Week 6 Hands-On

Task 1:

1. **What is an SPA and why is it useful?**

SPA stands for **Single-Page Application**. It’s a type of web app that loads a single HTML page and updates content dynamically without refreshing the page. Instead of loading new pages from the server each time, it updates the current page using JavaScript.

**Faster navigation** – No page reloads.

**Feels like a mobile app** – Smooth and interactive.

**Feels like a mobile app** – Smooth and interactive.

**Less server load** – Just fetches data, not entire pages.

1. **What is React and how does it work?**

React is a **JavaScript library** built by Facebook for building **user interfaces**, mainly for SPAs. It works as

* It breaks the UI into **small components** like buttons, forms, cards, etc.
* React keeps a **virtual copy of the DOM** and updates only what's changed, making it super fast.
* It uses **JSX**, which lets you write HTML-like code in JavaScript.
* It follows **one-way data flow** – data moves in one direction, which makes apps predictable and easier to debug.

1. **Identify the differences between** **SPA vs MPA**

| **SPA (Single Page App)** | **MPA (Multi Page App)** |
| --- | --- |
| Loads one page, updates content | Loads a new page every time |
| Feels faster and smoother | Slower due to full reloads |
| More complex to build | Easier to set up |
| Not great for SEO by default | Better SEO |
| Used in apps like Gmail, Facebook | Used in sites like Amazon, blogs |

1. **Explain Pros & Cons of Single-Page Application**

Pros**:** Fast and responsive, Better user experience, Reusable components, Less server work

Cons**: Not SEO-friendly** (unless optimized), Longer initial load time, **Needs JavaScript** to function properly, Can be tricky to manage routing and state.

1. **Explain about React**

React is a tool that helps to build modern web apps. It’s like Lego for web pages – create small blocks (components) and put them together to build the site.

1. **Define virtual DOM**

The **Virtual DOM** is like a **blueprint** of web pages. React uses it to keep track of what the page should look like. When something changes (like user clicks a button), React **doesn’t update the whole page**. Instead, it compares the old blueprint with the new one, finds the changes, and updates only those parts in the real page.

This makes the app **faster and more efficient**.

1. **Explain Features of React**

One of the key features is JSX, which allows developers to write HTML-like syntax directly within JavaScript, making the code more readable and easier to write.

React follows a component-based architecture, meaning the UI is divided into reusable, independent components that help in maintaining and scaling large applications.

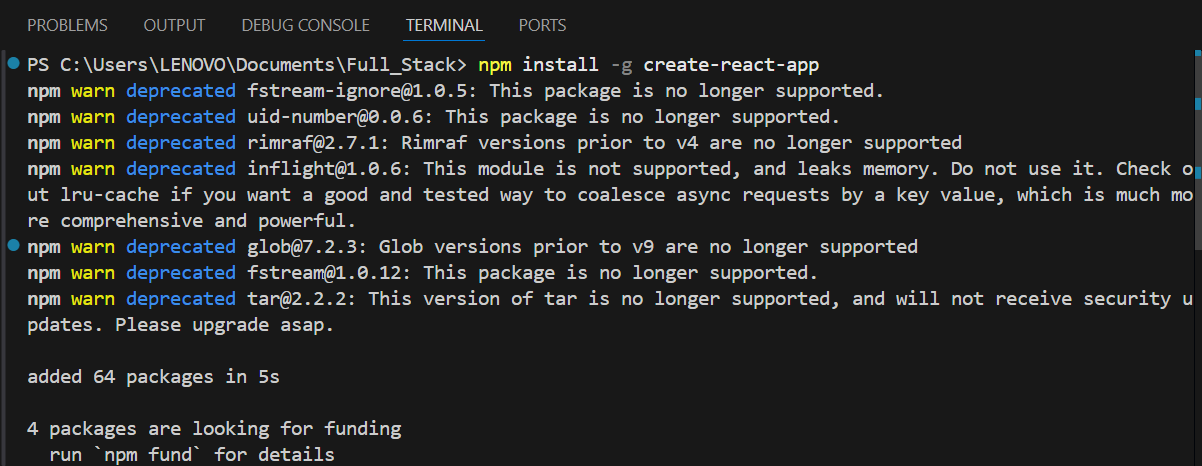
One of the important feature is the Virtual DOM, which optimizes rendering by updating only the parts of the actual DOM that have changed, improving performance.

