React .JS-1

Objectives:

Q1>Define SPA and its benefits.
Ans: A single page application is a type of web application that loads and updates content dynamically without refreshing the entire page.SPA is a modern technology that enhance user experience by minimizing interuptions and providing a smooth interface.When clicked on something in a SPA,it only send the necessary information to the browser and the browser rednders it.
Benefits:Dynamic Intercations;Real time updates;
Mobile friendly;Rich user interface;Minimized server load;Single page content;Cross platform consistency

Q2>Define React and identify its working.
Ans: React is a powerful JS library for building fast,scalable front-end apllications.React is known for component based structure ,SPA's,virtual DOM,enabling efficient UI updatedand seamless user experience.Allows for the creation of reusable UI components.
Working:-
>Initially there is an actual DOM(Real DOM) containing a DIV with 2 child elements:h1 and h2
>Reacts maintains a previous virtual DOM to track the UI stage before any updates
>When a change occurs(eg:adding a h3 element) react generates a new virtual DOM.
>React compares the previous virtual DOM with the new virtual DOM using a procces called reconcilation.
>React identifies the differnence (in this case the h3 elemet)
>Instead of updating the entire DOM,react updates only the changed part in the actual DOM,making the update process more efficient.

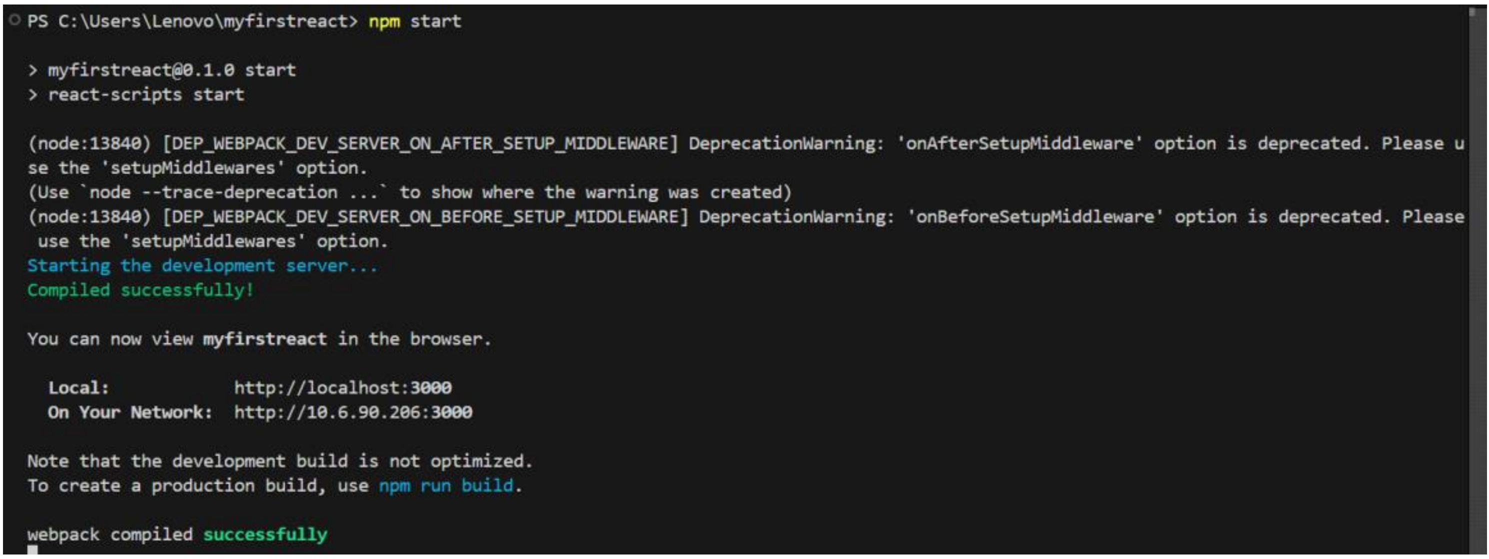
Q3>Identify the differences between SPA and MPA
Ans:
SPA:- A single page application(SPA) is a type of web application that loads and updates content dynamically without refreshing the entire page.SPA's is a modern technology that enhance user experience by minimizing interuptions and providing a smooth interface.Full page reload is the main advantage.
MPA:- A multi page apllication(MPA) loads a new HTML page from the server for each user interaction and request.It comes with full page reload.Slower since each request reloads the entire page.Faster initial loads,but slower navigation between pages.Suitable for heavy content heavy website.

Q4>Explain Pros & Cons of Single-Page Application
Ans:
Pros: Dynamic Interactions,Real time updates,Mobile friendly,Rich user interfaces,Minimized Server Loads,Single Page content.
Cons: Slow first load,SEO challenges,Limited browser support,Security risks,Dependency on javascript,Complex development.

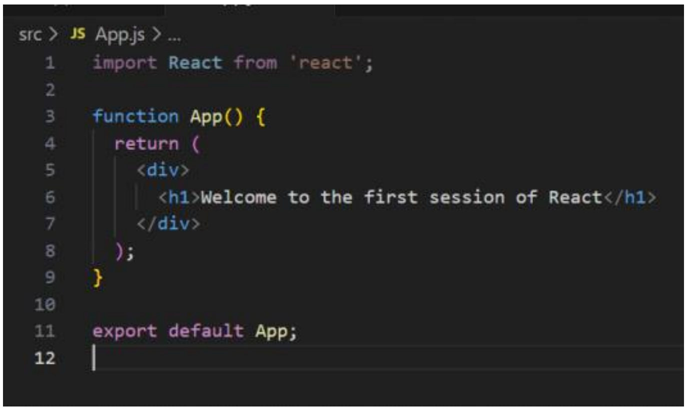
Q5>Explain about React
Ans:
React is a powerful JS library for building fast,scalable front end applications.
React is known for component based structure ,SPA's,Virtual Dom,Enabling efficient UI updates and seamless user experience.
Used to create SPA's.
Allows for the creation of reusable UI components.
Choosing react because it offers virtual DOM,One way data binding,Component based architecture.
Before react front end development struggles with- Manual DOM manipulation,Complex stage management,Tight coupling in frameworks.

Q6>Define virtual DOM
Ans:
The Virtual DOM (VDOM) is a lightweight, in-memory representation of the real DOM (Document Object Model). It helps React manage UI updates more efficiently by keeping a virtual version of the UI in memory. When changes occur, React updates only the necessary parts of the real DOM, instead of re-rendering everything.

