Date:</dt>

        <dd>{cohort.startDate}</dd>

        <dt>End Date:</dt>

        <dd>{cohort.endDate}</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;

  background-color: #f9f9f9;

}

dt {

  font-weight: 500;

}

.ongoing {

  color: green;

}

.completed {

  color: blue;

}

Output