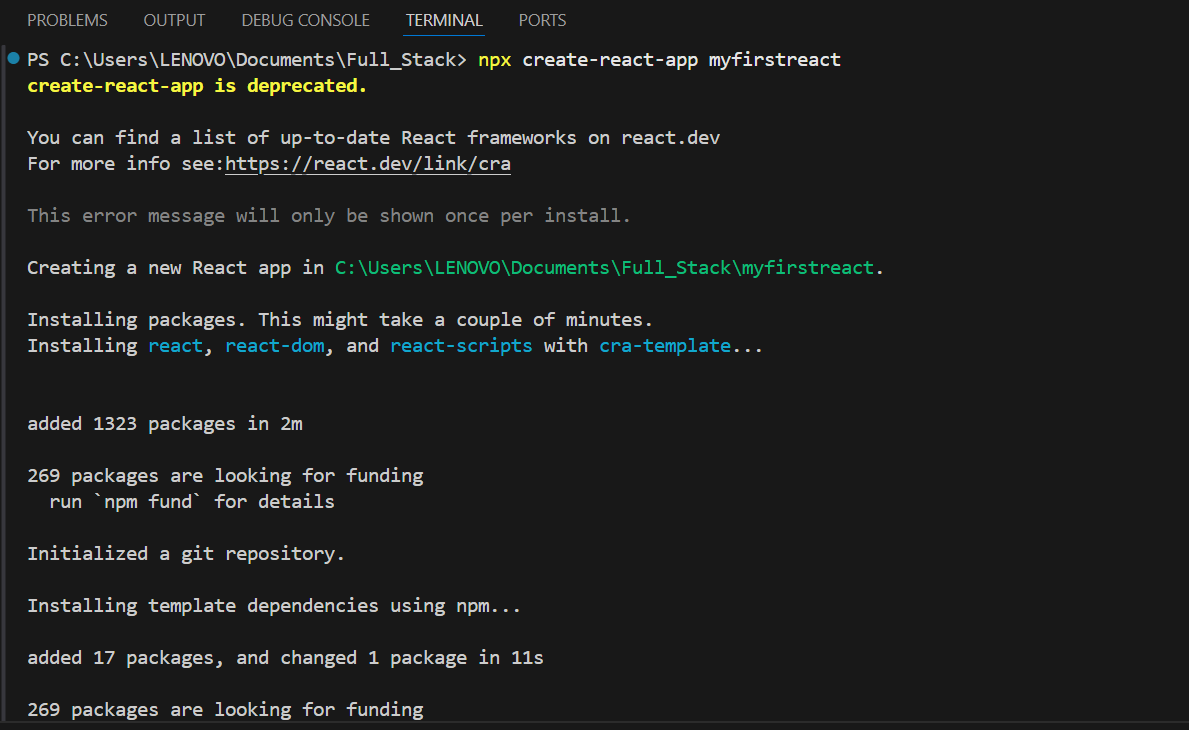
React also uses one-way data binding, ensuring a predictable flow of data which makes debugging and testing easier.

React Hooks, developers can now manage state and side effects in functional components without relying on class components.

Create a new React Application with the name “myfirstreact”, Run the application to print “welcome to the first session of React” as heading of that page.

**Project Creation**



****

**App.js**

function App() {

  return (

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

  );

}

export default App;

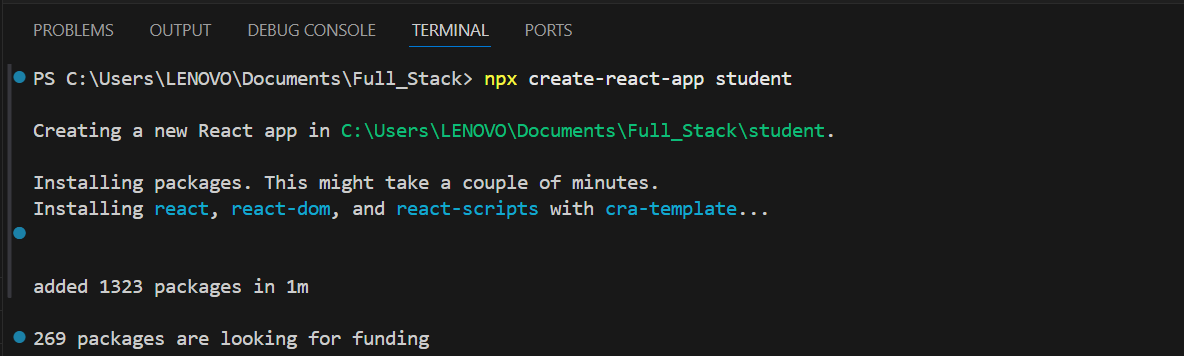
**Output:**



Task 2

Create a react app for Student Management Portal named StudentApp and create a component named Home which will display the Message “Welcome to the Home page of Student Management Portal”. Create another component named About and display the Message “Welcome to the About page of the Student Management Portal”. Create a third component named Contact and display the Message “Welcome to the Contact page of the Student Management Portal”. Call all the three components.