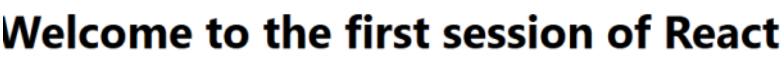
Q7>Explain Features of React
Ans:
>Virtual DOM:-Real updates only the chnaged parts of the DOM,resulting in a faster rendering
>One way data binding:-ensures predictable and easy to debug data flow
>Component based architecture-Breaks UI into reusable pieces,improving the code reusability and scalability.



```
PS C:\Users\Lenovo\myfirstreact> npm start

> myfirstreact@0.1.0 start
> react-scripts start

(node:13840) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please u
se the 'setupMiddlewares' option.
(Use `node --trace-deprecation ...` to show where the warning was created)
(node:13840) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please
 use the 'setupMiddlewares' option.
Starting the development server...
Compiled successfully!

You can now view myfirstreact in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://10.6.90.206:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```



```
import React from 'react';

function App() {
  return (
    <div>
      <h1>Welcome to the first session of React</h1>
    </div>
  );
}

export default App;
```



Welcome to the first session of React

React.JS-2

Q1>Explain React components.

Ans:These are independent,reusable building blocks in a react application that defines what gets displayed on the UI.They accept inputs called props and return react elements describing the UI.

Q2>Identify the differences between components and JavaScript functions.

Ans:

>Components:-A component is a building block that encapsulated a piece of UI.It defines the structure and behaviour of the UI,either by managing it's own stat eor by receiving data through props and rendering content accordingly .Components are reusable and can be composed together to build complex UI's in a modular way.

>JS funtions:-At its core, a **JavaScript function** is a block of code designed to perform a particular task. It's a fundamental building block of JavaScript programming, enabling you to write modular, reusable, and organized code.

Q3>Identify the types of components.

Ans:

>Functional components:-These are simpler and preferred for most use cases.They are JS functions that return react elements.With intro of react hooks,functional components can also manage state and lifecycle events.

>Class components:-These are ES6 classes that extend React.component.Including additional features like state management and lifecycle methods.

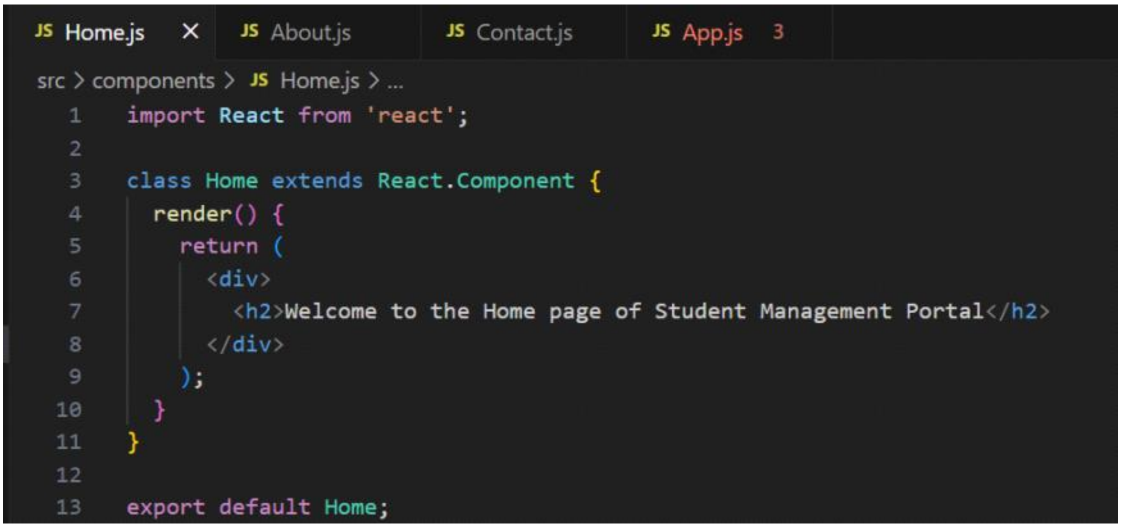
Q4>Explain class component.

Ans:These are ES6 classes that extend React.component.Including additional features like state management and lifecycle methods.

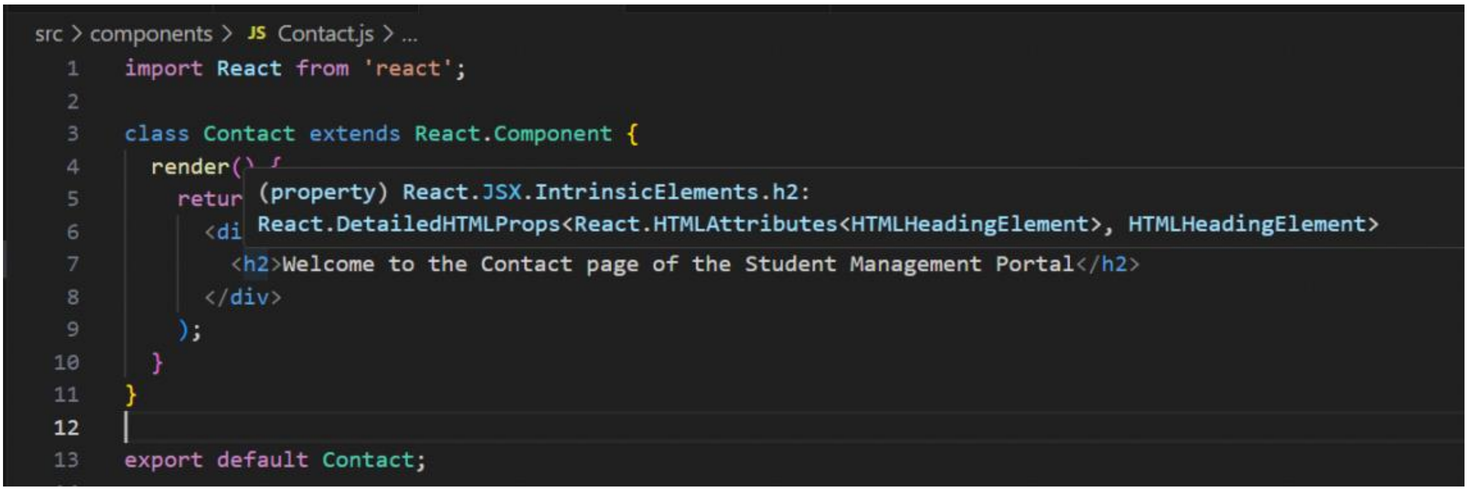
Q5>Explain function component.
Ans:These are simpler and preferred for most use cases.They are JS functions that return react elements.With intro of react hooks,functional components can also manage state and lifecycle events.

Q6>Define component constructor.
Ans:
In React, applications are built using isolated, reusable pieces of code called **components**. Think of them as custom, self-contained HTML elements that encapsulate their own logic, state, and UI.
The fundamental structure of a React component can be understood through two primary types: **Functional Components** and **Class Components**. While Class Components were historically prevalent, **Functional Components with Hooks** are now the recommended and dominant way to write React components.

Q7>Define render() function.
Ans:The render() function is a **core and essential method in React class components**. It's the heart of what a class component displays on the screen.
Its primary purpose is to **describe what should be rendered to the browser's DOM (Document Object Model)**. It returns a description of the UI, which React then efficiently translates into actual DOM changes.

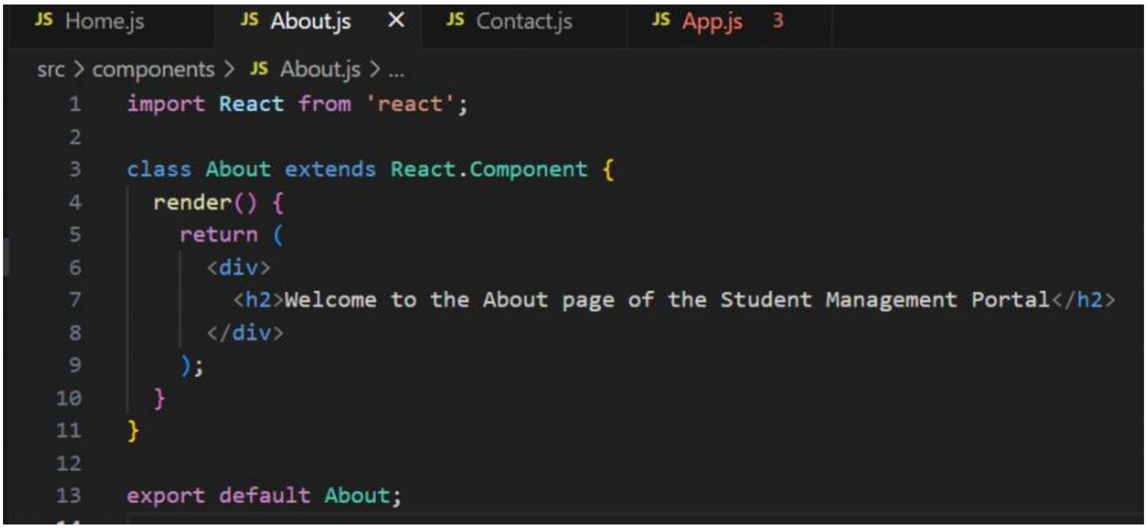


