1. ReactJS-HOL

Objectives

1. **Define SPA and its benefits:**
   A Single-Page Application (SPA) is a web application that loads a single HTML page and dynamically updates that page as the user interacts with the app. This provides a more fluid and desktop-like user experience by avoiding full page reloads.

2. **Define React and identify its working:**
   React is a JavaScript library for building user interfaces. It works by creating a component-based structure where each component is a small, reusable piece of code that controls a part of the UI.

3. **Identify the differences between SPA and MPA:**
   1. SPA (Single-Page Application): Loads a single HTML page and updates content dynamically. Faster and more responsive user experience after the initial load.
   2. MPA (Multi-Page Application): Each user action (e.g., clicking a link) triggers a full page reload from the server. This can be slower and less fluid.

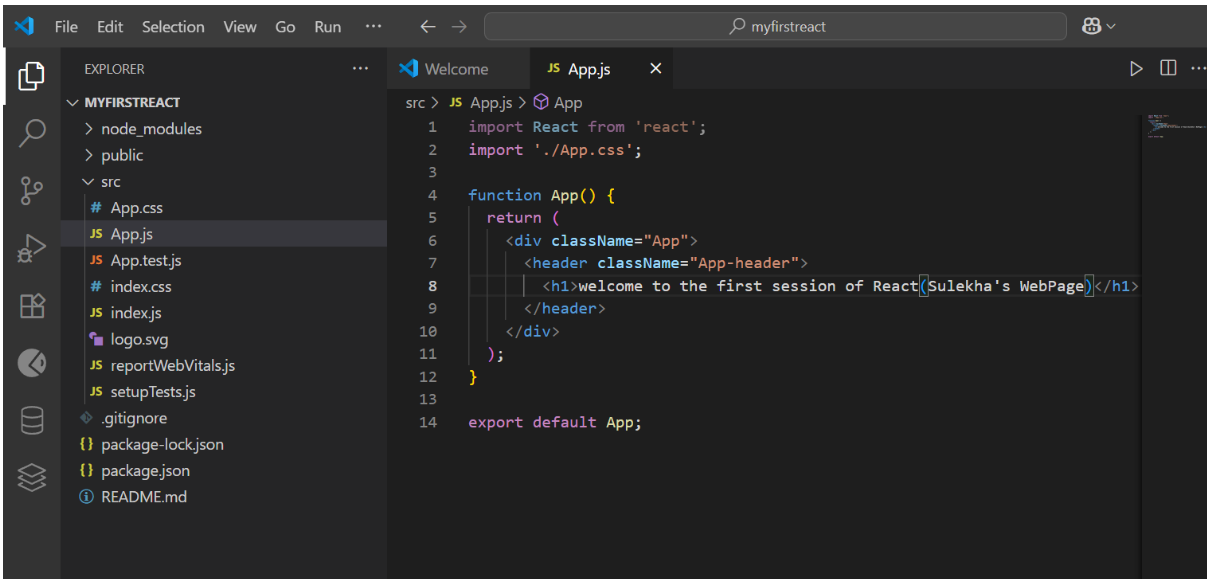
4. **Explain Pros & Cons of Single-Page Application:**
   1. Pros: Improved user experience, faster performance (after initial load), and easier debugging with modern browsers.
   2. Cons: Slower initial load time, SEO challenges (can be mitigated), and requires JavaScript to be enabled.

5. **Explain about React:**
   React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets you compose complex UIs from small and isolated pieces of code called "components."

6. **Explain Features of React:** Component-based architecture, declarative views, JSX syntax, Virtual DOM, and a one-way data flow.

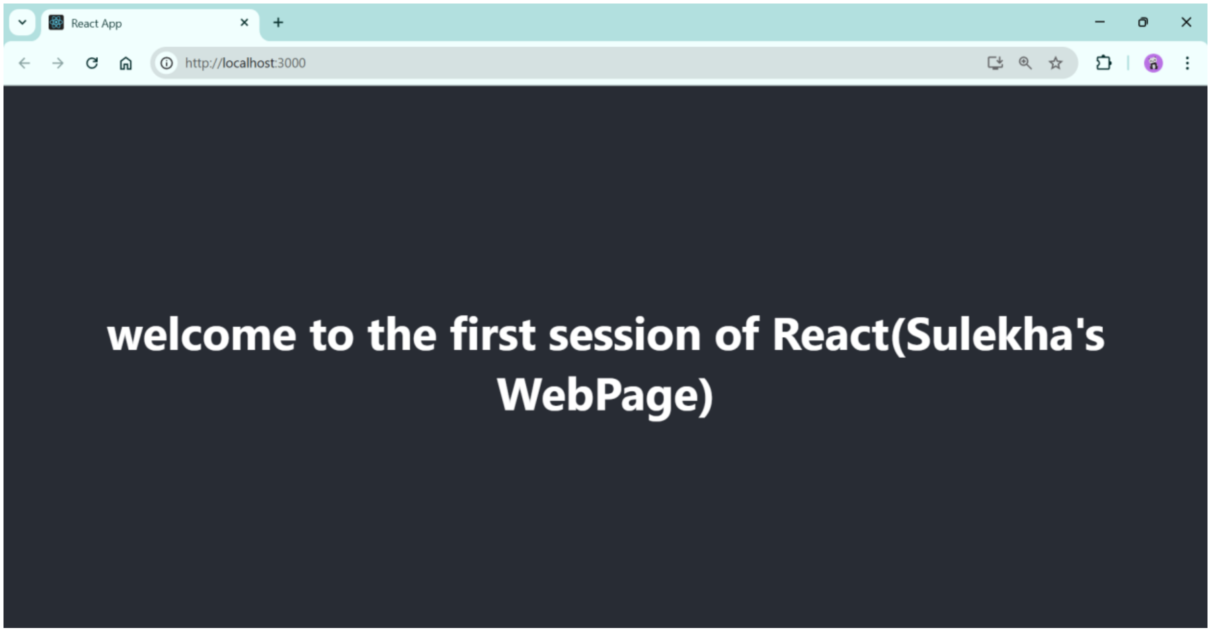
7. **Define Virtual DOM:**
The Virtual DOM is a lightweight copy of the real DOM. When the state of a React component changes, React first updates the Virtual DOM, then efficiently calculates the minimum number of changes needed to update the real DOM. This process, known as "reconciliation," significantly improves performance.

Code:



```
import React from 'react';
import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <h1>welcome to the first session of React(Sulekha's WebPage)</h1>
      </header>
    </div>
  );
}

export default App;
```

Output



welcome to the first session of React(Sulekha's WebPage)

2. ReactJS-HOL

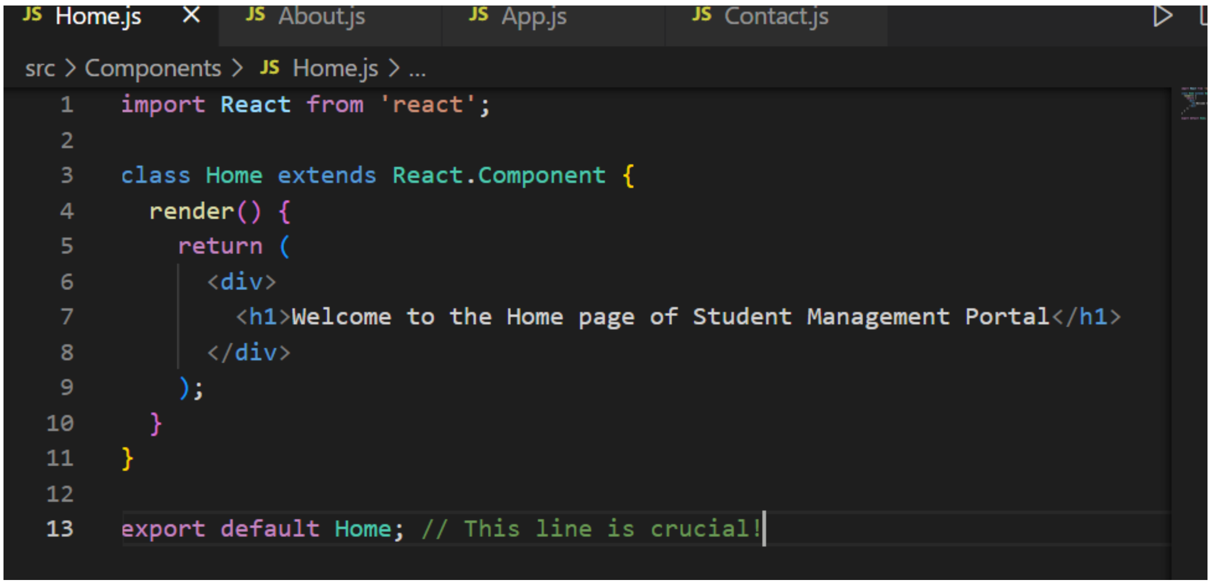
Objectives

1. **Explain React components:** A React component is an independent, reusable piece of code that represents a part of a user interface (UI).
2. **Identify the differences between components and JavaScript functions:** React components are specialized functions or classes that take properties (props) as input and return React elements that describe what should appear on the screen. Standard JavaScript functions perform specific tasks and return an output.
3. **Identify the types of components:** The two main types of components in React are Class Components and Function Components.
4. **Explain class component:** Class components are defined using ES6 class syntax, extend React.Component, and have a render() method that returns JSX. They can manage their own state and lifecycle methods.
5. **Explain function component:** Function components are defined using a JavaScript function and are simpler to write. With React Hooks, they can also manage state and handle lifecycle events.
6. **Define component constructor:** For class components, the constructor() is a special method called when a component is created, used to initialize state and bind event handlers. It's crucial to call
7. **Define render() function:** The render() method is the only mandatory method in a class component, responsible for returning the JSX that describes what the component should render. It should be a pure function.