**Project Creation**



Home.js

import React from 'react';

function Home() {

  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 logo from './logo.svg';

import './App.css';

import Home from './Components/Home';

import About from './Components/About';

import Contact from './Components/Contact';

function App() {

  return (

    <div className="container">

      <Home />

      <About />

      <Contact />

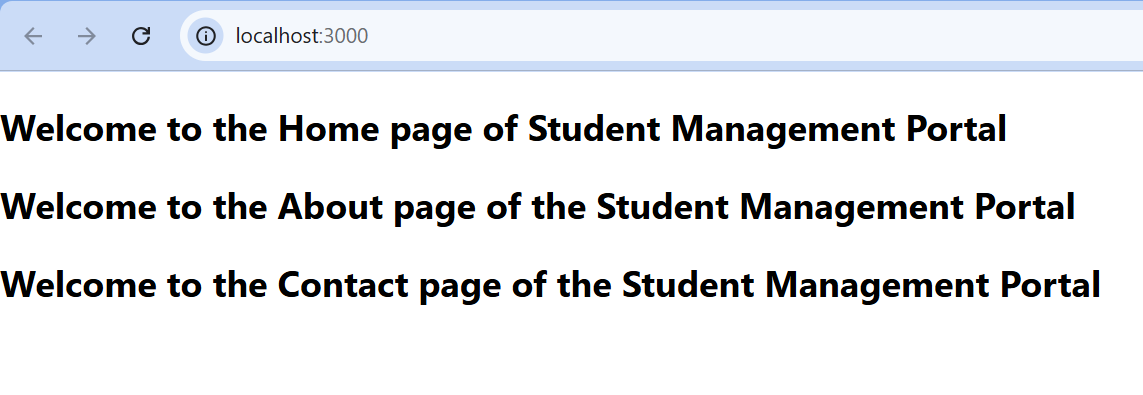
    </div>

  );

}

export default App;

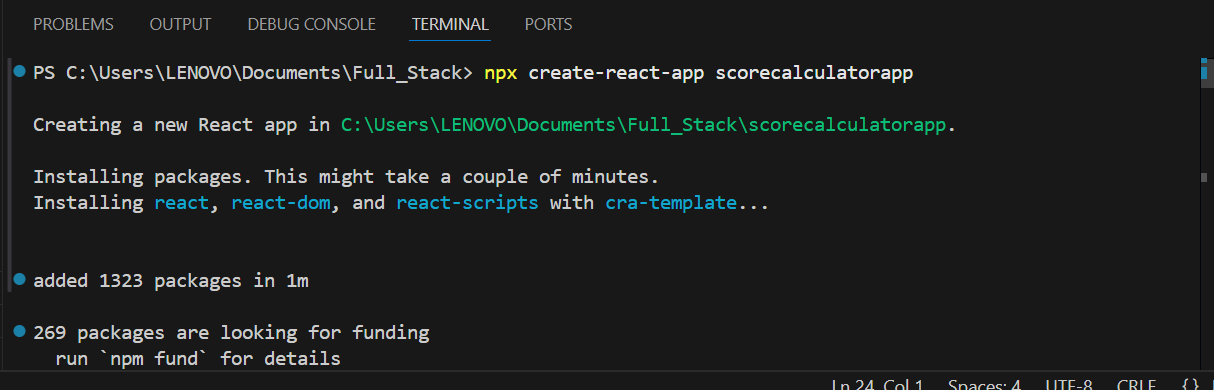
Output:



Task 3

Create a react app for Student Management Portal named scorecalculatorapp and create a function component named “CalculateScore” which will accept Name, School, Total and goal in order to calculate the average score of a student and display the same.

Project Creation:



CalculateScore.js

import React from 'react';

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

function CalculateScore() {

  const name = "John Doe";

  const school = "Springfield High School";

  const total = 450;

  const goal = 500;

  const average = total / 5;

  return (

    <div className="formatstyle">

      <h2>Student Score Details</h2>

      <p className="Name"><strong>Name:</strong> {name}</p>

      <p className="School"><strong>School:</strong> {school}</p>

      <p className="Total"><strong>Total Marks:</strong> {total}</p>

      <p><strong>Goal Marks:</strong> {goal}</p>

      <p className="Score"><strong>Average Score:</strong> {average}</p>

    </div>

  );