```
import React from 'react';

class Home extends React.Component {
  render() {
    return (
      <div>
        <h2>Welcome to the Home page of Student Management Portal</h2>
      </div>
    );
  }
}

export default Home;
```

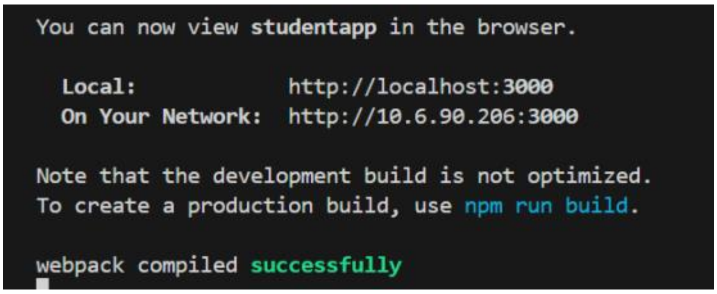


```
import React from 'react';

class Contact extends React.Component {
  render() {
    retur
      <di
        <h2>Welcome to the Contact page of the Student Management Portal</h2>
      </div>
    );
  }
}

export default Contact;
```

```
(property) React.JSX.IntrinsicElements.h2:
React.DetailedHTMLProps<React.HTMLAttributes<HTMLHeadingElement>, HTMLHeadingElement>
```

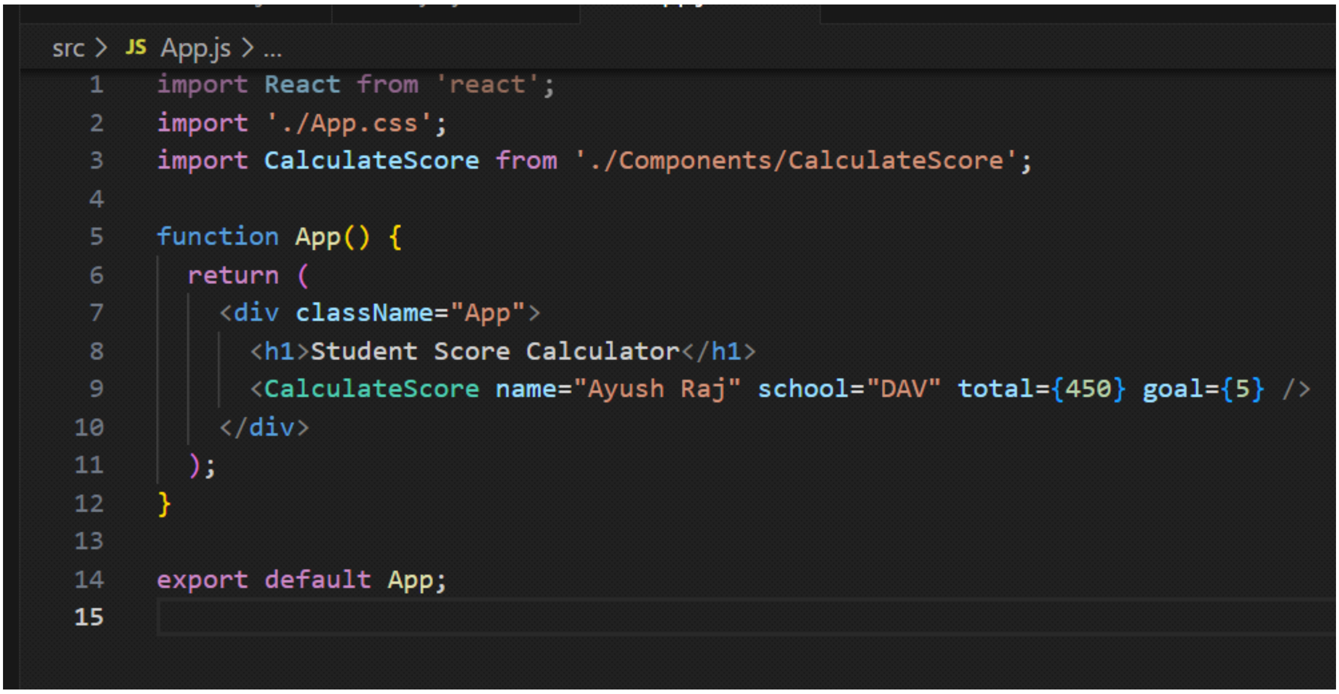