Student Management Portal

Code:

Home.js

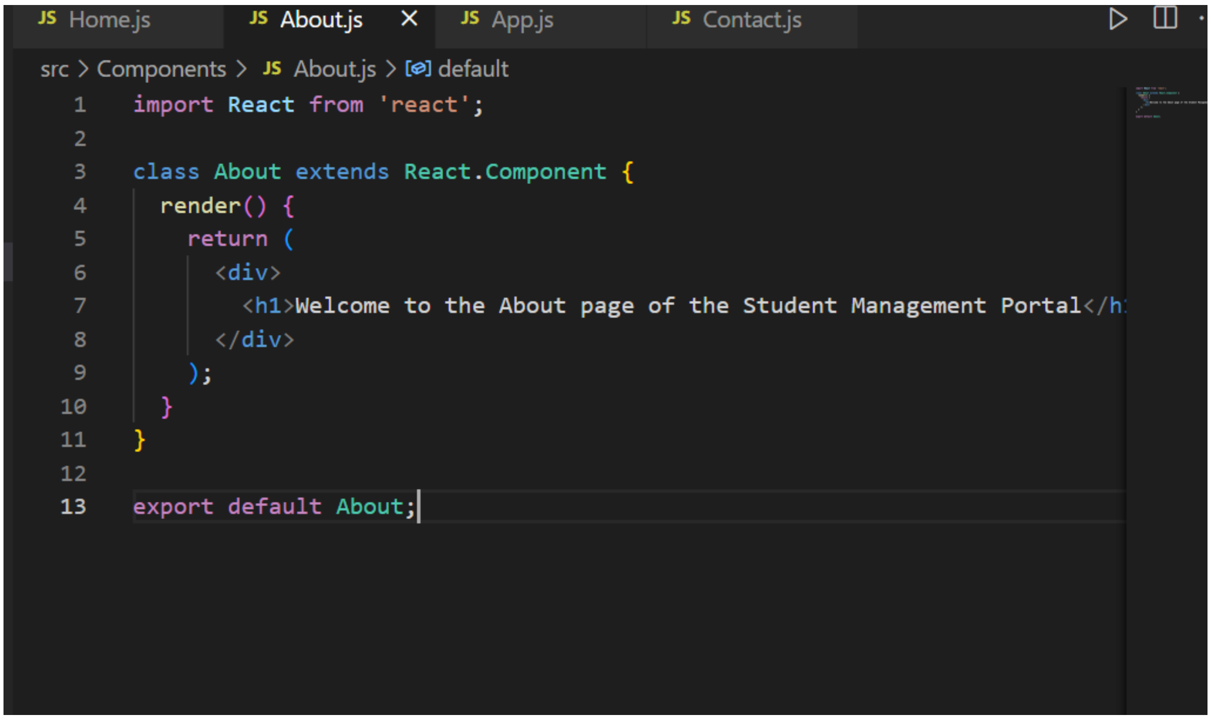


Home.js × | About.js | App.js | Contact.js

src > Components > Home.js > ...

```
import React from 'react';

class Home extends React.Component {
  render() {
    return (
      <div>
        <h1>Welcome to the Home page of Student Management Portal</h1>
      </div>
    );
  }
}

export default Home; // This line is crucial!
```

About.js

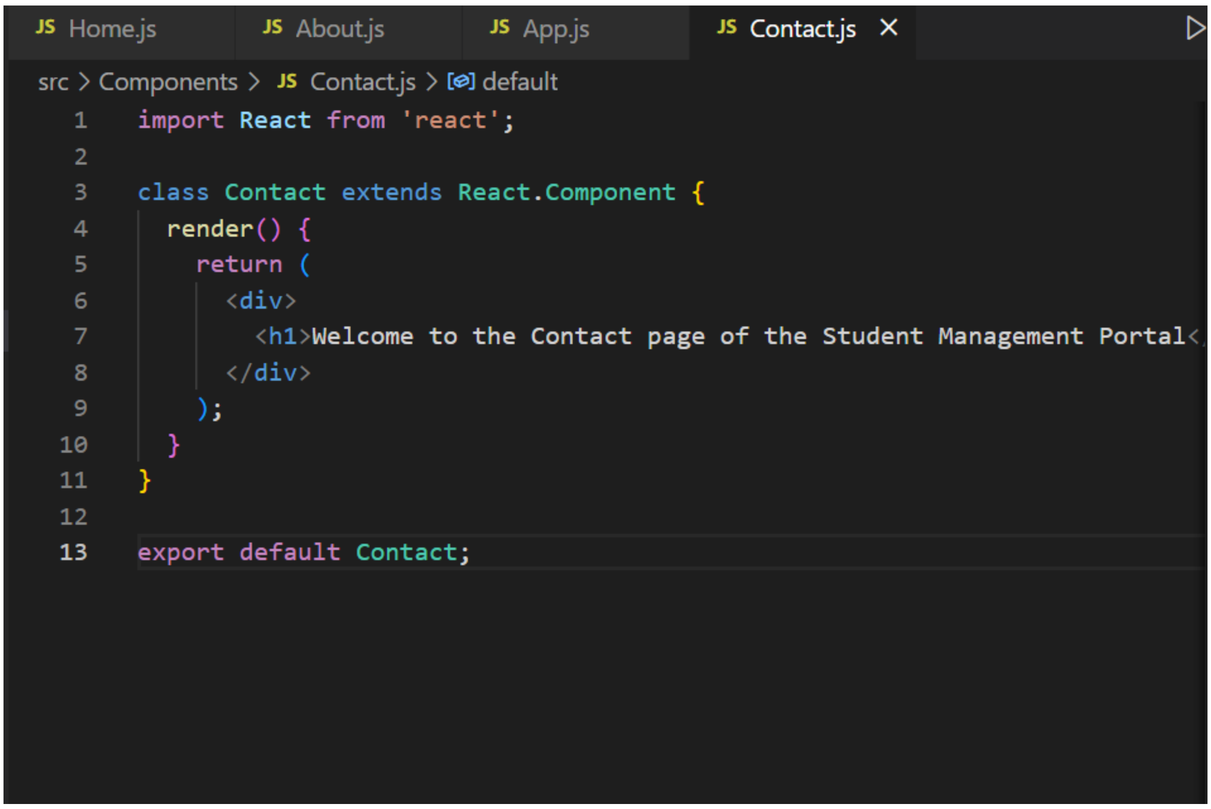


Home.js | About.js × | App.js | Contact.js

src > Components > About.js > default

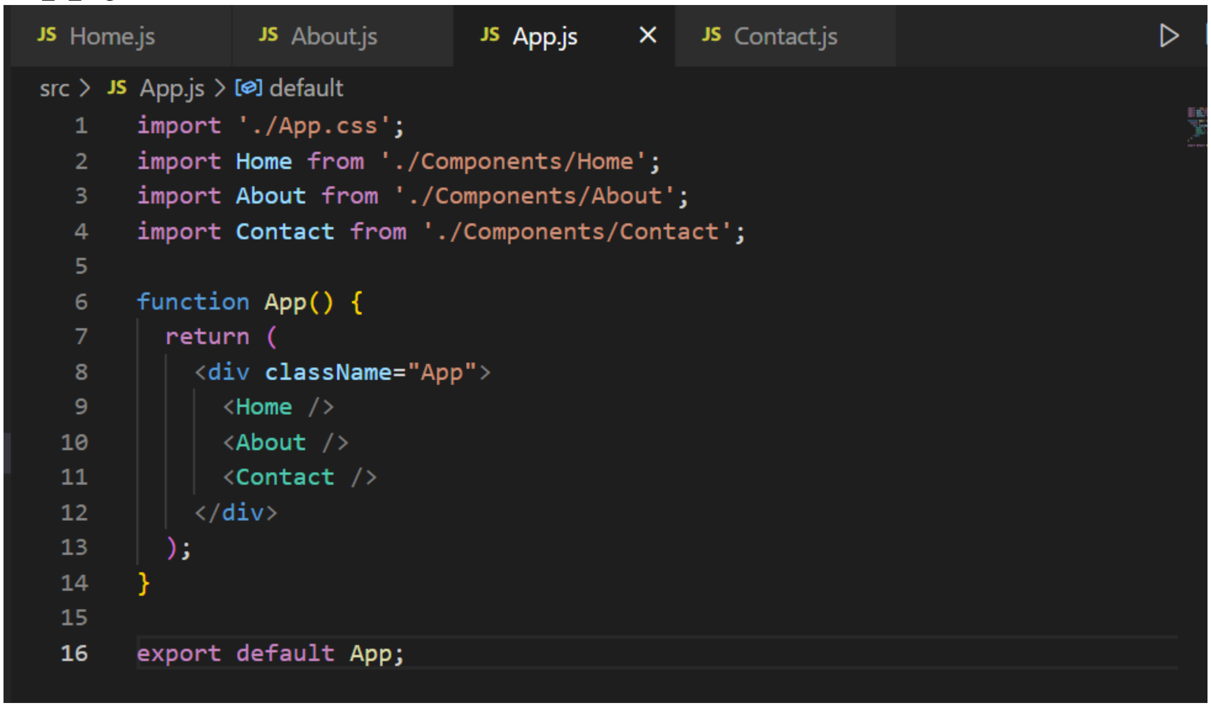
```
import React from 'react';

class About extends React.Component {
  render() {
    return (
      <div>
        <h1>Welcome to the About page of the Student Management Portal</h
      </div>
    );
  }
}

export default About;
```

