**REACT**

**1. ReactJS-HOL**

**1. Define SPA and its Benefits**

**SPA (Single Page Application)** is a web application that loads a single HTML page and dynamically updates the content without refreshing the entire page.

**Benefits:**

* Faster navigation since there's no full page reload.
* Better user experience with smooth transitions.
* Uses APIs efficiently by only fetching required data.
* Works well on mobile and dynamic UIs.

**2. Define React and Identify its Working**

**React** is a JavaScript library for building user interfaces, especially for single-page applications.

**How React works:**

* React divides the UI into small reusable pieces called **components**.
* It uses **JSX**, a mix of JavaScript and HTML-like syntax.
* It updates the user interface using a **Virtual DOM**.
* It follows **one-way data flow** (data flows from parent to child).

3. Differences Between SPA and MPA

| **Feature** | **SPA (Single Page App)** | **MPA (Multi Page App)** |
| --- | --- | --- |
| Page Loading | Loads once, updates dynamically | Loads a new page on every click |
| Speed | Faster after the first load | Slower due to full page reloads |
| User Experience | Smooth and interactive | Traditional and sometimes laggy |
| Server Interaction | Uses APIs to fetch data | Loads new pages from the server |

**4. Pros and Cons of Single-Page Application**

**Pros:**

* Very fast and responsive interface.
* No need to reload the entire page.
* Smooth navigation like mobile apps.
* Easy to reuse code using components.

**Cons:**

* SEO (Search Engine Optimization) is harder.
* Initial load time may be slower.
* Requires JavaScript to be enabled in browser.
* Needs extra care for browser history and security.

**5. Explain About React**

React is a front-end JavaScript library developed by Facebook for building interactive user interfaces, especially for single-page applications.

**Key points:**

* Uses a component-based structure.
* Efficiently updates the UI using Virtual DOM.
* JSX makes code readable and easier to write.
* React has features like hooks for handling logic and state.
* Popularly used in apps like Instagram, Facebook, and Netflix.

**6. Define Virtual DOM**

**Virtual DOM** is a lightweight JavaScript object that is a copy of the real DOM.

**Working of Virtual DOM:**

* React makes changes in the virtual DOM first.
* Then it compares (diffs) it with the old virtual DOM.
* Only the differences are updated in the real DOM.

**Advantages:**

* Improves performance.
* Reduces time-consuming DOM updates.
* Makes UI rendering fast and efficient.

**7. Features of React**

* **JSX:** Allows writing HTML in JavaScript.
* **Component-Based:** Breaks UI into independent, reusable pieces.
* **Virtual DOM:** Speeds up UI updates by avoiding full-page reloads.
* **One-Way Data Binding:** Data flows in one direction.
* **Hooks:** Functions like useState and useEffect handle logic.
* **High Performance:** Faster rendering and better efficiency.
* **Rich Ecosystem:** Includes tools like React Router and Redux.

**APP.js**

function App() {

return (

<div>

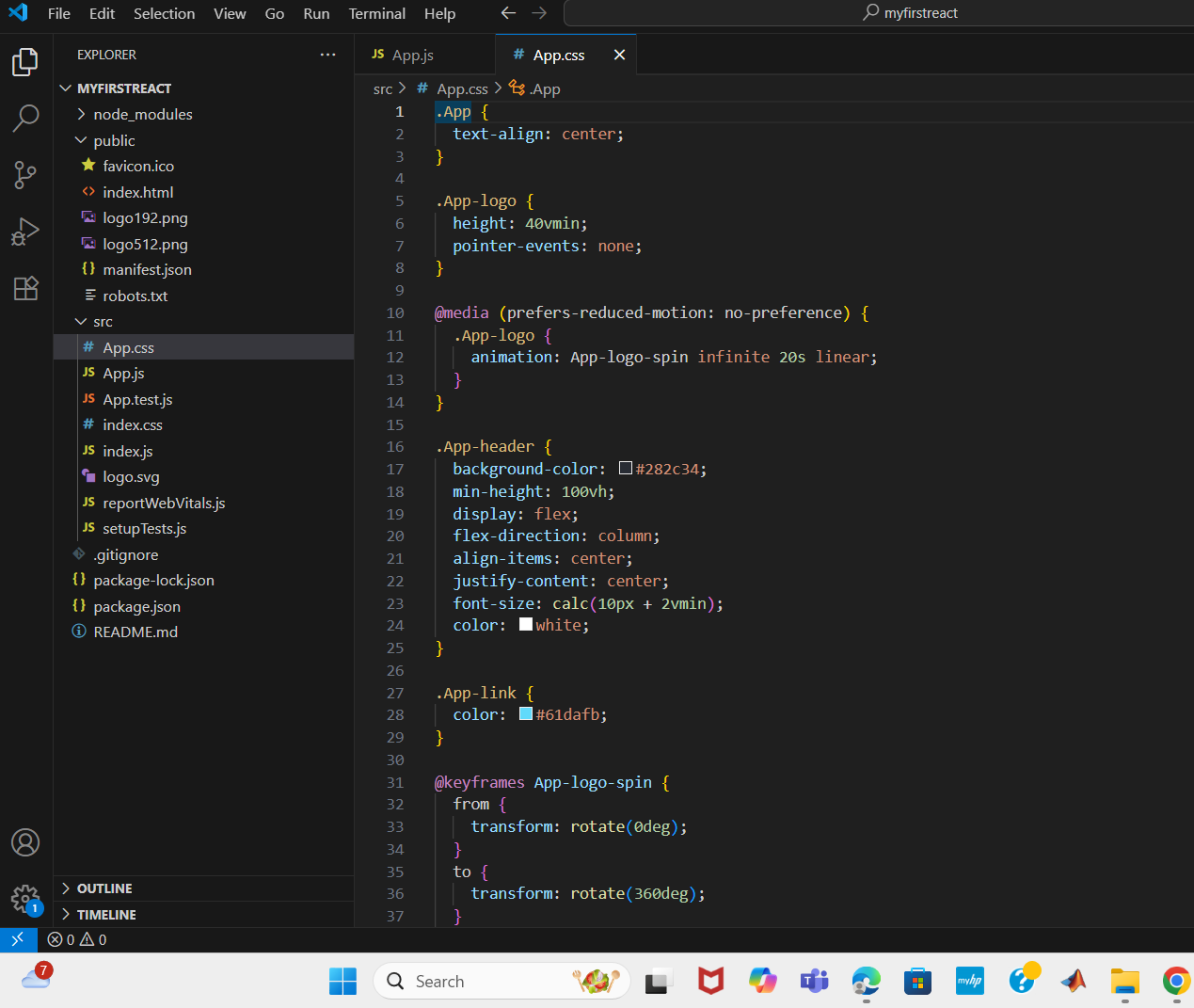
<h1>Welcome to the first session of React</h1>