```
import React from 'react';

class About extends React.Component {
  render() {
    return (
      <div>
        <h2>Welcome to the About page of the Student Management Portal</h2>
      </div>
    );
  }
}

export default About;
```



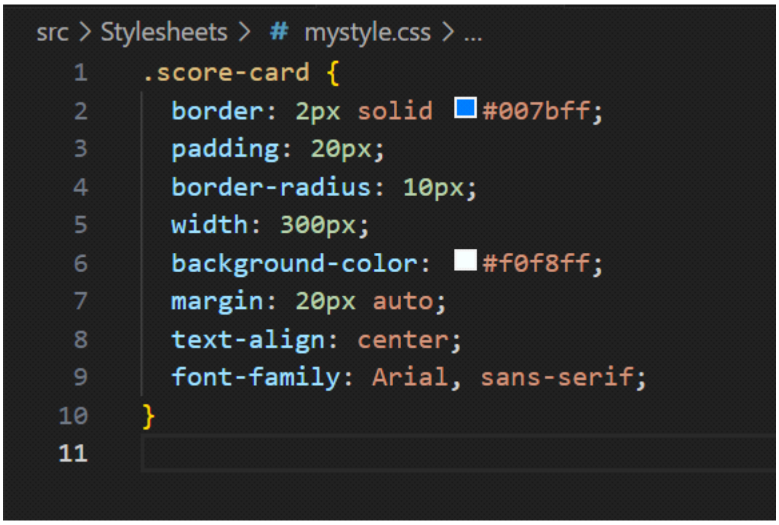
```
You can now view studentapp in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://10.6.90.206:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

React.JS-3



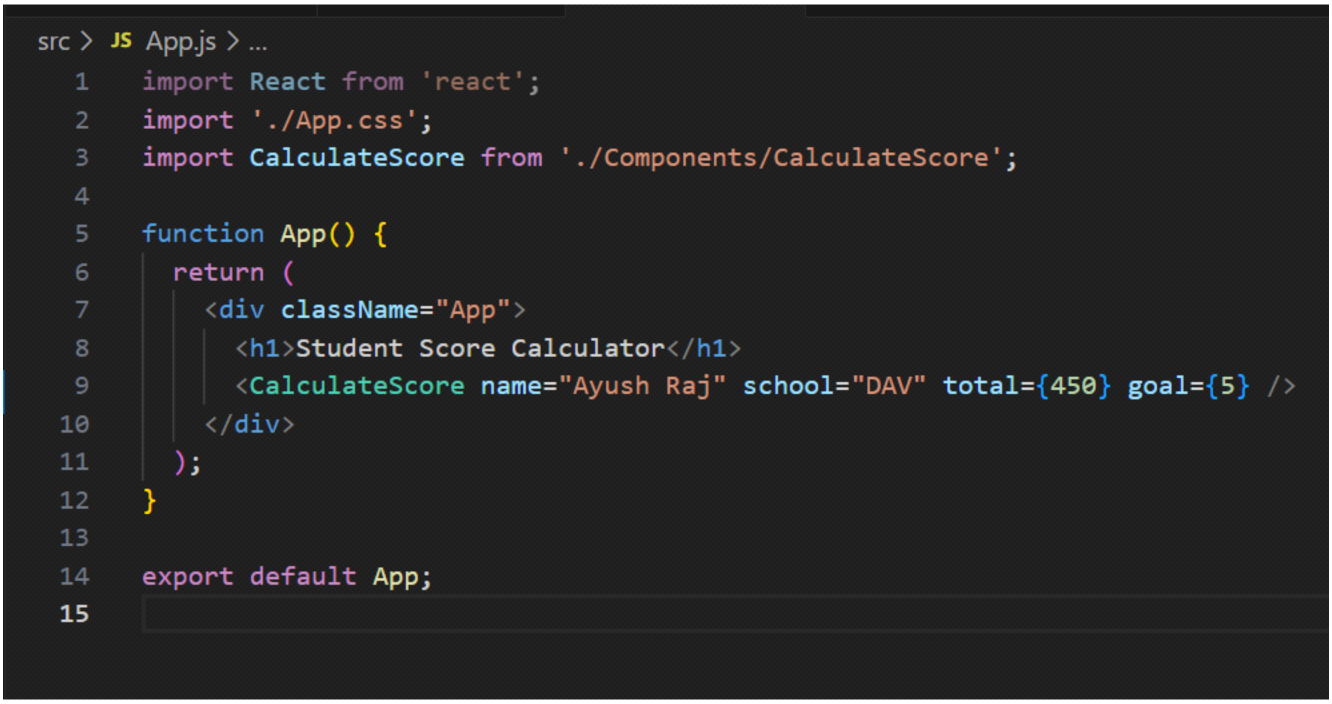
src > JS App.js > ...

```jsx
import React from 'react';
import './App.css';
import CalculateScore from './Components/CalculateScore';

function App() {
  return (
    <div className="App">
      <h1>Student Score Calculator</h1>
      <CalculateScore name="Ayush Raj" school="DAV" total={450} goal={5} />
    </div>
  );
}

export default App;