Contact.js



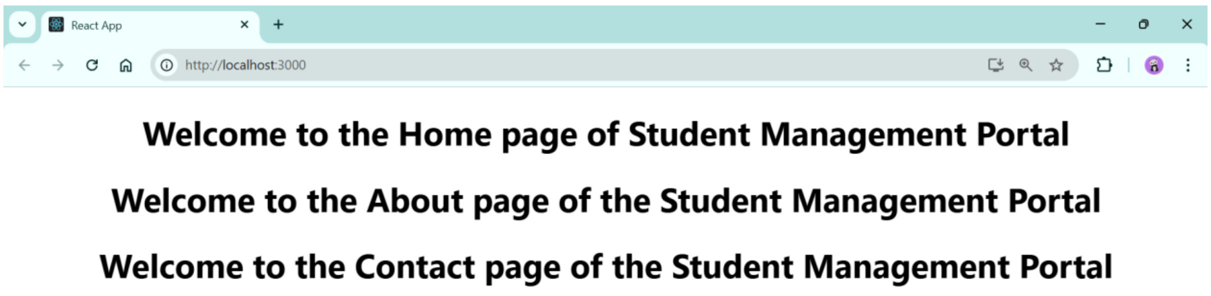
```
import React from 'react';

class Contact extends React.Component {
  render() {
    return (
      <div>
        <h1>Welcome to the Contact page of the Student Management Portal<
      </div>
    );
  }
}

export default Contact;
```

App.js



```
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;
```

Output



```
Welcome to the Home page of Student Management Portal
Welcome to the About page of the Student Management Portal
Welcome to the Contact page of the Student Management Portal
```

3. ReactJS-HOL

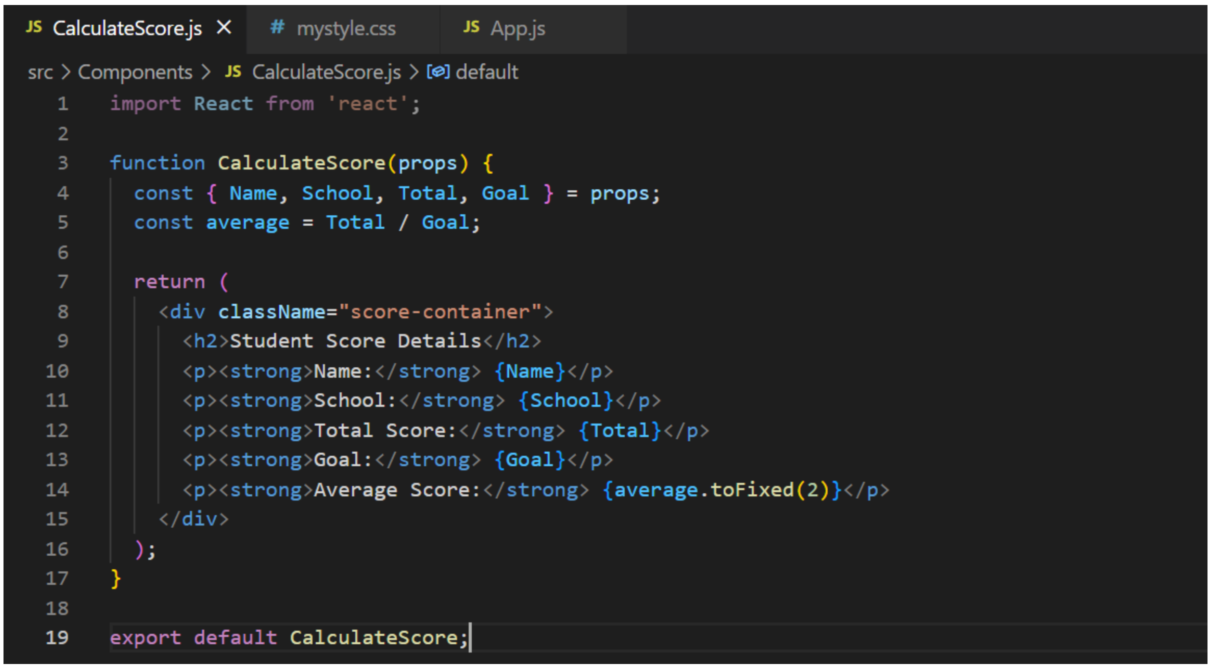
Objectives

1. **Explain React components**: Independent, reusable pieces of UI code.
2. **Identify the differences between components and JavaScript functions**: React components are specialized functions or classes that take props and return React elements (UI), while JavaScript functions perform tasks and return outputs.
3. **Identify the types of components**: The two main types are Class Components and Function Components.
4. **Explain class component**: Defined using ES6 class syntax, extending React.Component, and including a render() method. Can manage state and lifecycle methods.
5. **Explain function component**: Defined using a JavaScript function. Simpler to write and, with Hooks, can manage state and lifecycle events.
6. **Define component constructor**: (Relevant to Class Components) A special method called upon component creation, used for state initialization and binding methods.
7. **Define render() function**: The mandatory method in a class component, responsible for returning the JSX that describes the UI. Should be a pure function.

Student Management Portal to Calculate Score

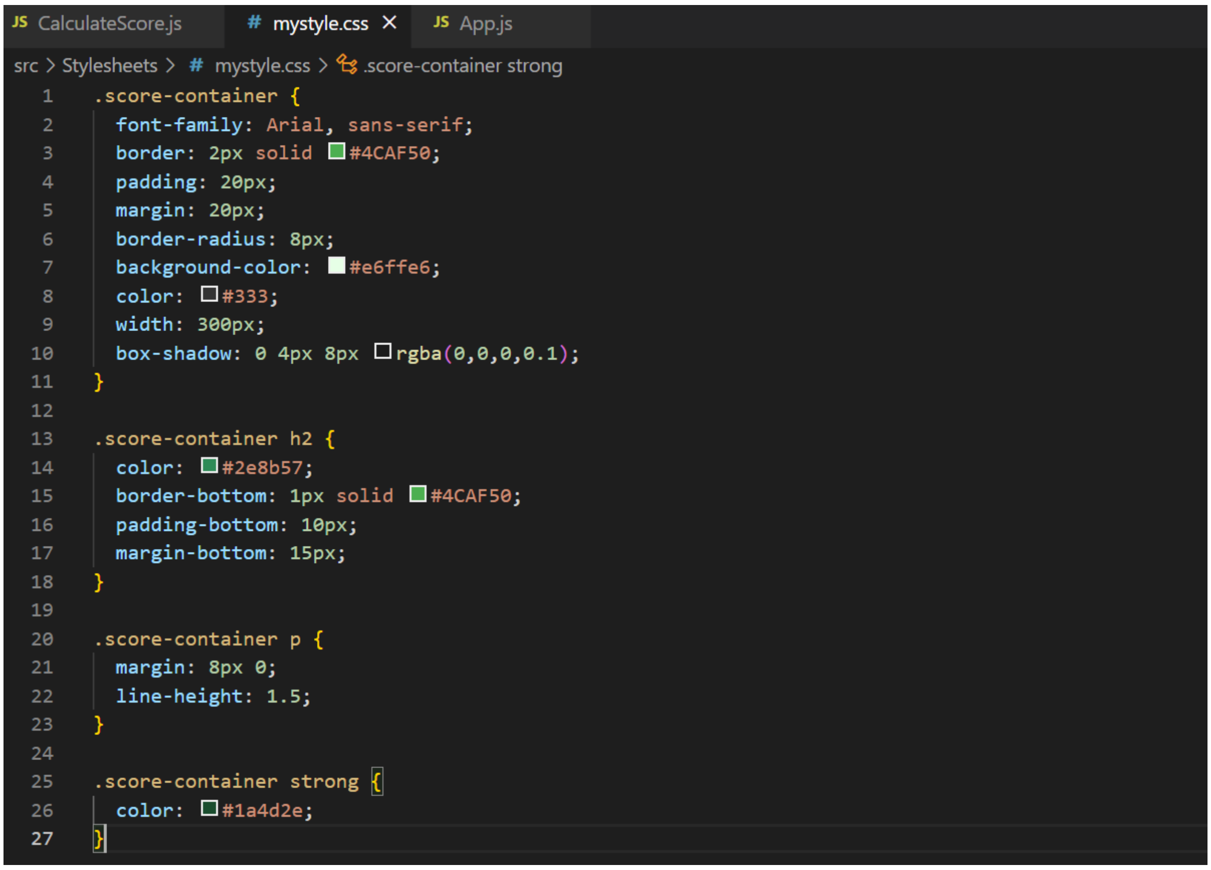
Code:

CalculateScore.js



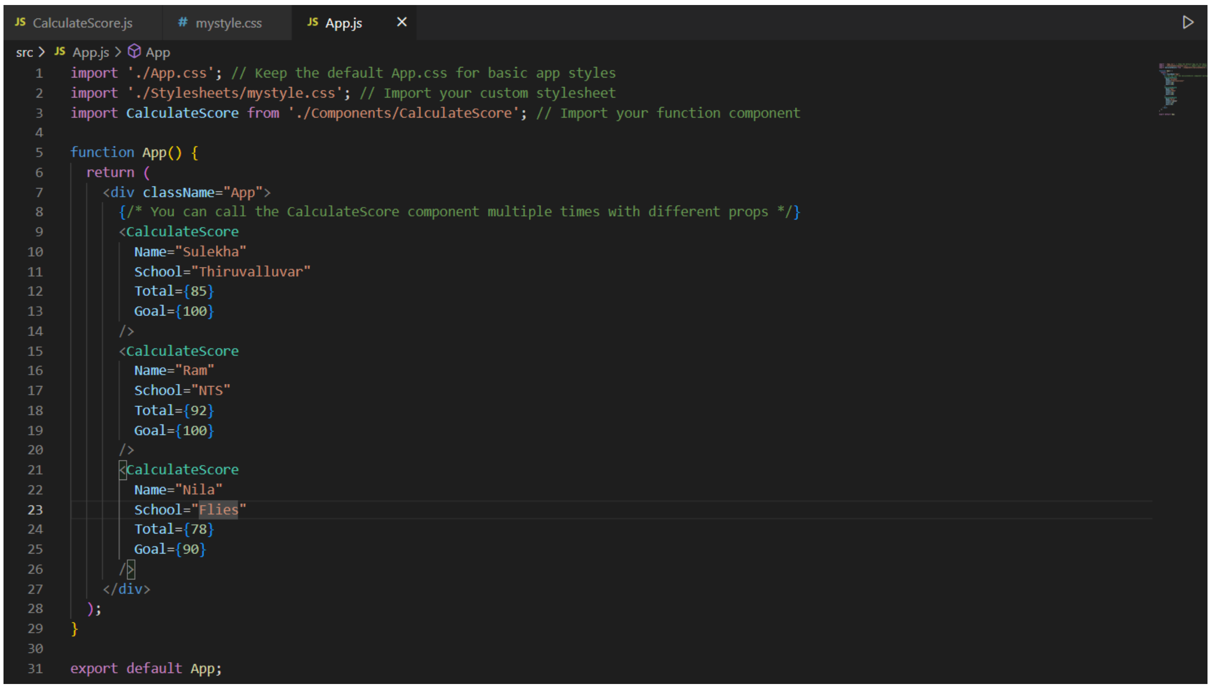
```
import React from 'react';

function CalculateScore(props) {
  const { Name, School, Total, Goal } = props;
  const average = Total / Goal;

  return (
    <div className="score-container">
      <h2>Student Score Details</h2>
      <p><strong>Name:</strong> {Name}</p>
      <p><strong>School:</strong> {School}</p>
      <p><strong>Total Score:</strong> {Total}</p>
      <p><strong>Goal:</strong> {Goal}</p>
      <p><strong>Average Score:</strong> {average.toFixed(2)}</p>
    </div>
  );
}

export default CalculateScore;
```

MyStyle.css



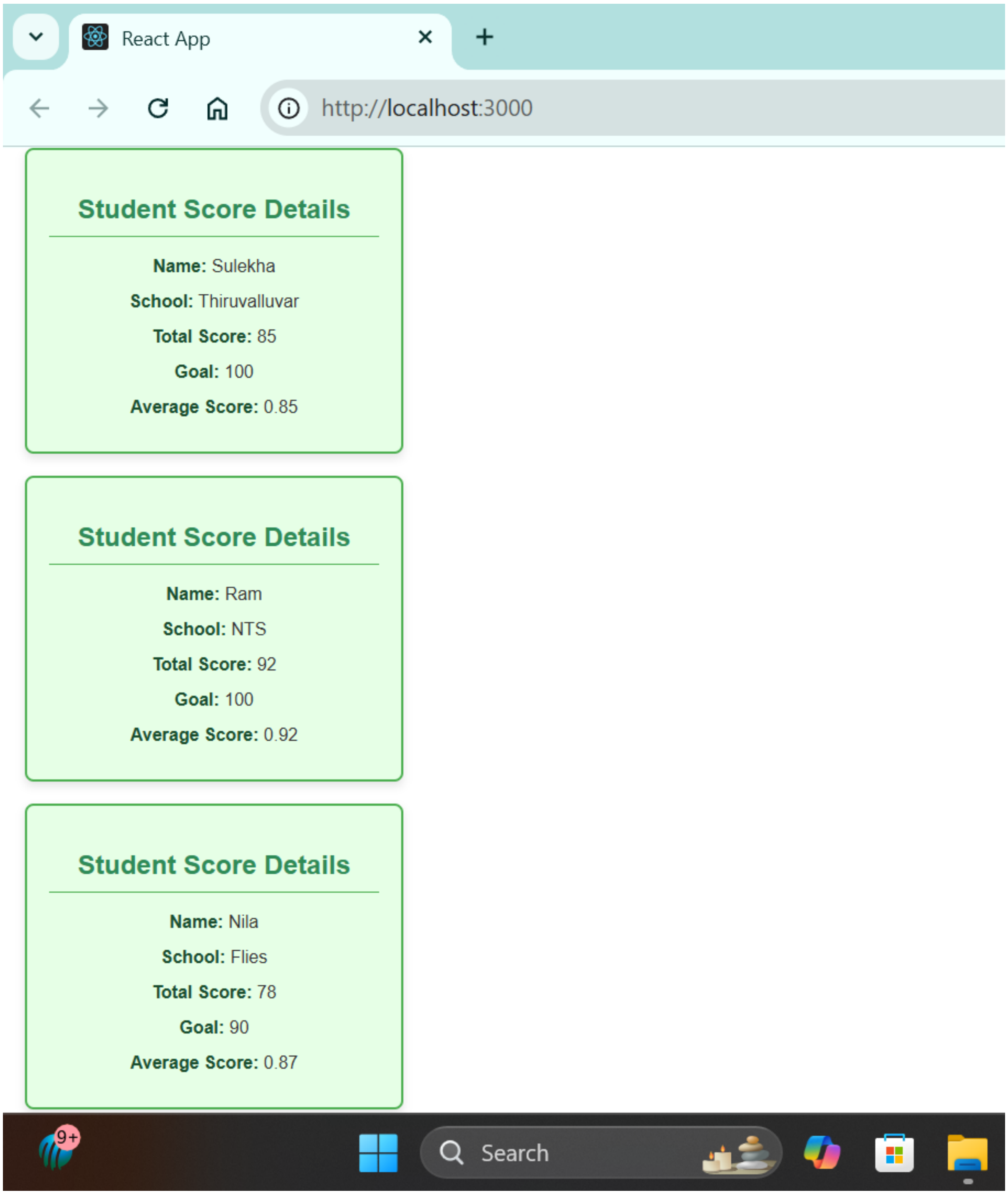
```css
.score-container {
  font-family: Arial, sans-serif;
  border: 2px solid #4CAF50;
  padding: 20px;
  margin: 20px;
  border-radius: 8px;
  background-color: #e6ffe6;
  color: #333;
  width: 300px;
  box-shadow: 0 4px 8px rgba(0,0,0,0.1);
}

.score-container h2 {
  color: #2e8b57;
  border-bottom: 1px solid #4CAF50;
  padding-bottom: 10px;
  margin-bottom: 15px;
}

.score-container p {
  margin: 8px 0;
  line-height: 1.5;
}

.score-container strong {
  color: #1a4d2e;
}
```

App.js



```javascript
import './App.css'; // Keep the default App.css for basic app styles
import './Stylesheets/mystyle.css'; // Import your custom stylesheet
import CalculateScore from './Components/CalculateScore'; // Import your function component

function App() {
  return (
    <div className="App">
      {/* You can call the CalculateScore component multiple times with different props */}
      <CalculateScore
        Name="Sulekha"
        School="Thiruvalluvar"
        Total={85}
        Goal={100}
      />
      <CalculateScore
        Name="Ram"
        School="NTS"
        Total={92}
        Goal={100}
      />
      <CalculateScore
        Name="Nila"
        School="Flies"
        Total={78}
        Goal={90}
      />
    </div>
  );
}

export default App;
```

Output:



Student Score Details

Name: Sulekha

School: Thiruvalluvar

Total Score: 85

Goal: 100

Average Score: 0.85

Student Score Details

Name: Ram

School: NTS

Total Score: 92

Goal: 100

Average Score: 0.92

Student Score Details

Name: Nila

School: Flies

Total Score: 78

Goal: 90

Average Score: 0.87

4. ReactJS-HOL

Objectives

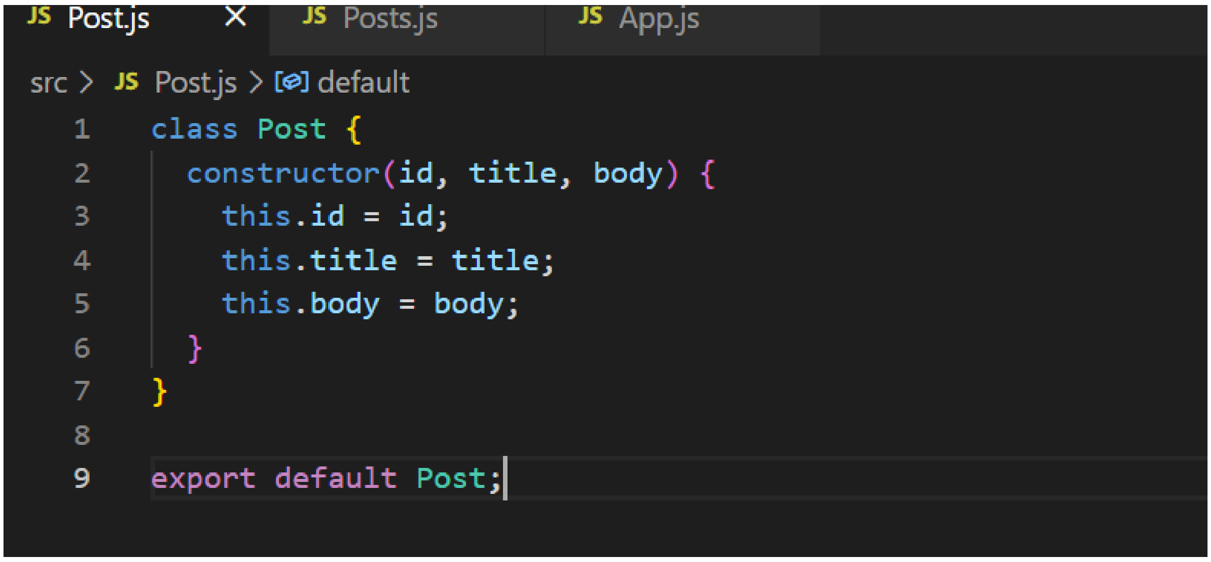
8. **Explain the need and Benefits of component lifecycle**: The component lifecycle refers to the various phases a component goes through from its creation to its destruction. Lifecycle methods (hooks) allow developers to execute code at specific points in a component's life, enabling control over rendering, data fetching, and interaction with the DOM.

9. **Identify various life cycle hook methods**: Key lifecycle methods include constructor(), render(), componentDidMount(), componentDidUpdate(), componentWillUnmount(), and error handling methods like componentDidCatch().

10. **List the sequence of steps in rendering a component**:
    1. **Mounting (Initial Render)**: constructor() -> render() -> React updates DOM -> componentDidMount().
    2. **Updating (Re-renders)**: render() -> React updates DOM -> componentDidUpdate().
    3. **Unmounting (Removal)**: componentWillUnmount().
    4. **Error Handling**: componentDidCatch().

Blog Application

Post.js

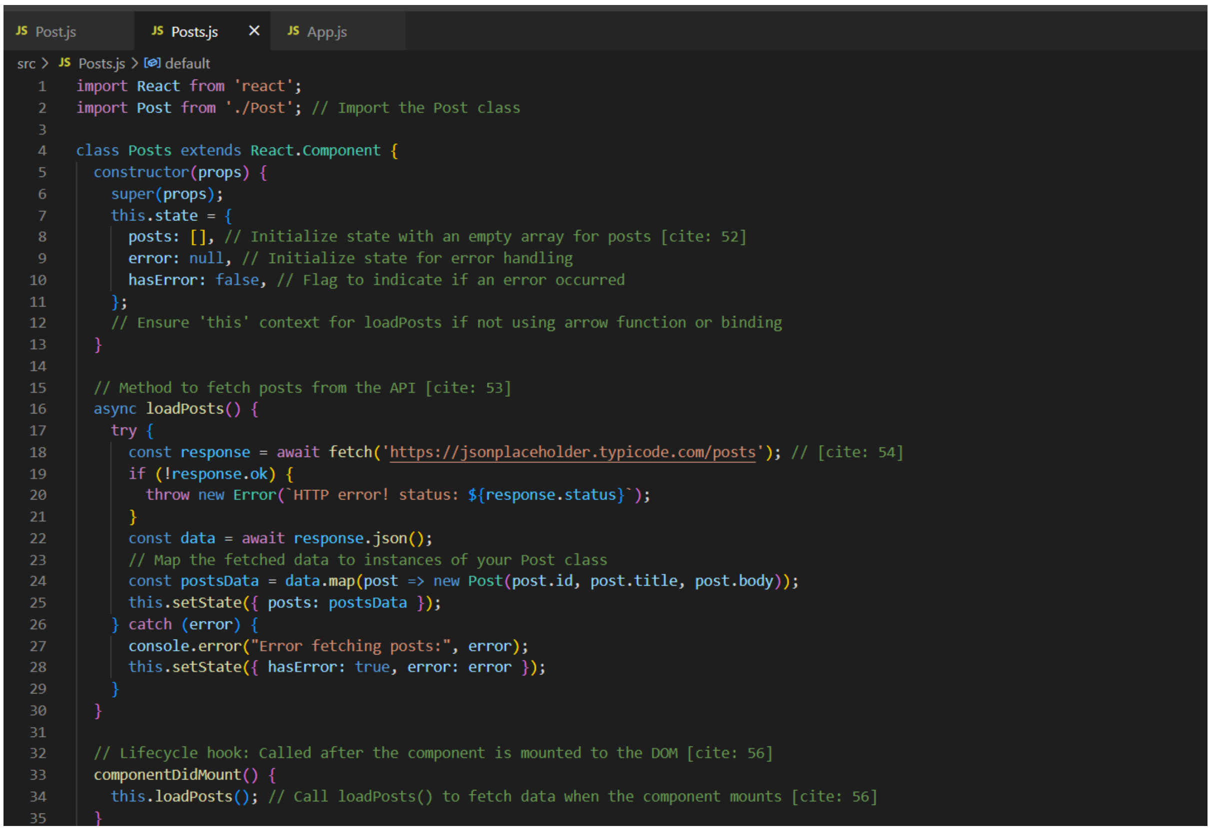


Post.js × | Posts.js | App.js

src > Post.js > default

```javascript
class Post {
  constructor(id, title, body) {
    this.id = id;
    this.title = title;
    this.body = body;
  }
}

export default Post;
```