}

export default CalculateScore;

mystyle.css

.Name {

    font-weight: 300;

    color: blue;

}

.School {

    color: crimson;

}

.Total {

    color: darkmagenta;

}

.formatstyle {

    text-align: center;

    font-size: large;

}

.Score {

    color: forestgreen;

}

App.js

import React from 'react';

import './App.css';

import CalculateScore from './Components/CalculateScore';

function App() {

  return (

    <div className="App">

      <CalculateScore />

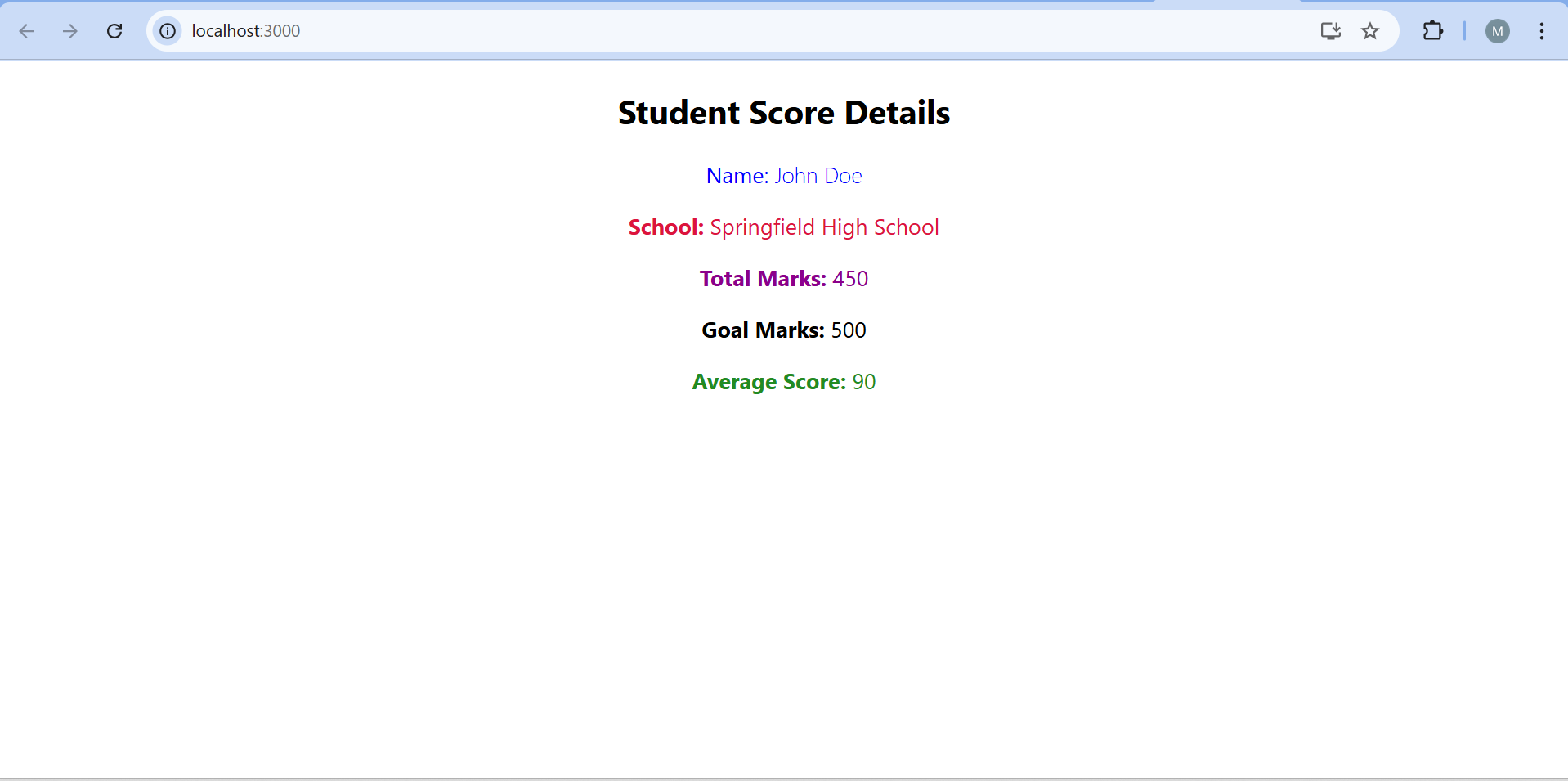
    </div>

  );

}

export default App;

Output:



Task 4

* Create a new react application using *create-react-app* tool with the name as “blogapp”
* Open the application using VS Code
* Create a new file named as **Post.js** in **src folder** with following properties