</div>

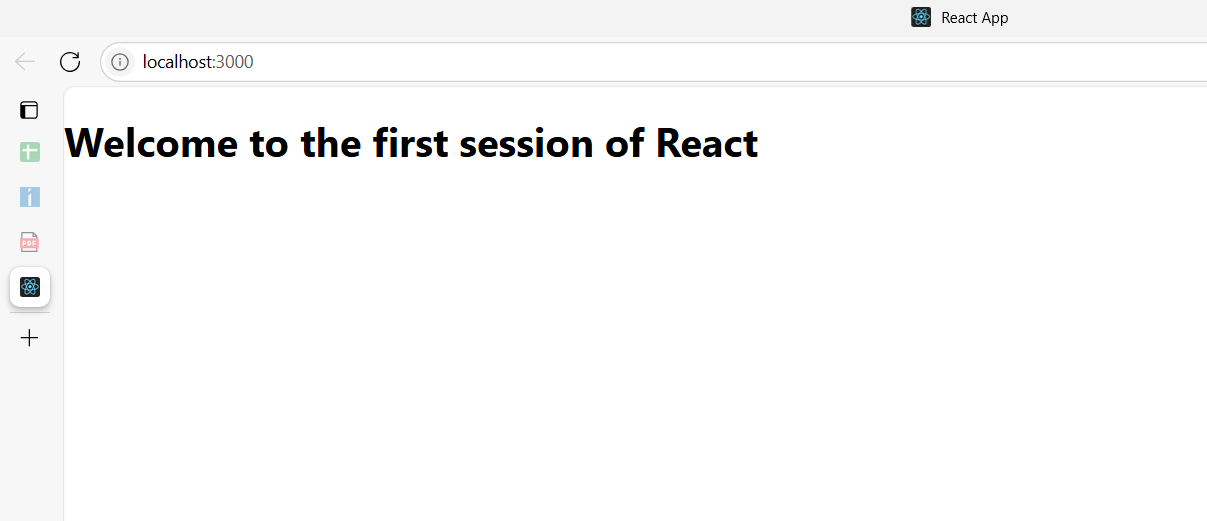
);

}

export default App;



**OUTPUT:**

****

**2. ReactJS-HOL**

**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;

**About.js**

import React from 'react';

function About() {

return (

<div>

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

</div>

);

}

export default About;

**Contact.js**

import React from 'react';

function Contact() {

return (

<div>

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

</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">

<h1>Student Management Portal</h1>

<Home />

<About />

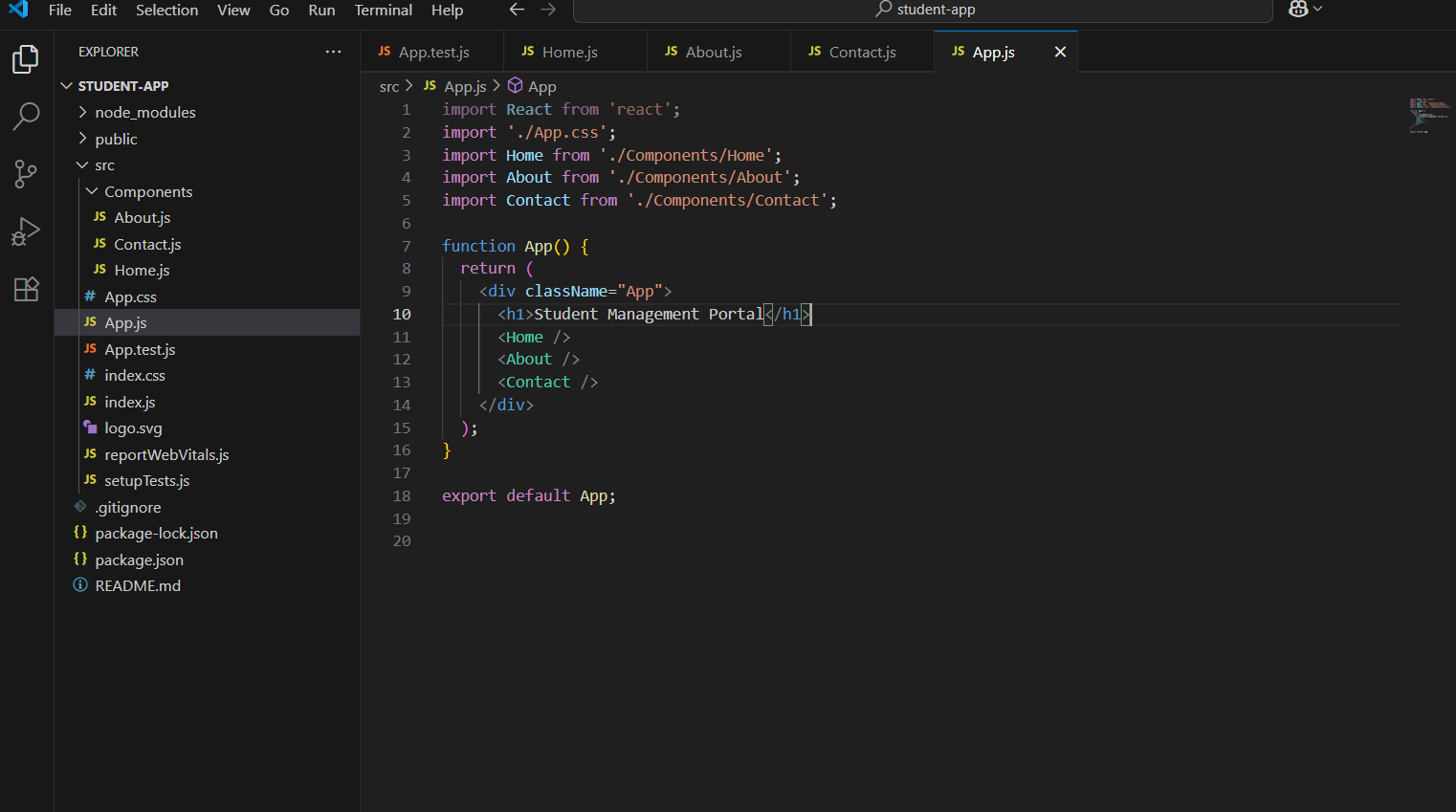
<Contact />

</div>

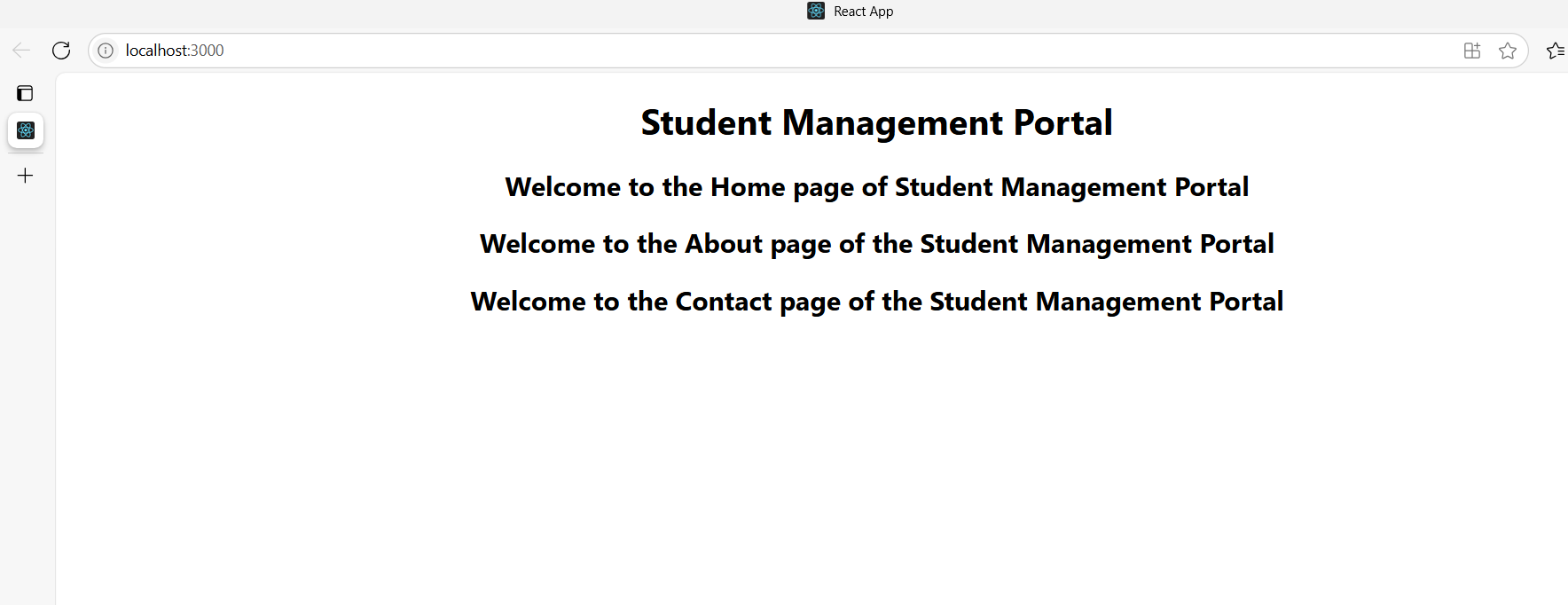
);