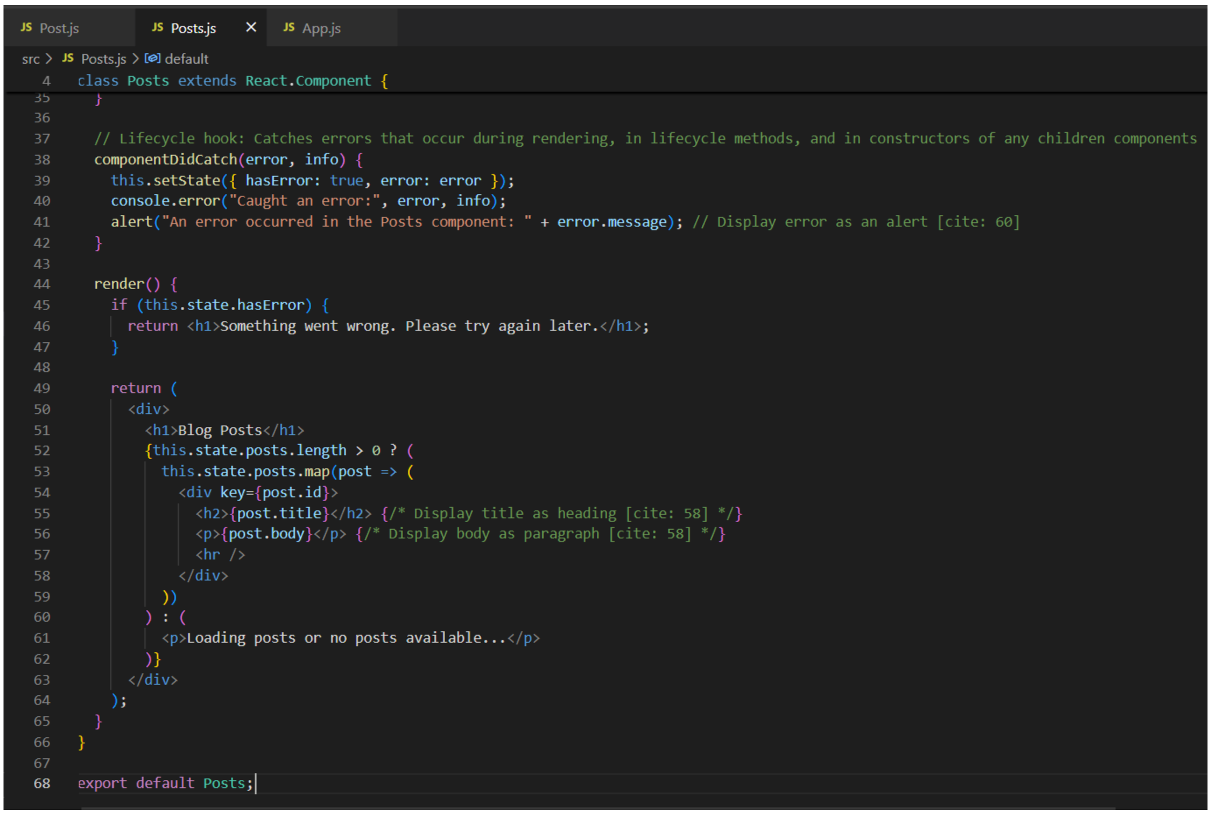
Posts.js



Post.js | Posts.js × | App.js

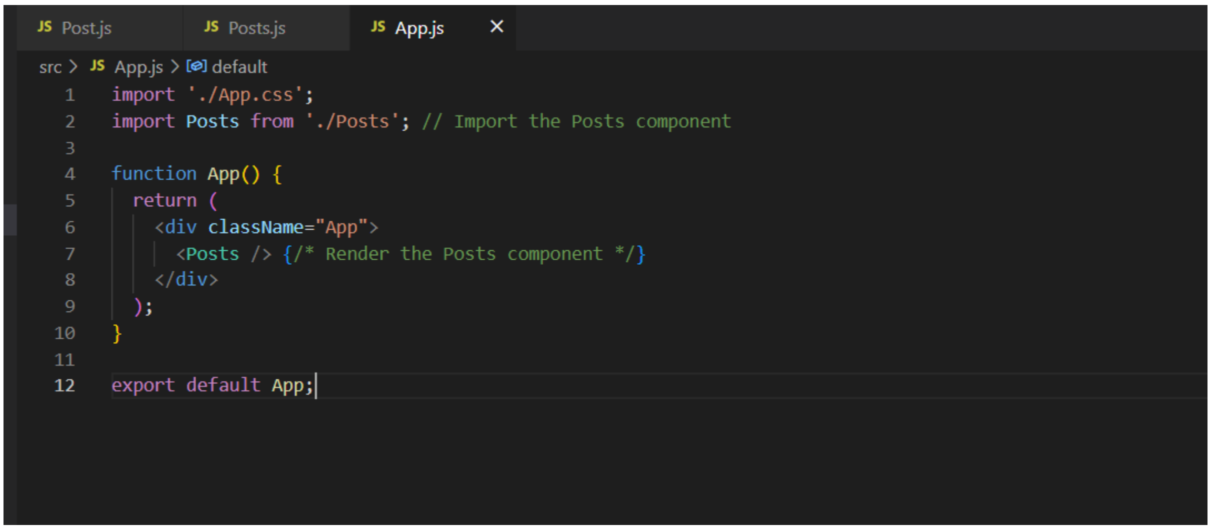
src > Posts.js > default

```javascript
import React from 'react';
import Post from './Post'; // Import the Post class

class Posts extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      posts: [], // Initialize state with an empty array for posts [cite: 52]
      error: null, // Initialize state for error handling
      hasError: false, // Flag to indicate if an error occurred
    };
    // Ensure 'this' context for loadPosts if not using arrow function or binding
  }

  // Method to fetch posts from the API [cite: 53]
  async loadPosts() {
    try {
      const response = await fetch('https://jsonplaceholder.typicode.com/posts'); // [cite: 54]
      if (!response.ok) {
        throw new Error(`HTTP error! status: ${response.status}`);
      }
      const data = await response.json();
      // Map the fetched data to instances of your Post class
      const postsData = data.map(post => new Post(post.id, post.title, post.body));
      this.setState({ posts: postsData });
    } catch (error) {
      console.error("Error fetching posts:", error);
      this.setState({ hasError: true, error: error });
    }
  }

  // Lifecycle hook: Called after the component is mounted to the DOM [cite: 56]
  componentDidMount() {
    this.loadPosts(); // Call loadPosts() to fetch data when the component mounts [cite: 56]
  }
```



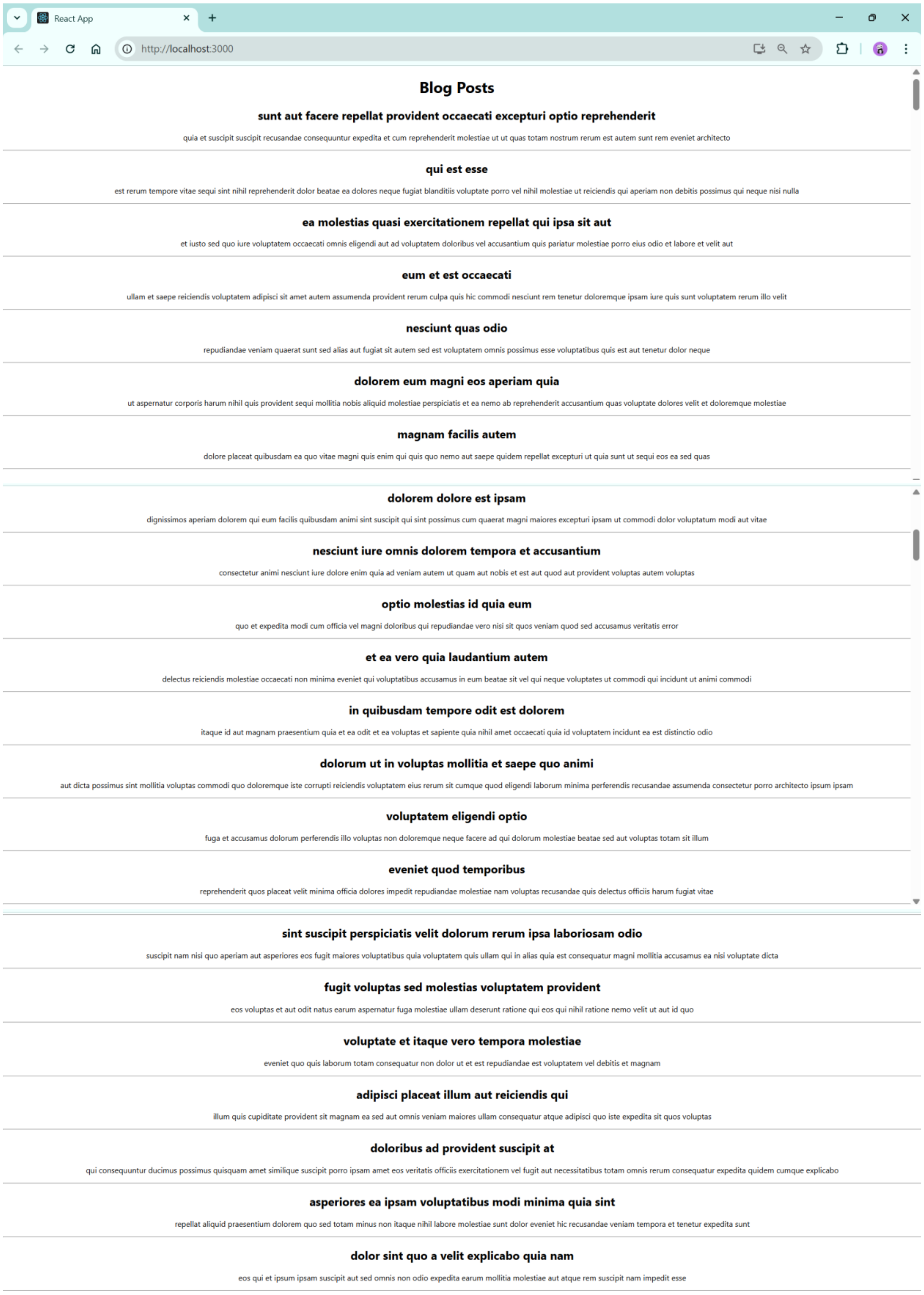
```
class Posts extends React.Component {
  }

  // Lifecycle hook: Catches errors that occur during rendering, in lifecycle methods, and in constructors of any children components
  componentDidCatch(error, info) {
    this.setState({ hasError: true, error: error });
    console.error("Caught an error:", error, info);
    alert("An error occurred in the Posts component: " + error.message); // Display error as an alert [cite: 60]
  }

  render() {
    if (this.state.hasError) {
      return <h1>Something went wrong. Please try again later.</h1>;
    }

    return (
      <div>
        <h1>Blog Posts</h1>
        {this.state.posts.length > 0 ? (
          this.state.posts.map(post => (
            <div key={post.id}>
              <h2>{post.title}</h2> {/* Display title as heading [cite: 58] */}
              <p>{post.body}</p> {/* Display body as paragraph [cite: 58] */}
              <hr />
            </div>
          ))
        ) : (
          <p>Loading posts or no posts available...</p>
        )}
      </div>
    );
  }
}

export default Posts;
```

App.js



```
import './App.css';
import Posts from './Posts'; // Import the Posts component

function App() {
  return (
    <div className="App">
      <Posts /> {/* Render the Posts component */}
    </div>
  );
}

export default App;
```

Output:



Blog Posts

sunt aut facere repellat provident occaecati excepturi optio reprehenderit

quia et suscipit suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut quas totam nostrum rerum est autem sunt rem eveniet architecto

qui est esse

est rerum tempore vitae sequi sint nihil reprehenderit dolor beatae ea dolores neque fugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis qui aperiam non debitis possimus qui neque nisi nulla

ea molestias quasi exercitationem repellat qui ipsa sit aut

et iusto sed quo iure voluptatem occaecati omnis eligendi aut ad voluptatem doloribus vel accusantium quis pariatur molestiae porro eius odio et labore et velit aut

eum et est occaecati

ullam et saepe reiciendis voluptatem adipisci sit amet autem assumenda provident rerum culpa quis hic commodi nesciunt rem tenetur doloremque ipsam iure quis sunt voluptatem rerum illo velit

nesciunt quas odio

repudiandae veniam quaerat sunt sed alias aut fugiat sit autem sed est voluptatem omnis possimus esse voluptatibus quis est aut tenetur dolor neque

dolorem eum magni eos aperiam quia

ut aspernatur corporis harum nihil quis provident sequi mollitia nobis aliquid molestiae perspiciatis et ea nemo ab reprehenderit accusantium quas voluptate dolores velit et doloremque molestiae

magnam facilis autem

dolore placeat quibusdam ea quo vitae magni quis enim qui quis quo nemo aut saepe quidem repellat excepturi ut quia sunt ut sequi eos ea sed quas

dolorem dolore est ipsam

dignissimos aperiam dolorem qui eum facilis quibusdam animi sint suscipit qui sint possimus cum quaerat magni maiores excepturi ipsam ut commodi dolor voluptatum modi aut vitae

nesciunt iure omnis dolorem tempora et accusantium

consectetur animi nesciunt iure dolore enim quia ad veniam autem ut quam aut nobis et est aut quod aut provident voluptas autem voluptas

optio molestias id quia eum

quo et expedita modi cum officia vel magni doloribus qui repudiandae vero nisi sit quos veniam quod sed accusamus veritatis error

et ea vero quia laudantium autem

delectus reiciendis molestiae occaecati non minima eveniet qui voluptatibus accusamus in eum beatae sit vel qui neque voluptates ut commodi qui incidunt ut animi commodi

in quibusdam tempore odit est dolorem

itaque id aut magnam praesentium quia et ea odit et ea voluptas et sapiente quia nihil amet occaecati quia id voluptatem incidunt ea est distinctio odio

dolorum ut in voluptas mollitia et saepe quo animi

aut dicta possimus sint mollitia voluptas commodi quo doloremque iste corrupti reiciendis voluptatem eius rerum sit cumque quod eligendi laborum minima perferendis recusandae assumenda consectetur porro architecto ipsum ipsam

voluptatem eligendi optio

fuga et accusamus dolorum perferendis illo voluptas non doloremque neque facere ad qui dolorum molestiae beatae sed aut voluptas totam sit illum

eveniet quod temporibus

reprehenderit quos placeat velit minima officia dolores impedit repudiandae molestiae nam voluptas recusandae quis delectus officiis harum fugiat vitae

sint suscipit perspiciatis velit dolorum rerum ipsa laboriosam odio

suscipit nam nisi quo aperiam aut asperiores eos fugit maiores voluptatibus quia voluptatem quis ullam qui in alias quia est consequatur magni mollitia accusamus ea nisi voluptate dicta

fugit voluptas sed molestias voluptatem provident

eos voluptas et aut odit natus earum aspernatur fuga molestiae ullam deserunt ratione qui eos qui nihil ratione nemo velit ut aut id quo

voluptate et itaque vero tempora molestiae

eveniet quo quis laborum totam consequatur non dolor ut et est repudiandae est voluptatem vel debitis et magnam

adipisci placeat illum aut reiciendis qui

illum quis cupiditate provident sit magnam ea sed aut omnis veniam maiores ullam consequatur atque adipisci quo iste expedita sit quos voluptas

doloribus ad provident suscipit at

qui consequuntur ducimus possimus quisquam amet similique suscipit porro ipsam amet eos veritatis officiis exercitationem vel fugit aut necessitatibus totam omnis rerum consequatur expedita quidem cumque explicabo

asperiores ea ipsam voluptatibus modi minima quia sint

repellat aliquid praesentium dolorem quo sed totam minus non itaque nihil labore molestiae sunt dolor eveniet hic recusandae veniam tempora et tenetur expedita sunt

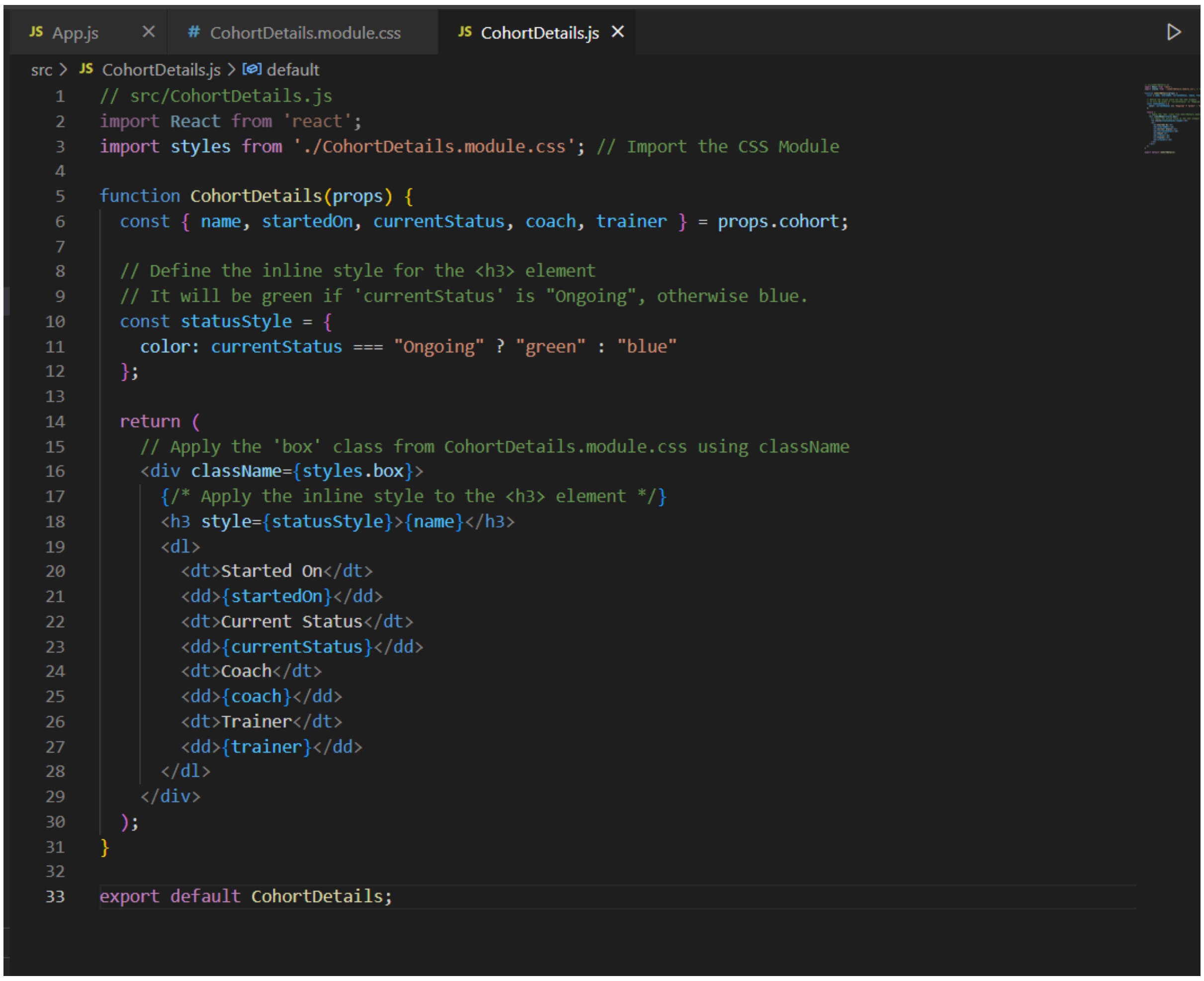
dolor sint quo a velit explicabo quia nam

eos qui et ipsum ipsam suscipit aut sed omnis non odio expedita earum mollitia molestiae aut atque rem suscipit nam impedit esse