```



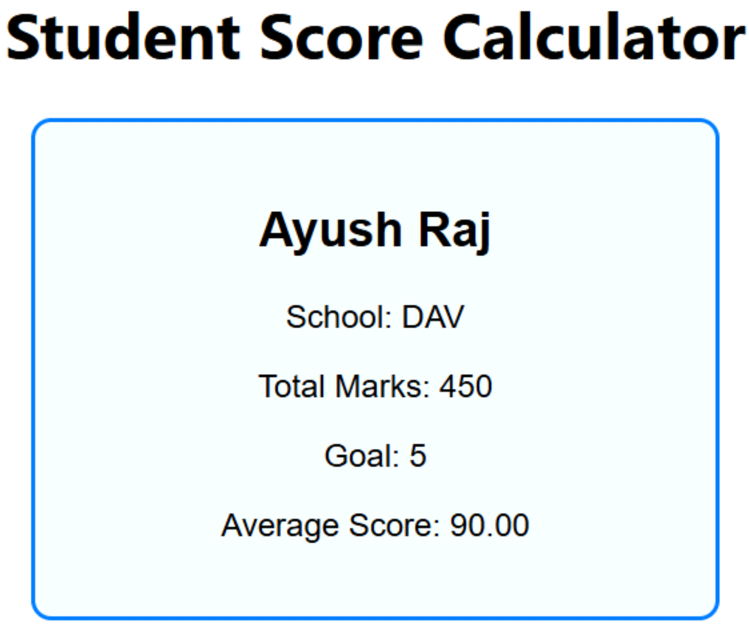
src > Stylesheets > # mystyle.css > ...

```css
.score-card {
  border: 2px solid #007bff;
  padding: 20px;
  border-radius: 10px;
  width: 300px;
  background-color: #f0f8ff;
  margin: 20px auto;
  text-align: center;
  font-family: Arial, sans-serif;
}

```



src > JS App.js > ...

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

function App() {
  return (
    <div className="App">
      <h1>Student Score Calculator</h1>
      <CalculateScore name="Ayush Raj" school="DAV" total={450} goal={5} />
    </div>
  );
}

export default App;

```



Student Score Calculator

Ayush Raj

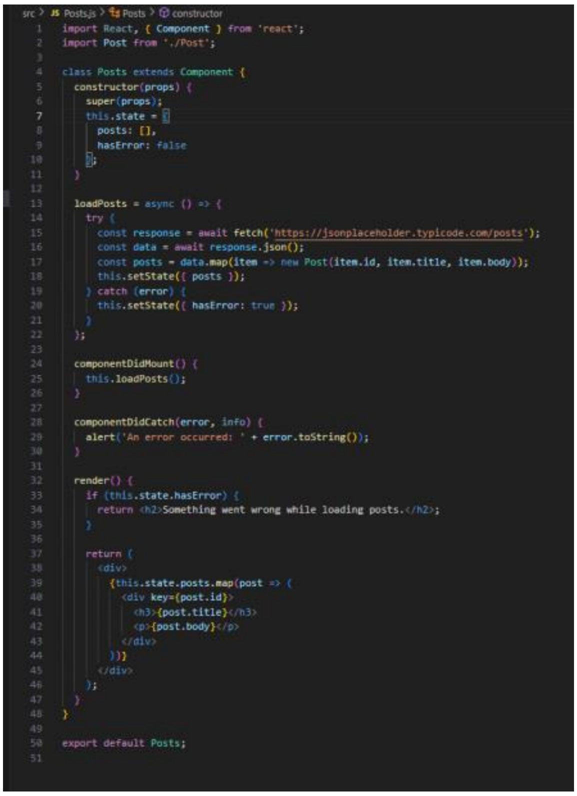
School: DAV

Total Marks: 450

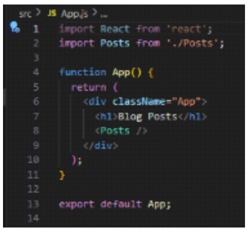
Goal: 5

Average Score: 90.00

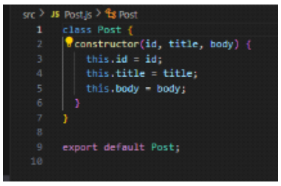
React.JS-4



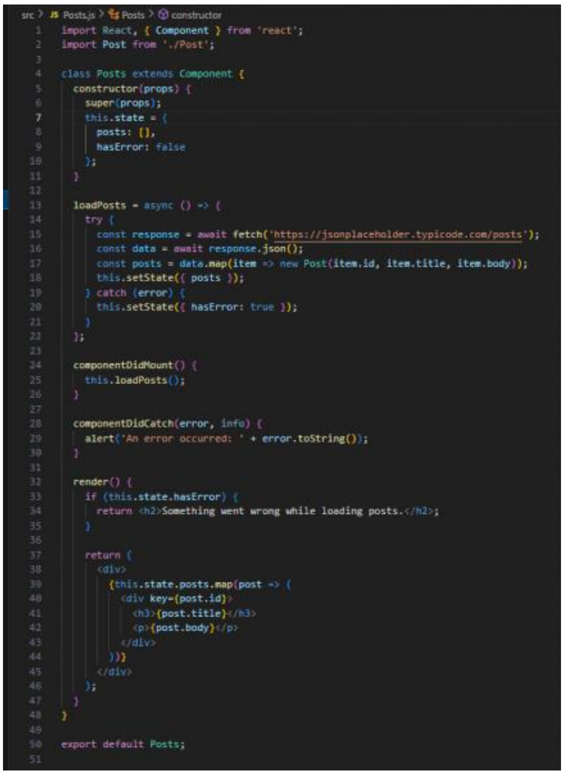
```javascript
import React, { Component } from 'react';
import Post from './Post';

class Posts extends Component {
  constructor(props) {
    super(props);
    this.state = {
      posts: [],
      hasError: false
    };
  }

  loadPosts = async () => {
    try {
      const response = await fetch('https://jsonplaceholder.typicode.com/posts');
      const data = await response.json();
      const posts = data.map(item => new Post(item.id, item.title, item.body));
      this.setState({ posts });
    } catch (error) {
      this.setState({ hasError: true });
    }
  };

  componentDidMount() {
    this.loadPosts();
  }

  componentDidCatch(error, info) {
    alert('An error occurred: ' + error.toString());
  }

  render() {
    if (this.state.hasError) {
      return <h2>Something went wrong while loading posts.</h2>;
    }

    return (
      <div>
        {this.state.posts.map(post => (
          <div key={post.id}>
            <h3>{post.title}</h3>
            <p>{post.body}</p>
          </div>
        ))}
      </div>
    );
  }
}