}

export default App;



**OUTPUT:**

****

**3. ReactJS-HOL**

**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 Report</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}</p>

</div>

);

}

export default CalculateScore;

**mystyle.css**

.score-card {

background-color: #f3f3f3;

border: 2px solid #ccc;

padding: 20px;

border-radius: 12px;

width: 300px;

margin: 20px auto;

font-family: Arial, sans-serif;

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

}

.score-card h2 {

text-align: center;

color: #333;

}

.score-card p {

font-size: 16px;

margin: 8px 0;

}

**App.js**

import React from 'react';

import './App.css';

import CalculateScore from './Components/CalculateScore';

function App() {

return (

<div className="App">

<CalculateScore

name="Affu"

school="ABC Matric Hr. Sec. School"

total={450}

goal={5}

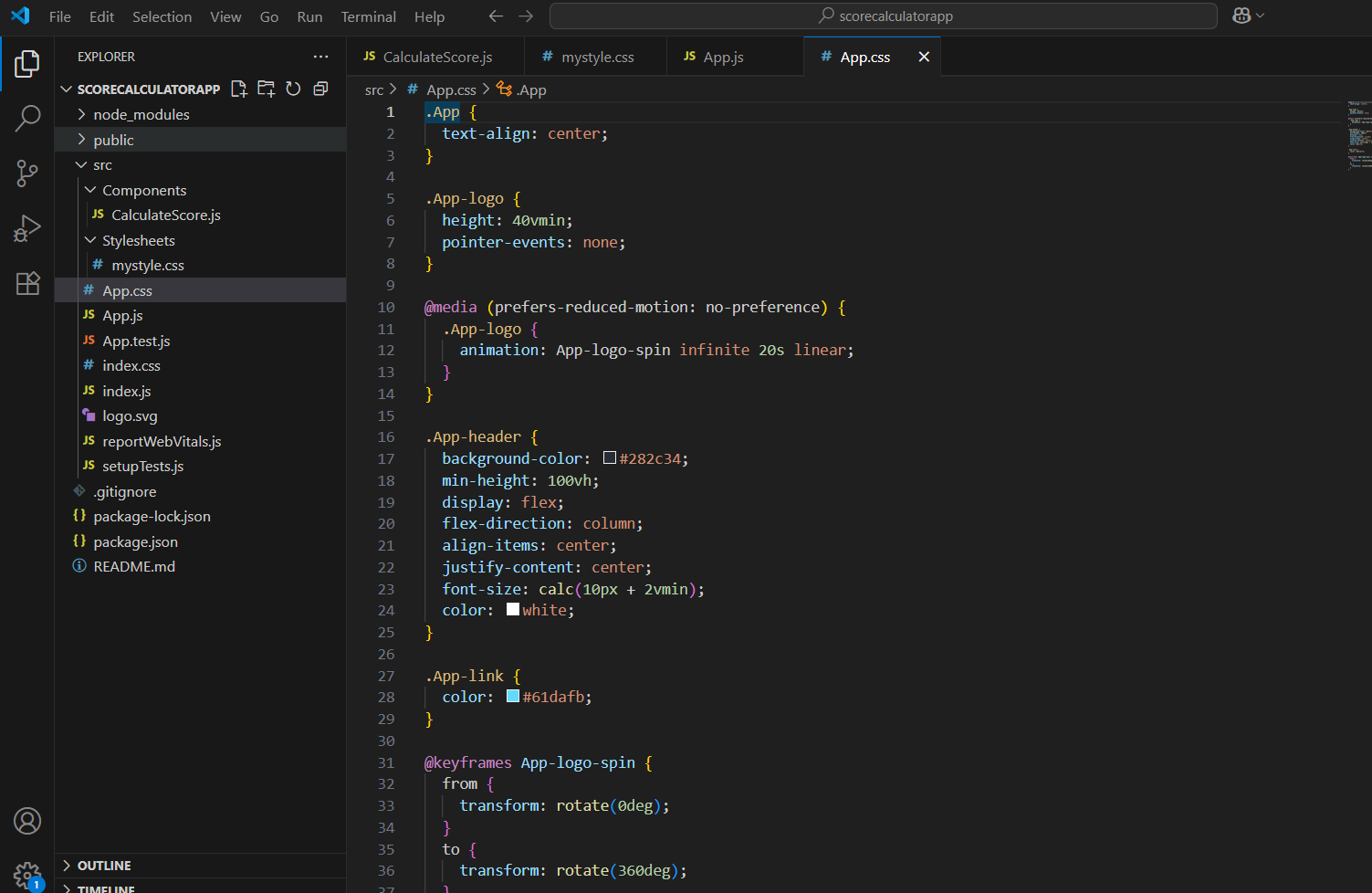