Post.js

class Post{

    constructor(id, title, body)

    {

        this.id=id;

        this.title=title;

        this.body=body;

    }

}

export default Post;

Posts.js

import React from 'react';

import Post from './Post';

class Posts extends React.Component {

    constructor(props) {

        super(props);

        this.state = {

            postList: [],

            error: null

        };

    }

    loadPosts() {

        const post1 = new Post(1, "React Intro", "Basics of React.js");

        const post2 = new Post(2, "React State", "Understanding state management");

        const post3 = new Post(3, "React Props", "Passing data via props");

        const posts = [post1, post2, post3];

        this.setState({ postList: posts });

    }

    componentDidMount() {

    this.loadPosts();

  }

  componentDidCatch(error, info) {

    this.setState({ error: error.toString() });

  }

  render() {

    return (

      <div className="post-container">

        <h2>Posts</h2>

        {this.state.error && <p className="error">Error: {this.state.error}</p>}

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

          <div key={post.id} className="post">

            <h3>{post.title}</h3>

            <p>{post.body}</p>

          </div>

        ))}

      </div>

    );

  }

}

export default Posts;

App.js

import React from 'react';

import Posts from './Posts';

import './styles.css';

function App() {

  return (

    <div className="App">

      <h1>My Blog</h1>

      <Posts />

    </div>

  );

}

export default App;

index.js

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<App />);

styles.css

/\* styles.css \*/

.App {

  font-family: Arial, sans-serif;

  padding: 20px;

}

.post-container {

  margin-top: 20px;

}

.post {

  border: 1px solid #ddd;

  padding: 15px;

  margin-bottom: 10px;

  border-radius: 8px;

  background-color: #f9f9f9;

}

.post h3 {

  margin-top: 0;

}

.error {

  color: red;

  font-weight: bold;

}

Output:



Task 5

My Academy team at Cognizant want to create a dashboard containing the details of ongoing and completed cohorts. A react application is created which displays the detail of the cohorts using react component. You are assigned the task of styling these react components.

Download and build the attached react application.



1. Unzip the react application in a folder
2. Open command prompt and switch to the react application folder
3. Restore the node packages using the following commands



Figure 1: Restore packages

1. Open the application using VS Code
2. Create a new CSS Module in a file called “CohortDetails.module.css”
3. Define a css class with the name as “box” with following properties

*Width = 300px;*

*Display = inline block;*

*Overall 10px margin*

*Top and bottom padding as 10px*

*Left and right padding as 20px*

*1 px border in black color*

*A border radius of 10px*

1. Define a css style for html <dt> element using tag selector. Set the font weight to 500.
2. Open the cohort details component and import the CSS Module
3. Apply the box class to the container div
4. Define the style for <h3> element to use “green” color font when cohort status is “ongoing” and “blue” color in all other scenarios.

CohortDetails.js

import React from 'react';

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

function CohortDetails({ cohort }) {

  const isOngoing = cohort.status.toLowerCase() === 'ongoing';

  const headingStyle = { color: isOngoing ? 'green' : 'blue' };

  return (

    <div className={styles.box}>

      <h3 style={headingStyle}>{cohort.id}</h3>

      <dl>

        <dt>Started On</dt>

        <dd>{cohort.started}</dd>

        <dt>Current Status</dt>

        <dd>{cohort.status}</dd>

        <dt>Coach</dt>

        <dd>{cohort.coach}</dd>

        <dt>Trainer</dt>

        <dd>{cohort.trainer}</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;

  vertical-align: top;

}

dt {

  font-weight: 500;

}

App.js

import React from 'react';

import CohortDetails from './components/CohortDetails';

function App() {

  const cohorts = [

    {

      id: 'INTADMDF10 -.NET FSD',

      started: '22-Feb-2022',

      status: 'Scheduled',

      coach: 'Aathma',

      trainer: 'Jojo Jose',

    },

    {

      id: 'ADM21JF014 - Java FSD',

      started: '10-Sep-2021',

      status: 'Ongoing',

      coach: 'Apoorv',

      trainer: 'Elisa Smith',

    },

    {

      id: 'CDBJF21025 - Java FSD',

      started: '24-Dec-2021',

      status: 'Ongoing',

      coach: 'Aathma',

      trainer: 'John Doe',

    },

  ];

  return (

    <div style={{ padding: '20px' }}>

      <h2>Cohorts Details</h2>

      {cohorts.map((cohort, index) => (

        <CohortDetails key={index} cohort={cohort} />

      ))}

    </div>

  );

}

export default App;

Index.js

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<App />);

Output:

