WEEK-6  
  
  
App.js  
import { render, screen } from '@testing-library/react';

import App from './App';

test('renders learn react link', () => {

render(<App />);

const linkElement = screen.getByText(/learn react/i);

expect(linkElement).toBeInTheDocument();

});

OUTPUT:  
  
  
  
  
  
  
  
2.ReactJS-HOL

Src/Components/Home.js  
import React, { Component } from 'react';

class Home extends Component {

render() {

return (

<div>

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

</div>

);

}

}

export default Home;

Src/Components/About.js  
  
import React, { Component } from 'react';

class About extends Component {

render() {

return (

<div>

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

</div>

);

}

}

export default About;

Src/Components/Contact.js  
import React, { Component } from 'react';

class Contact extends Component {

render() {

return (

<div>

<h2>Welcome to the Contact page of the Student Management Portal</h2>

</div>

);

}

}

export default Contact;

Src/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 className="App">

<h1>Student Management Portal</h1>

<Home />

<About />

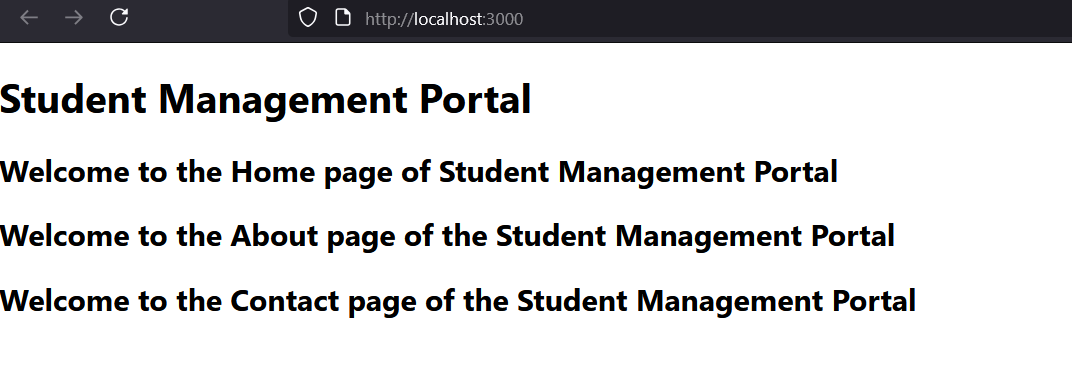
<Contact />

</div>

);

}

export default App;

OUTPUT:  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
3-ReactJS-HOL  
  
  
  
src/Components/CalculateScore.js  
  
import React from 'react';

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

function CalculateScore(props) {

const average = props.total / props.goal;

return (

<div className="score-card">

<h2>Student Score Calculator</h2>

<p><strong>Name:</strong> {props.name}</p>

<p><strong>School:</strong> {props.school}</p>

<p><strong>Total Marks:</strong> {props.total}</p>

<p><strong>Goal:</strong> {props.goal}</p>

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

</div>

);

}

export default CalculateScore;

Src/Stylesheets/mystyles.css  
  
.score-card {

background-color: #f2f2f2;

border: 2px solid #4caf50;

border-radius: 10px;

padding: 20px;

margin: 30px auto;

width: 60%;

text-align: left;

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

}

.score-card h2 {

color: #4caf50;

text-align: center;

}

.score-card p {

font-size: 18px;

line-height: 1.5;

}

Src/App.js  
  
import React from 'react';

import './App.css';

import CalculateScore from './Components/CalculateScore';

function App() {

return (

<div className="App">

<CalculateScore

name="John Doe"

school="Greenwood High School"