/>

</div>

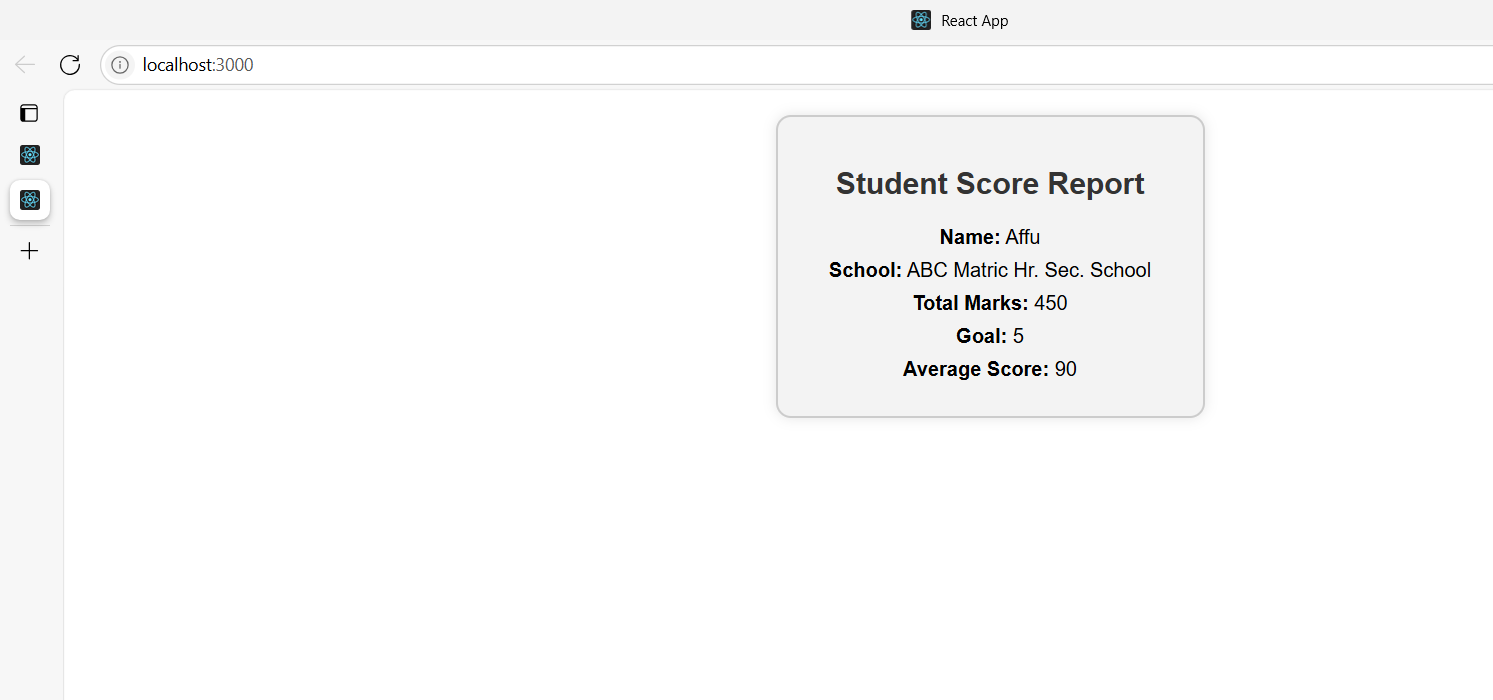
);

}

export default App;



**OUTPUT:**

****

**4. ReactJS-HOL**

**Post.js**

import React from "react";

const Post = (props) => {

return (

<div>

<h2>{props.title}</h2>

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

</div>

);

};

export default Post;

**Posts.js**

import React, { Component } from "react";

import Post from "./Post";

class Posts extends Component {

constructor(props) {

super(props);

this.state = {

posts: [],

hasError: false

};

}

loadPosts = () => {

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

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

.then((data) => this.setState({ posts: data }))

.catch((error) => {

console.error("Error loading posts:", error);

this.setState({ hasError: true });

});

};

componentDidMount() {

this.loadPosts();