5.ReactJS-HOL

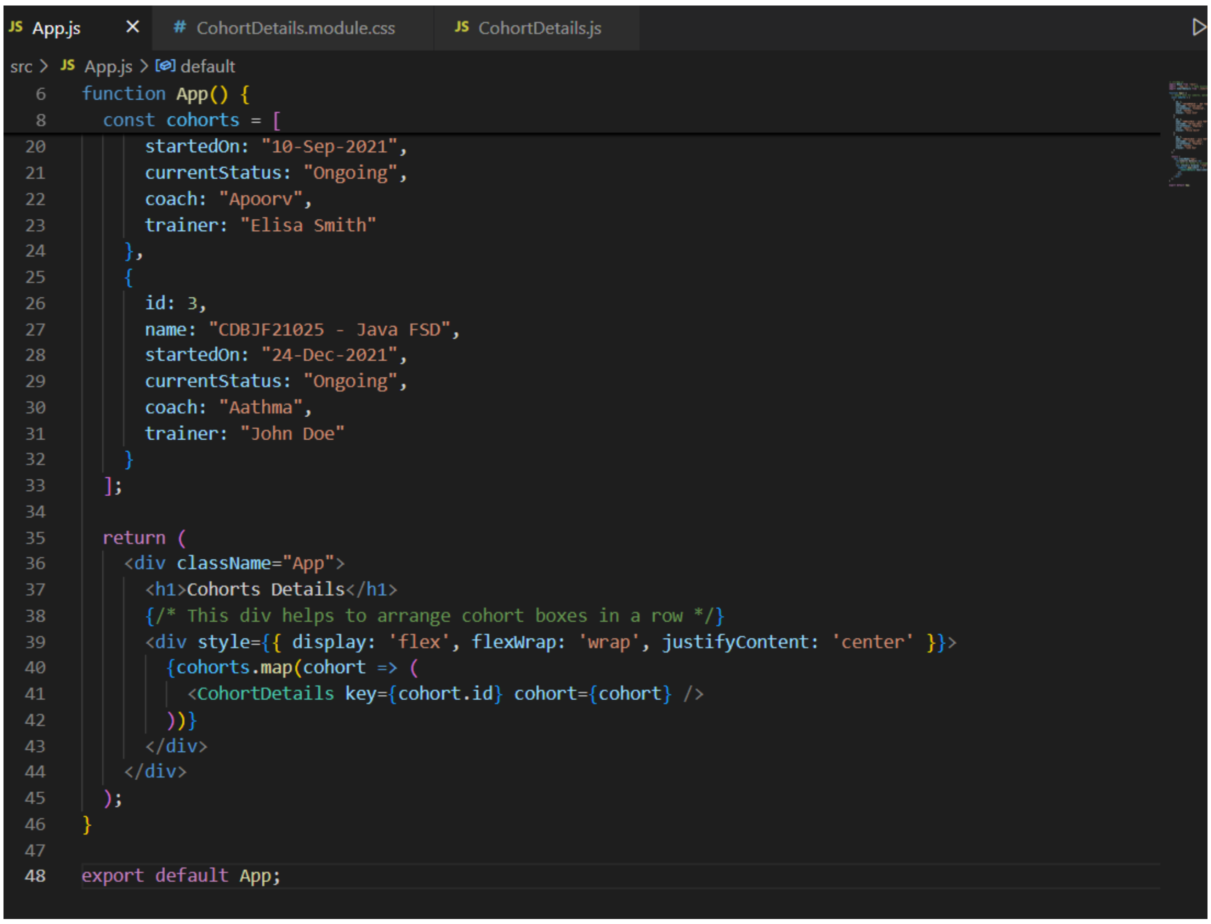
Code:

CohortDetails.js



```
// src/CohortDetails.js
import React from 'react';
import styles from './CohortDetails.module.css'; // Import the CSS Module

function CohortDetails(props) {
  const { name, startedOn, currentStatus, coach, trainer } = props.cohort;

  // Define the inline style for the <h3> element
  // It will be green if 'currentStatus' is "Ongoing", otherwise blue.
  const statusStyle = {
    color: currentStatus === "Ongoing" ? "green" : "blue"
  };

  return (
    // Apply the 'box' class from CohortDetails.module.css using className
    <div className={styles.box}>
      {/* Apply the inline style to the <h3> element */}
      <h3 style={statusStyle}>{name}</h3>
      <dl>
        <dt>Started On</dt>
        <dd>{startedOn}</dd>
        <dt>Current Status</dt>
        <dd>{currentStatus}</dd>
        <dt>Coach</dt>
        <dd>{coach}</dd>
        <dt>Trainer</dt>
        <dd>{trainer}</dd>
      </dl>
    </div>
  );
}

export default CohortDetails;
```

App.js

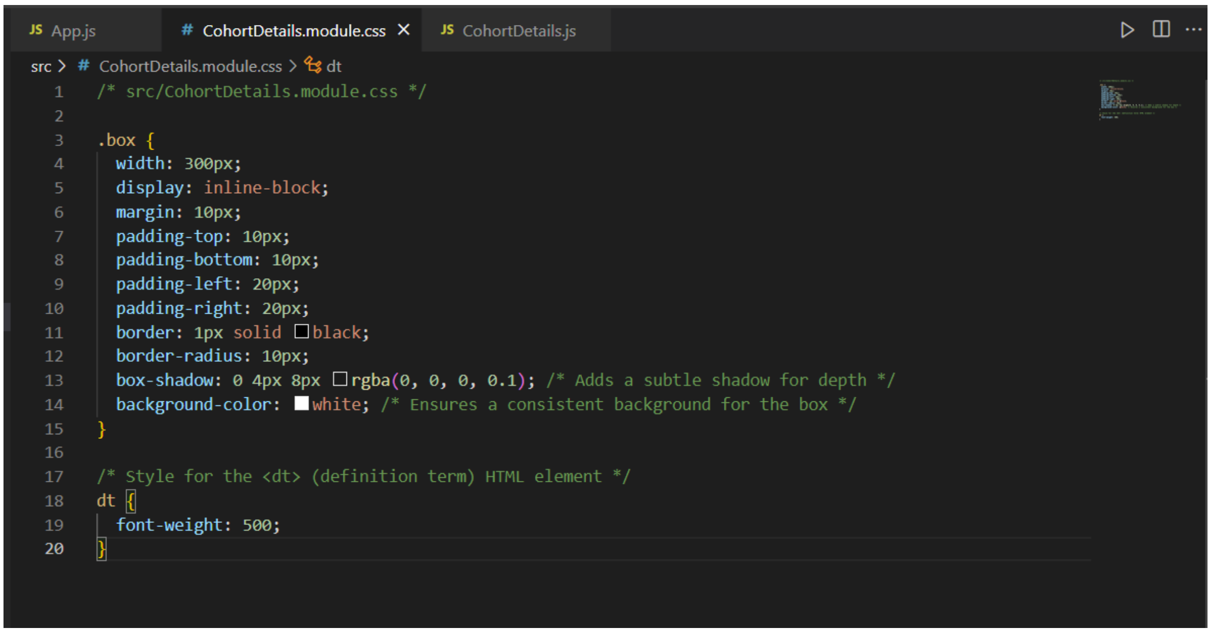


App.js | CohortDetails.module.css | CohortDetails.js

src > App.js > default

```
function App() {
  const cohorts = [
      startedOn: "10-Sep-2021",
      currentStatus: "Ongoing",
      coach: "Apoorv",
      trainer: "Elisa Smith"
    },
    {
      id: 3,
      name: "CDBJF21025 - Java FSD",
      startedOn: "24-Dec-2021",
      currentStatus: "Ongoing",
      coach: "Aathma",
      trainer: "John Doe"
    }
  ];

  return (
    <div className="App">
      <h1>Cohorts Details</h1>
      {/* This div helps to arrange cohort boxes in a row */}
      <div style={{ display: 'flex', flexWrap: 'wrap', justifyContent: 'center' }}>
        {cohorts.map(cohort => (
          <CohortDetails key={cohort.id} cohort={cohort} />
        ))}
      </div>
    </div>
  );
}

export default App;
```

CohortDetails.modules.css

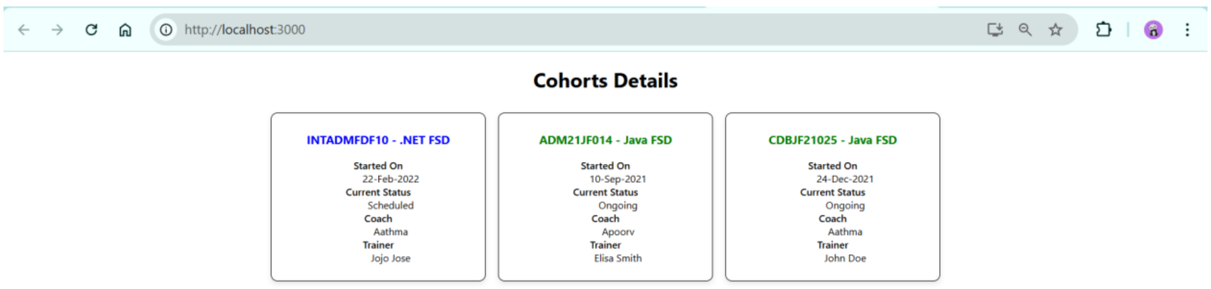


App.js | CohortDetails.module.css | CohortDetails.js

src > CohortDetails.module.css > dt

```
/* src/CohortDetails.module.css */

.box {
  width: 300px;
  display: inline-block;
  margin: 10px;
  padding-top: 10px;
  padding-bottom: 10px;
  padding-left: 20px;
  padding-right: 20px;
  border: 1px solid black;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /* Adds a subtle shadow for depth */
  background-color: white; /* Ensures a consistent background for the box */
}

/* Style for the <dt> (definition term) HTML element */
dt {
  font-weight: 500;
}
```

Ouput:



http://localhost:3000

Cohorts Details

INTADMFDF10 - .NET FSD
Started On
22-Feb-2022
Current Status
Scheduled
Coach
Aathma
Trainer
Jojo Jose

ADM21JF014 - Java FSD
Started On
10-Sep-2021
Current Status
Ongoing
Coach
Apoorv
Trainer
Elisa Smith

CDBJF21025 - Java FSD
Started On
24-Dec-2021
Current Status
Ongoing
Coach
Aathma
Trainer
John Doe