total={450}

goal={5}

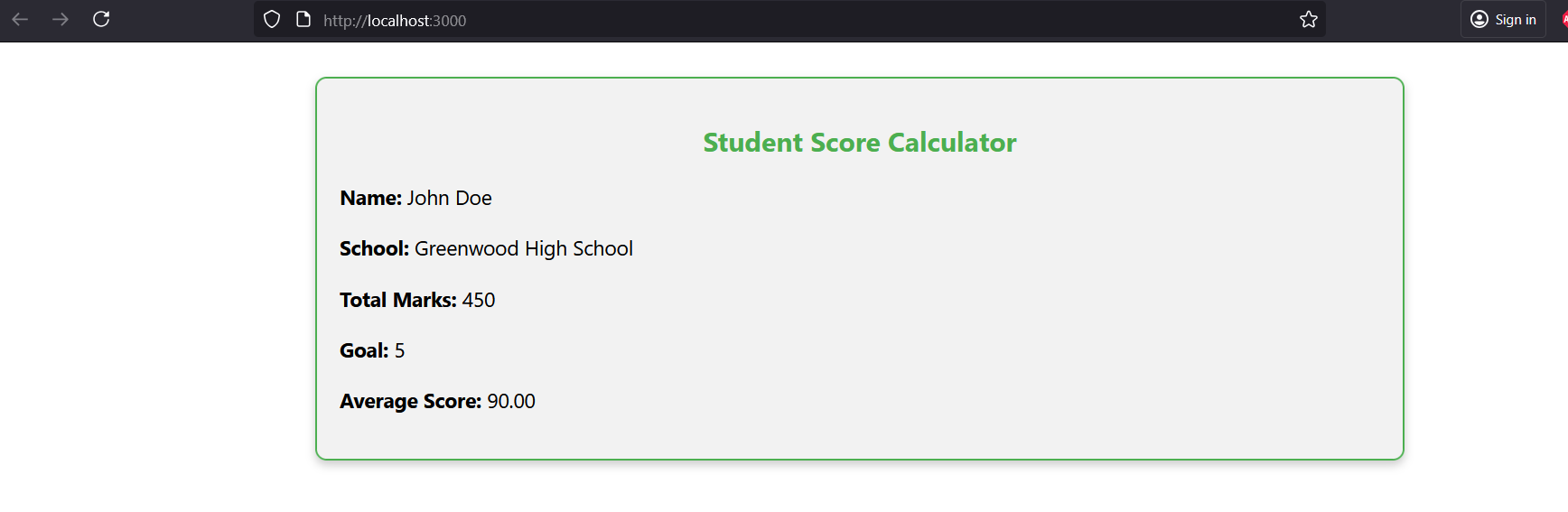
/>

</div>

);

}

export default App;

OUTPUT:  
  
  
  
  
  
  
  
  
  
  
  
4-ReactJS-HOL  
  
src/Post.js  
import React from 'react';

function Post(props) {

return (

<div className="post">

<h3>{props.title}</h3>

<p>{props.body}</p>

</div>

);

}

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,

};

}

// Method to load posts using fetch

loadPosts = () => {

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

.then((response) => {

if (!response.ok) {

throw new Error('Failed to fetch posts');

}

return response.json();

})

.then((data) => {

this.setState({ posts: data });

})

.catch((error) => {

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

this.setState({ hasError: true });

});

};

// componentDidMount lifecycle hook

componentDidMount() {

this.loadPosts();

}

// Error boundary lifecycle hook

componentDidCatch(error, info) {

alert('An error occurred in Posts component.');

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

console.error('Error Info:', info);

this.setState({ hasError: true });

}

// Render posts

render() {

if (this.state.hasError) {

return <h2>Something went wrong. Please try again later.</h2>;

}

return (

<div>

<h2>Blog Posts</h2>

{this.state.posts.slice(0, 10).map((post) => (

<Post key={post.id} title={post.title} body={post.body} />

))}

</div>

);

}

}

export default Posts;

Src/App.js  
  
import React from 'react';

import './App.css';

import Posts from './Posts';

function App() {

return (

<div className="App">

<h1>Welcome to the BlogApp</h1>

<Posts />

</div>

);

}

export default App;

Src/App.css  
.App {

text-align: center;

font-family: Arial, sans-serif;

}

.post {

border: 1px solid #ccc;

padding: 15px;

margin: 10px auto;

width: 80%;

text-align: left;

border-radius: 8px;

background-color: #f9f9f9;