}

componentDidCatch(error, info) {

alert("An error occurred in the Posts component.");

this.setState({ hasError: true });

}

render() {

const { posts, hasError } = this.state;

if (hasError) {

return <h2>Something went wrong.</h2>;

}

return (

<div>

<h1>Blog Posts</h1>

{posts.map((post) => (

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

))}

</div>

);

}

}

export default Posts;

**App.js**

import React from "react";

import "./App.css";

import Posts from "./Posts";

function App() {

return (

<div className="App">

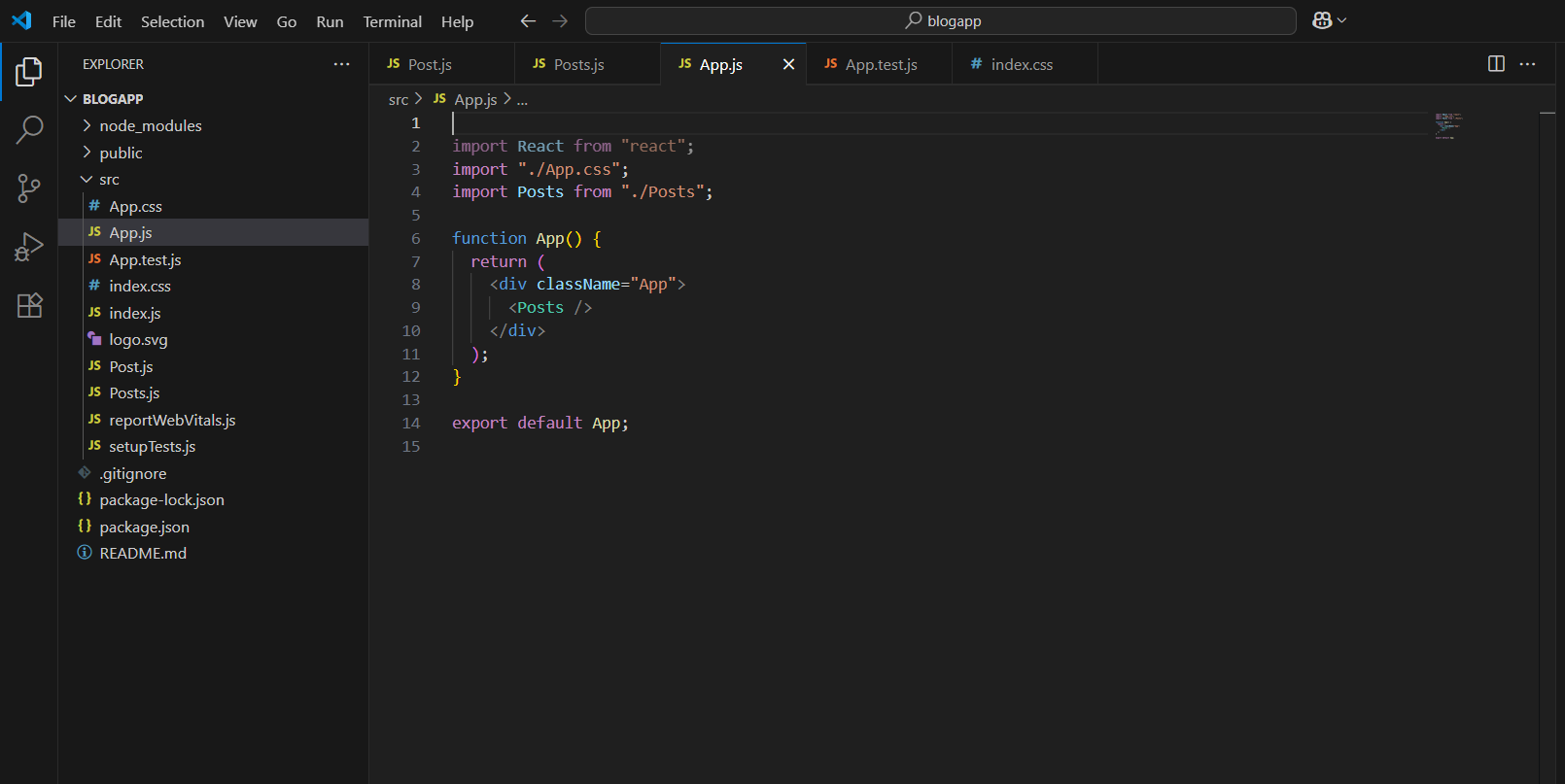
<Posts />

</div>

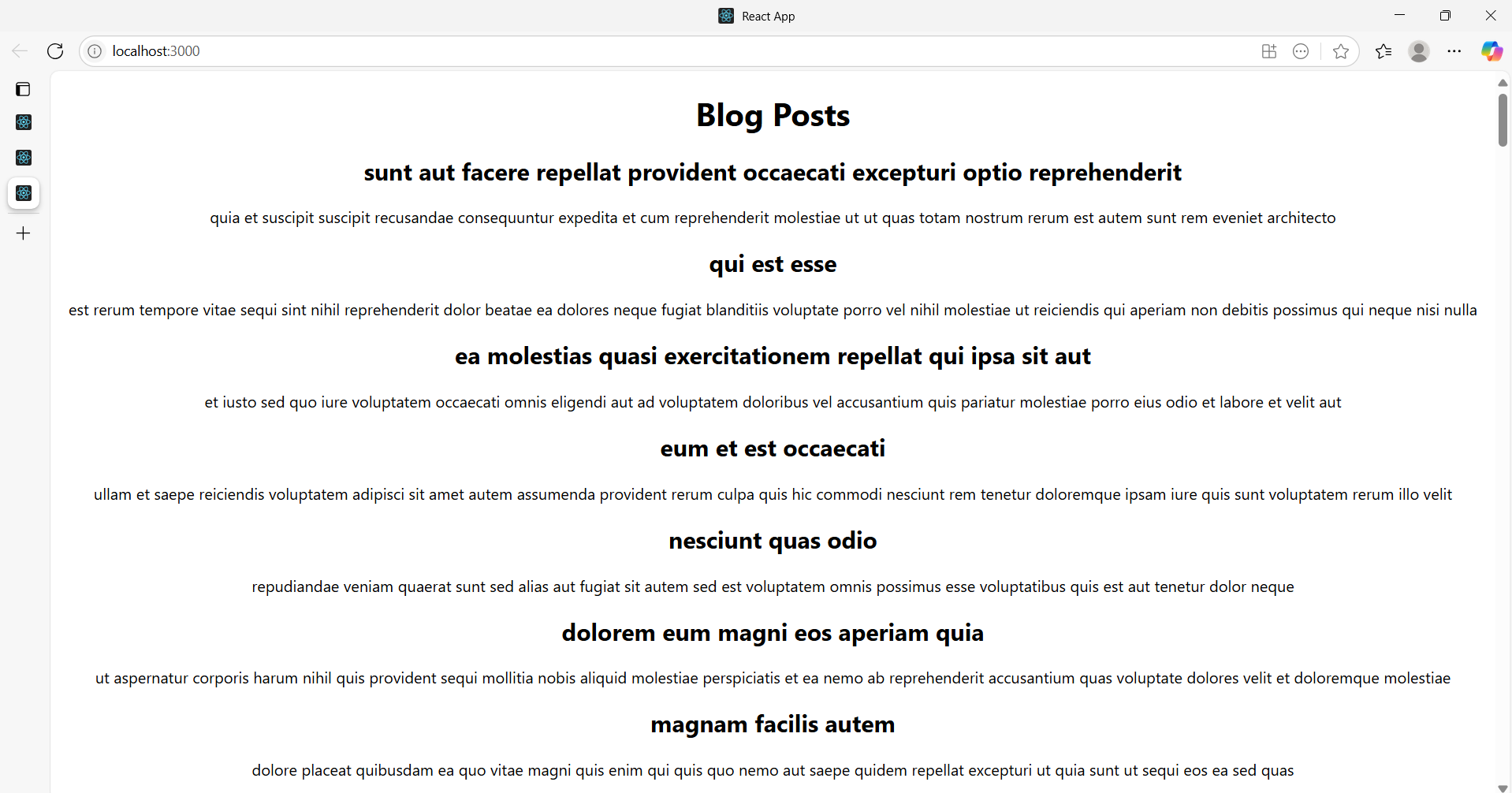
);

}

export default App;



**OUTPUT:**

****

**5. ReactJS-HOL**

**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;

}

**App.js**

import React from 'react';

import CohortDetails from './CohortDetails';

function App() {

return (

<div>

<CohortDetails

status="ongoing"

name="React Bootcamp"

startDate="2025-08-01"

endDate="2025-09-15"

/>

<CohortDetails

status="completed"

name="Java Full Stack"

startDate="2025-06-01"

endDate="2025-07-15"

/>

</div>

);

}

export default App;

**CohortDetails.js**

import React from 'react';

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

function CohortDetails({ cohort }) {

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

return (

<div className={styles.box}>

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

Cohort Status: {status}

</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;

**Cohort.js**

export const CohortsData = [

{

name: 'React Bootcamp',

startDate: '2025-08-01',

endDate: '2025-09-15',

status: 'ongoing'