export default Posts;
```



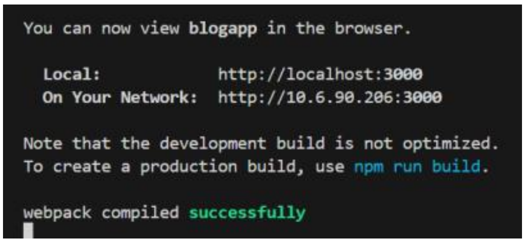
```javascript
import React from 'react';
import Posts from './Posts';

function App() {
  return (
    <div className="App">
      <h1>Blog Posts</h1>
      <Posts />
    </div>
  );
}

export default App;
```



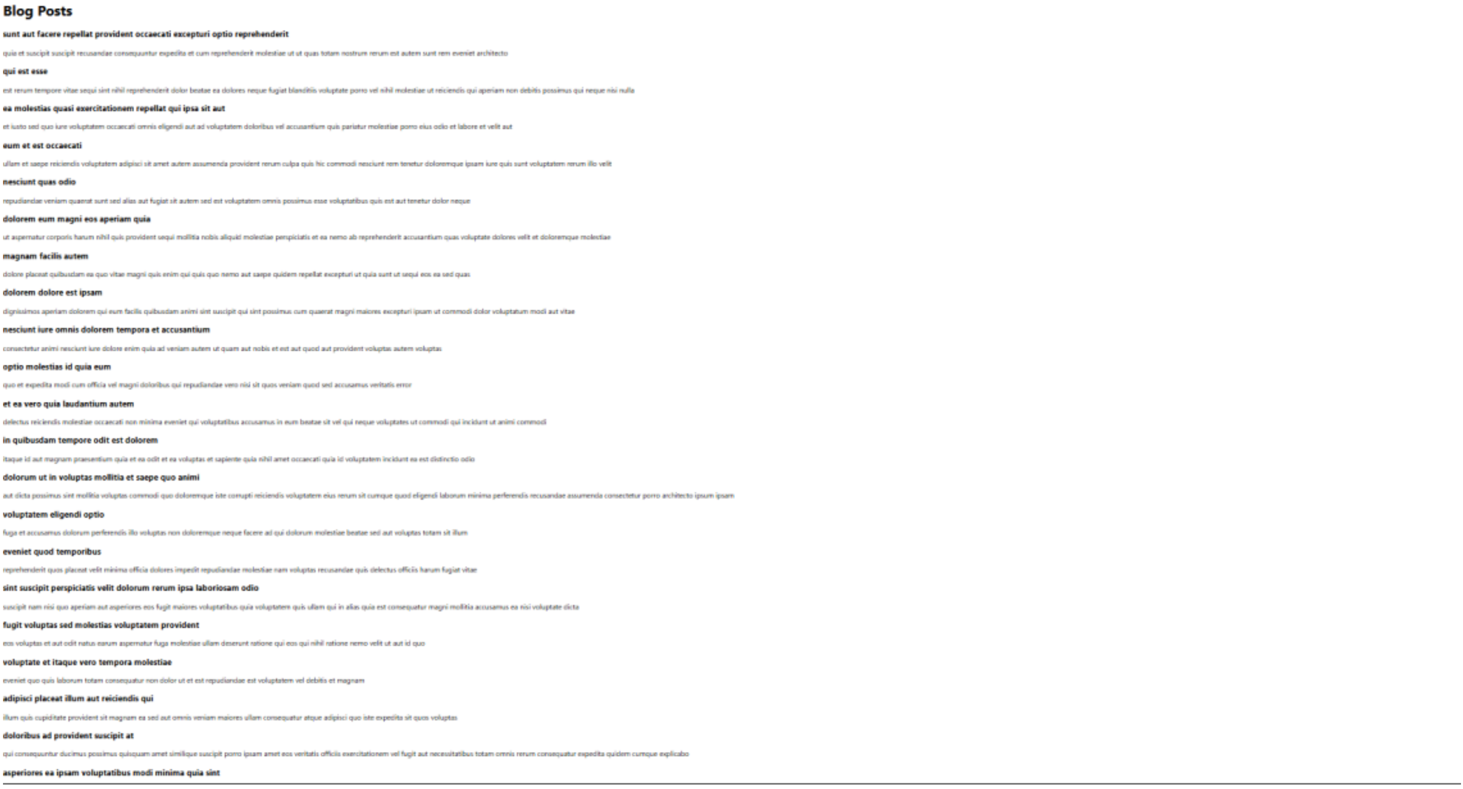
```javascript
class Post {
  constructor(id, title, body) {
    this.id = id;
    this.title = title;
    this.body = body;
  }
}

export default Post;
```



```javascript
import React, { Component } from 'react';
import Post from './Post';

class Posts extends Component {
  constructor(props) {
    super(props);
    this.state = {
      posts: [],
      hasError: false
    };
  }

  loadPosts = async () => {
    try {
      const response = await fetch('https://jsonplaceholder.typicode.com/posts');
      const data = await response.json();
      const posts = data.map(item => new Post(item.id, item.title, item.body));
      this.setState({ posts });
    } catch (error) {
      this.setState({ hasError: true });
    }
  };

  componentDidMount() {
    this.loadPosts();
  }

  componentDidCatch(error, info) {
    alert('An error occurred: ' + error.toString());
  }

  render() {
    if (this.state.hasError) {
      return <h2>Something went wrong while loading posts.</h2>;
    }

    return (
      <div>
        {this.state.posts.map(post => (
          <div key={post.id}>
            <h3>{post.title}</h3>
            <p>{post.body}</p>
          </div>
        ))}
      </div>
    );
  }
}

export default Posts;

```



```
You can now view blogapp in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://10.6.90.206:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```



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

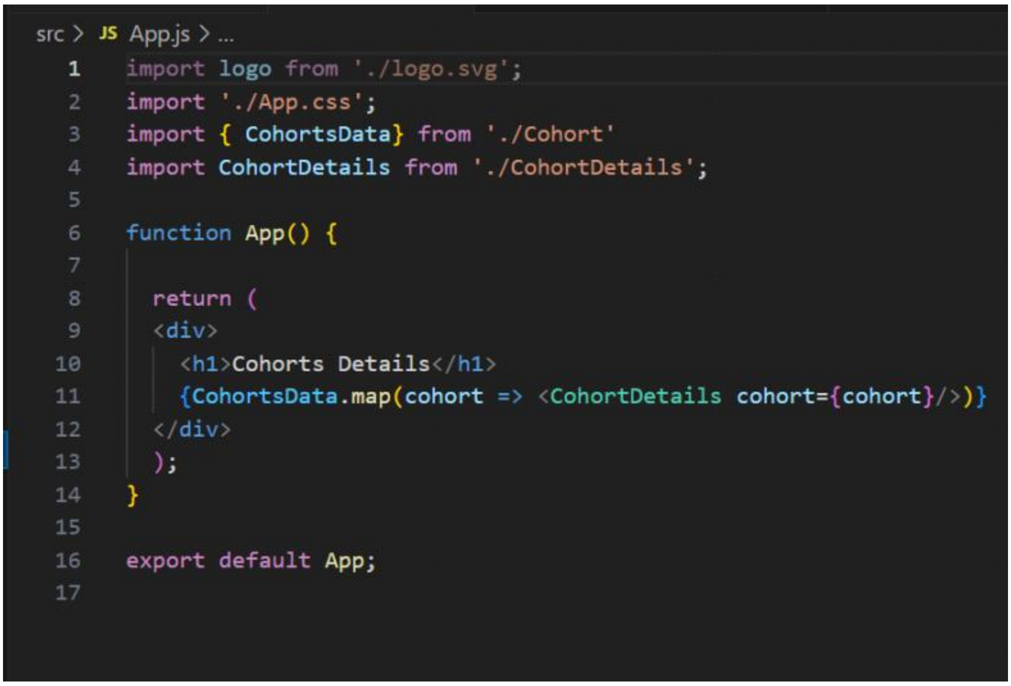
illum quis cupiditate provident sit magnam ea sed aut omnis veniam maiores ullam consequatur atque adipisci quo iste expedita sit quos voluptas

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

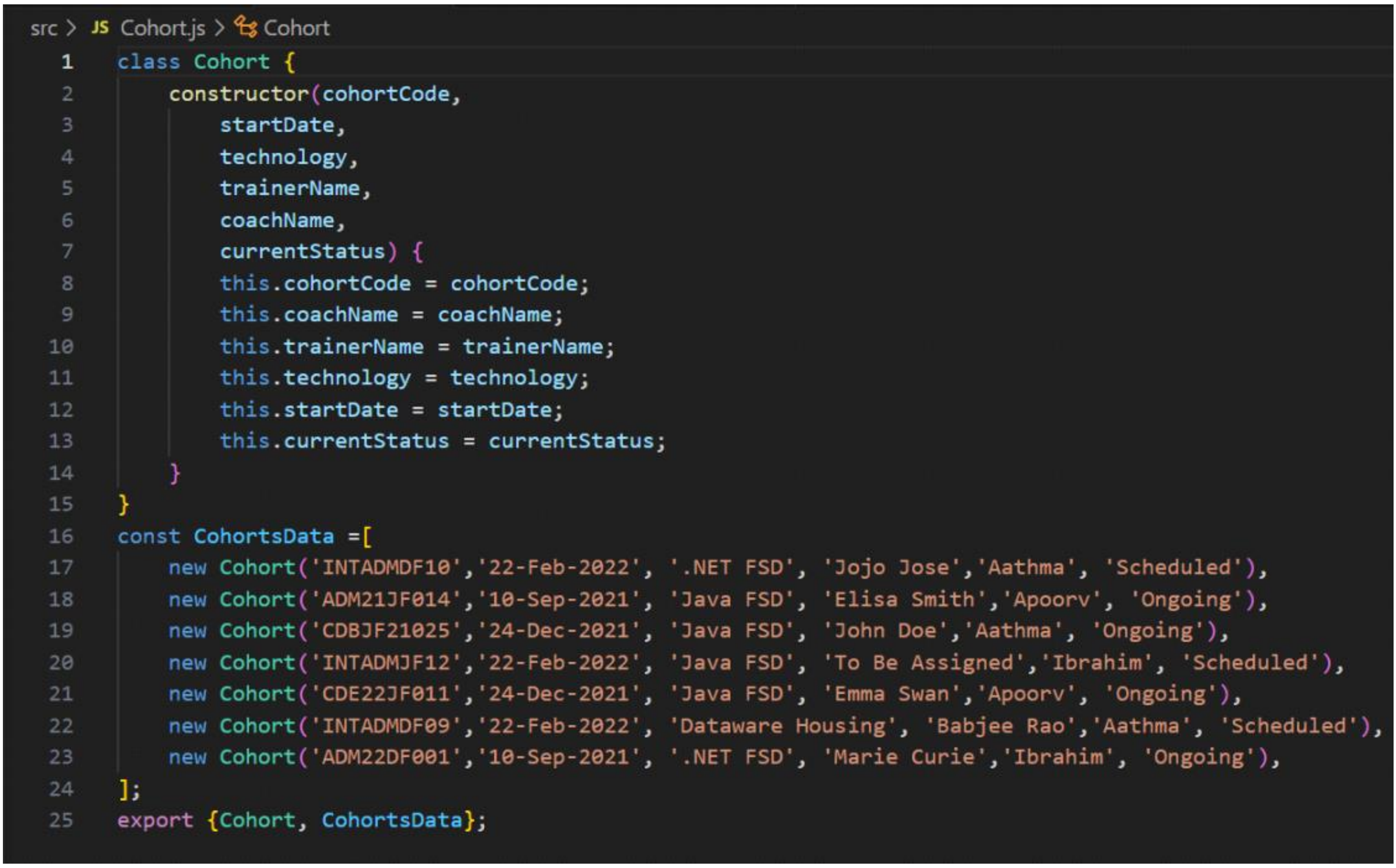
asperiores ea ipsam voluptatibus modi minima quia sint