}

.post h3 {

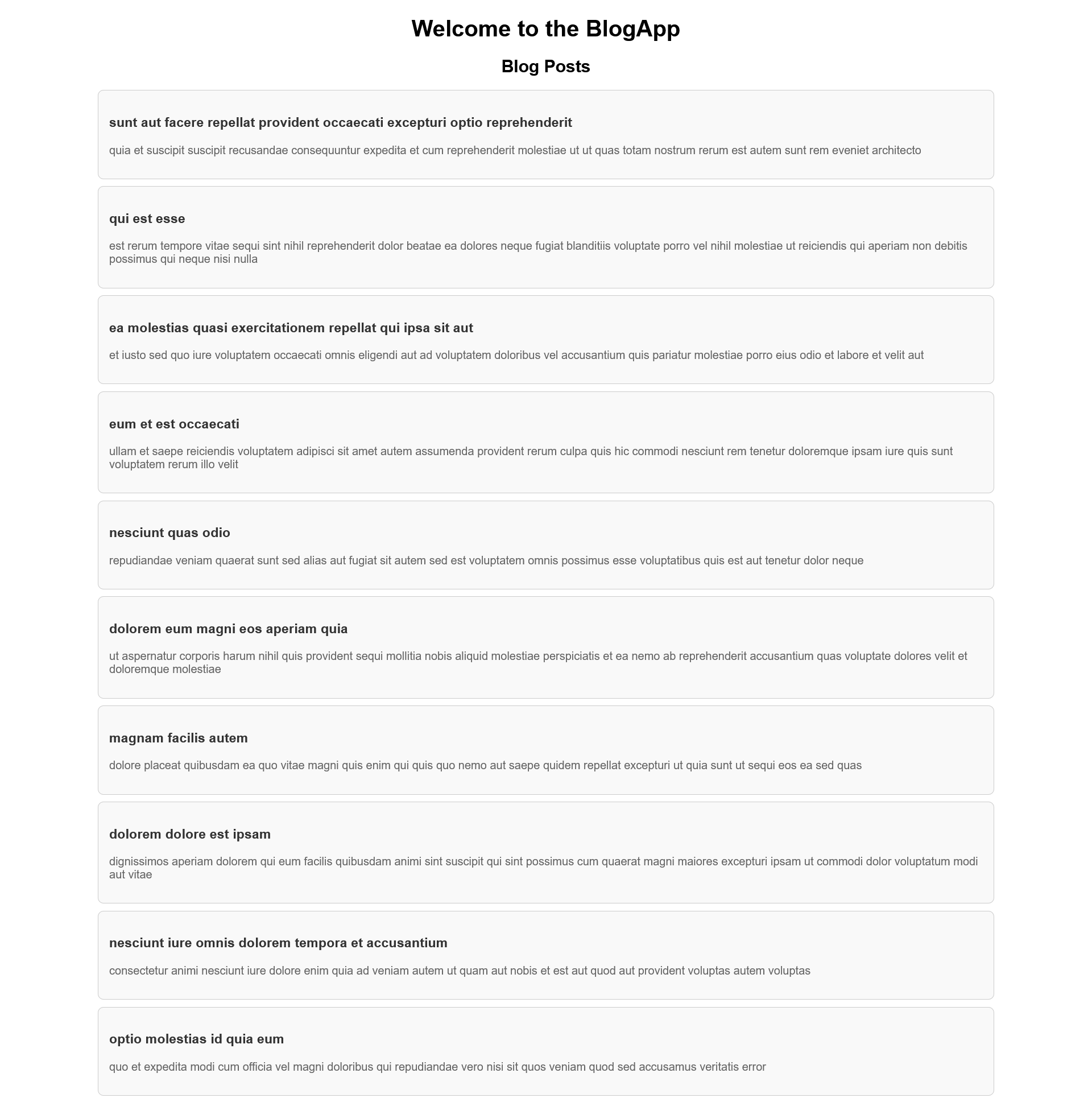
color: #333;

}

.post p {

color: #666;

}

OUTPUT:  
  
  
  
  
  
  
  
  
  
  
  
  
5-ReactJS-HOL  
  
src/Components/CohortDetails.module.css  
.box {

width: 300px;

display: inline-block;

margin: 10px;

padding: 10px 20px;

border: 1px solid black;

border-radius: 10px;

vertical-align: top;

}

dt {

font-weight: 500;

}

Src/Components/CohortDetails.js  
  
import React from 'react';

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

function CohortDetails(props) {

const { cohortId, track, startDate, status, coach, trainer } = props;

// Conditional color for title based on status

const titleColor = status.toLowerCase() === 'ongoing' ? 'green' : 'blue';

return (

<div className={styles.box}>

<h3 style={{ color: titleColor }}>

{cohortId} - {track}

</h3>

<dl>

<dt>Started On</dt>

<dd>{startDate}</dd>

<dt>Current Status</dt>

<dd>{status}</dd>

<dt>Coach</dt>

<dd>{coach}</dd>

<dt>Trainer</dt>

<dd>{trainer}</dd>

</dl>

</div>

);

}

export default CohortDetails;

Src/App.js  
import React from 'react';

import './App.css';

import CohortDetails from './Components/CohortDetails';

function App() {

return (

<div className="App">

<h2>Cohorts Details</h2>

<CohortDetails

cohortId="INTADMDF10"

track=".NET FSD"

startDate="22-Feb-2022"

status="Scheduled"

coach="Aathma"

trainer="Jojo Jose"

/>

<CohortDetails

cohortId="ADM21JF014"

track="Java FSD"

startDate="10-Sep-2021"

status="Ongoing"

coach="Apoorv"

trainer="Elisa Smith"

/>

<CohortDetails

cohortId="CDBJF21025"

track="Java FSD"

startDate="24-Dec-2021"

status="Ongoing"

coach="Aathma"

trainer="John Doe"

/>

</div>

);

}

export default App;

src/ App.css

.App {

padding: 20px;

font-family: sans-serif;

}

OUTPUT:  