},

{

name: 'Java Full Stack',

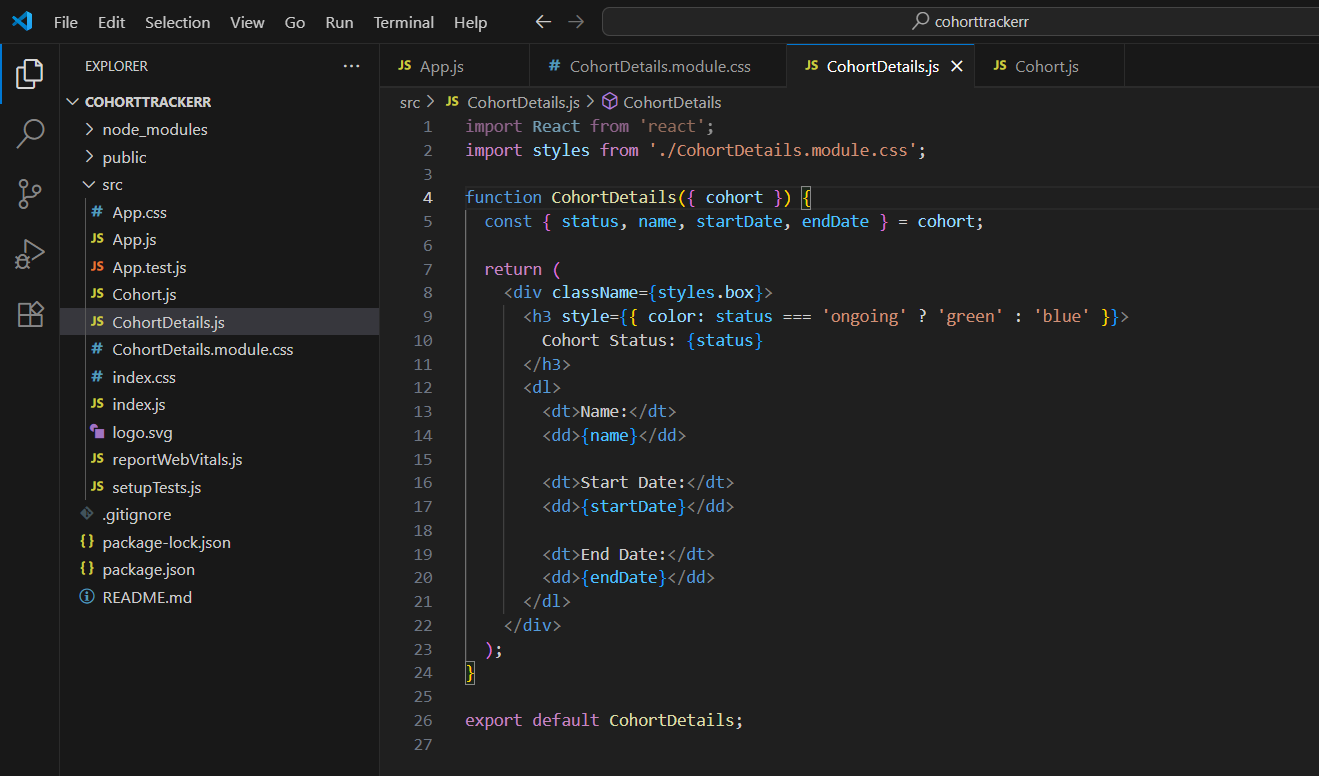
startDate: '2025-06-01',

endDate: '2025-07-15',

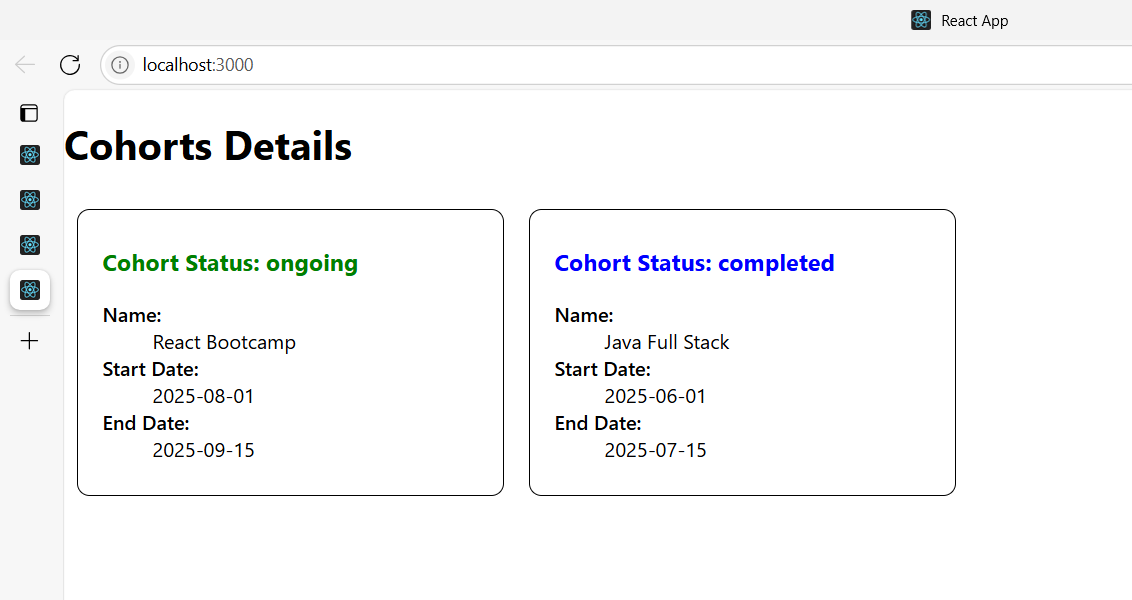
status: 'completed'

}

];



**OUTPUT:**

****