React.JS-5



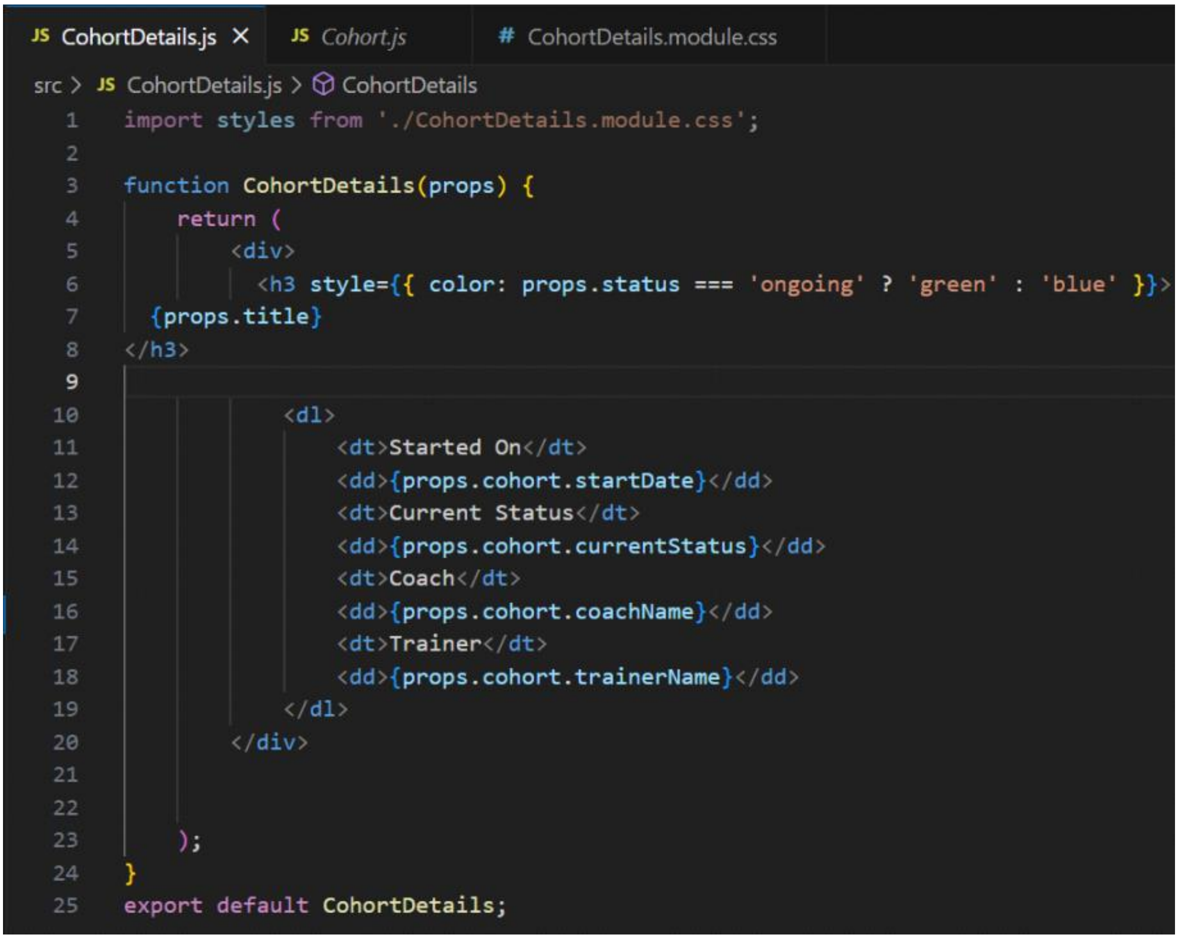
src > JS App.js > ...

```
import logo from './logo.svg';
import './App.css';
import { CohortsData} from './Cohort'
import CohortDetails from './CohortDetails';

function App() {

  return (
  <div>
    <h1>Cohorts Details</h1>
    {CohortsData.map(cohort => <CohortDetails cohort={cohort}/>)}
  </div>
  );
}

export default App;
```

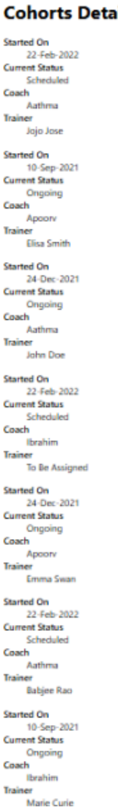


src > JS Cohort.js > Cohort

```
class Cohort {
    constructor(cohortCode,
        startDate,
        technology,
        trainerName,
        coachName,
        currentStatus) {
        this.cohortCode = cohortCode;
        this.coachName = coachName;
        this.trainerName = trainerName;
        this.technology = technology;
        this.startDate = startDate;
        this.currentStatus = currentStatus;
    }
}
const CohortsData =[
    new Cohort('INTADMDF10','22-Feb-2022', '.NET FSD', 'Jojo Jose','Aathma', 'Scheduled'),
    new Cohort('ADM21JF014','10-Sep-2021', 'Java FSD', 'Elisa Smith','Apoorv', 'Ongoing'),
    new Cohort('CDBJF21025','24-Dec-2021', 'Java FSD', 'John Doe','Aathma', 'Ongoing'),
    new Cohort('INTADMJF12','22-Feb-2022', 'Java FSD', 'To Be Assigned','Ibrahim', 'Scheduled'),
    new Cohort('CDE22JF011','24-Dec-2021', 'Java FSD', 'Emma Swan','Apoorv', 'Ongoing'),
    new Cohort('INTADMDF09','22-Feb-2022', 'Dataware Housing', 'Babjee Rao','Aathma', 'Scheduled'),
    new Cohort('ADM22DF001','10-Sep-2021', '.NET FSD', 'Marie Curie','Ibrahim', 'Ongoing'),
];
export {Cohort, CohortsData};
```



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

function CohortDetails(props) {
    return (
        <div>
          <h3 style={{ color: props.status === 'ongoing' ? 'green' : 'blue' }}>
  {props.title}
</h3>

            <dl>
                <dt>Started On</dt>
                <dd>{props.cohort.startDate}</dd>
                <dt>Current Status</dt>
                <dd>{props.cohort.currentStatus}</dd>
                <dt>Coach</dt>
                <dd>{props.cohort.coachName}</dd>
                <dt>Trainer</dt>
                <dd>{props.cohort.trainerName}</dd>
            </dl>
        </div>


    );
}
export default CohortDetails;
```



```
Cohorts Details

Started On
    22-Feb-2022
Current Status
    Scheduled
Coach
    Aathma
Trainer
    Jojo Jose

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

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

Started On
    22-Feb-2022
Current Status
    Scheduled
Coach
    Ibrahim
Trainer
    To Be Assigned

Started On
    24-Dec-2021
Current Status
    Ongoing
Coach
    Apoorv
Trainer
    Emma Swan

Started On
    22-Feb-2022
Current Status
    Scheduled
Coach
    Aathma
Trainer
    Babjee Rao

Started On
    10-Sep-2021
Current Status
    Ongoing
Coach
    Ibrahim
Trainer
    Marie Curie
```