# ReactJS-HOL Handson

**Objective:-**

* Define SPA and its benefits

Ans) A Single Page Application (SPA) is a web application that loads a single HTML page and dynamically updates content as the user interacts with it.

Benefits:

- Faster navigation

- No full page reloads

- Better user experience

- Easier to manage with front-end frameworks like React

* Define React and identify its working

Ans)React is a JavaScript library for building user interfaces. It uses a component-based architecture and a virtual DOM to efficiently update and render UI changes.

* Identify the differences between SPA and MPA

Ans)SPA (Single Page Application):

- Loads only one HTML page

- Fast navigation

- Uses JavaScript for routing

MPA (Multi Page Application):

- Loads new HTML pages for each route

- Slower navigation

- Better for SEO

* Explain about React

Ans) React is a front-end JavaScript library developed by Meta. It is used to build interactive user interfaces using components. React uses a virtual DOM to optimize updates.

* Define virtual DOM

Ans) The Virtual DOM is an in-memory representation of the real DOM. React uses it to efficiently compare changes (diffing) and update only what is necessary in the browser.

* Explain Features of React

Ans) - Component-Based

- Virtual DOM

- JSX Syntax

- One-Way Data Binding

- Fast rendering

- Reusable Components

**Lab Objective) To create a React application named "myfirstreact" that displays:**

**"Welcome to the first session of React" as a heading on the web page**.

**Tools Used**

- Visual Studio Code

- Node.js and npm

- create-react-app

- Google Chrome (or any browser)

**Code):- App.js**

import React from 'react';

function App() {

return (

<div>

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

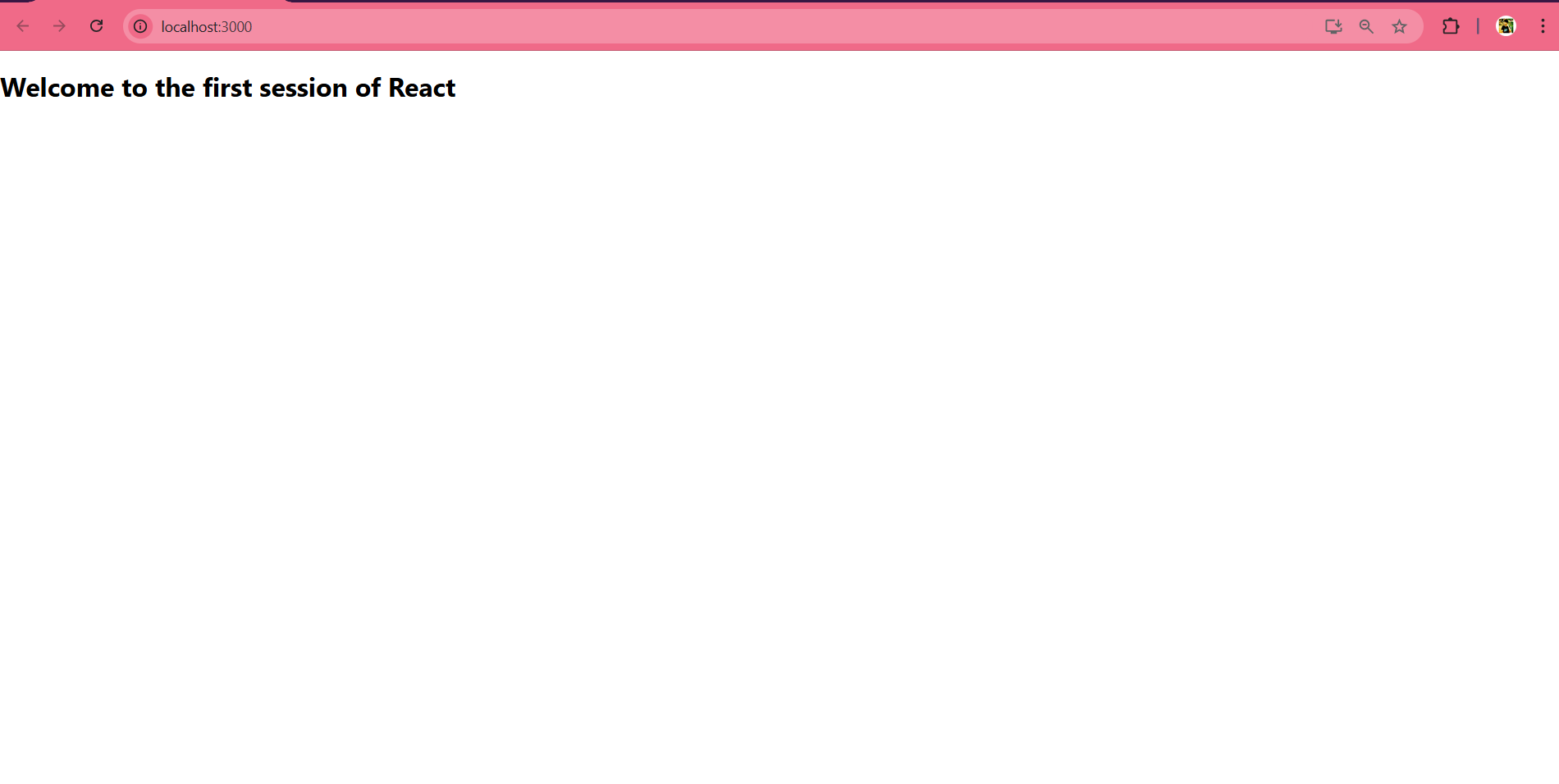
</div>

);

}

export default App;

**Output:-**



# ReactJS-HOL Handson

## **Objectives**

* Explain React components
* Identify the differences between components and JavaScript functions
* Identify the types of components
* Explain class component
* Explain function component
* Define component constructor
* Define render() function

**Tools Used**

- Visual Studio Code

- Node.js and npm

- create-react-app

- Google Chrome (or any browser)

**Code):-**

**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>

      <h1>Student Management Portal</h1>

      <Home />

      <About />

      <Contact />

    </div>

  );

}

export default App;

**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>

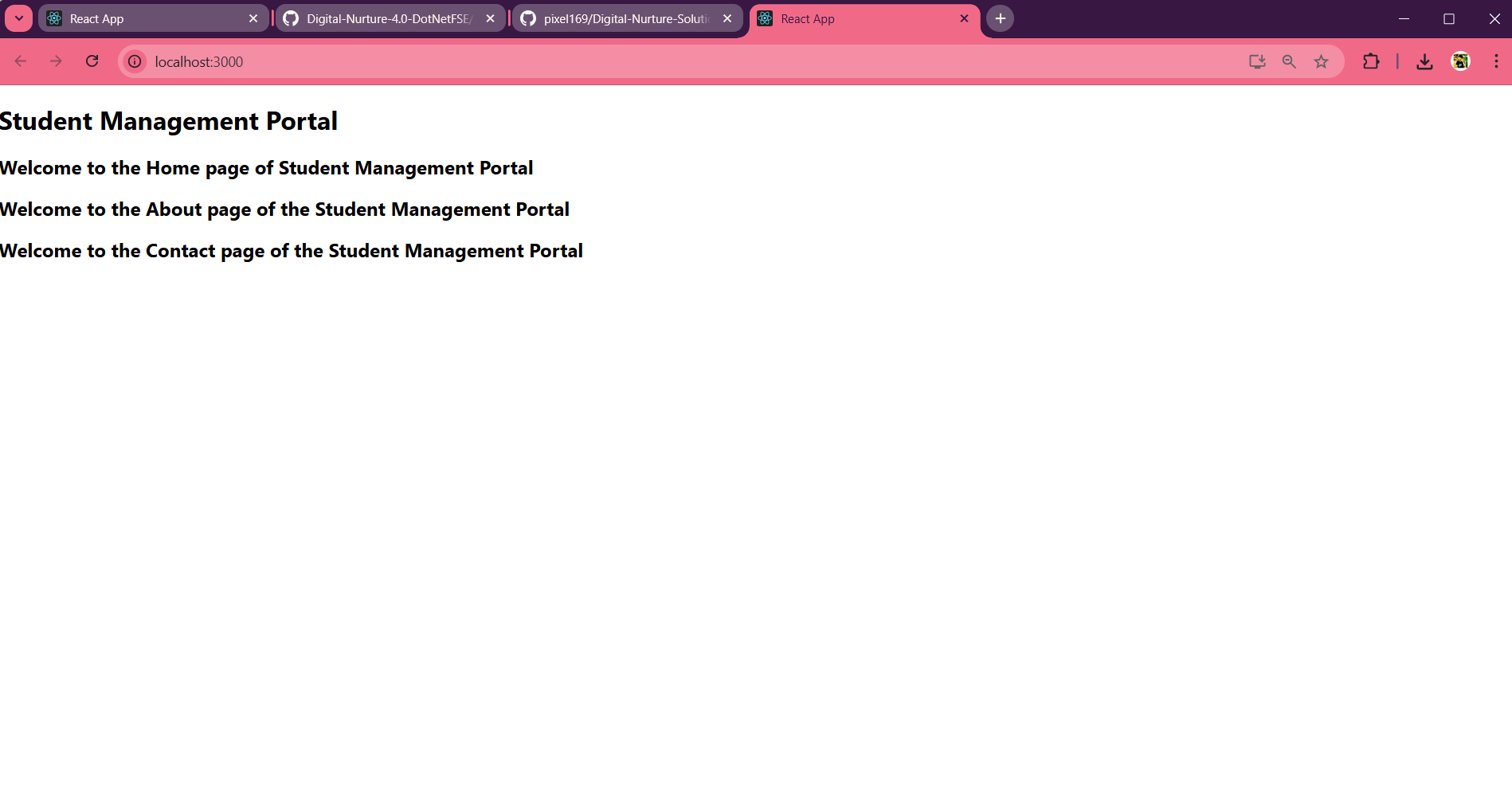
    </div>

  );

}

export default Contact;

**Output:-**



# ReactJS-HOL Handson

## **Objectives**

* Explain React components
* Identify the differences between components and JavaScript functions
* Identify the types of components
* Explain class component
* Explain function component
* Define component constructor
* Define render() function

**Tools Used**

- Visual Studio Code

- Node.js and npm

- create-react-app

- Google Chrome (or any browser)

**CODE:-**

**App.js**

import React from 'react';

import CalculateScore from './components/calculatescore';

function App() {

  return (

    <div>

      <h1>Score Calculator App</h1>

      <CalculateScore

        Name="Asish Sarangi"

        School="Odm Public School"

        Total={450}

        Goal={5}

      />

    </div>

  );

}

export default App;

**Calculatescore.js**

import React from 'react';

import '../stylesheets/mystyle.css';

function CalculateScore(props) {

  const average = props.Total / props.Goal;

  return (

    <div className="score-box">

      <h2>Student Score Report</h2>

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

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

      <p><strong>Total Score:</strong> {props.Total}</p>

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

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

    </div>

  );

}

export default CalculateScore;

**mystyle.css**

.score-box {

  background-color: #f5f5f5;

  padding: 20px;

  margin: 20px auto;

  width: 50%;

  border-radius: 8px;

  box-shadow: 0px 2px 8px rgba(0, 0, 0, 0.1);

  font-family: Arial, sans-serif;

}

.score-box h2 {

  color: #333;

}

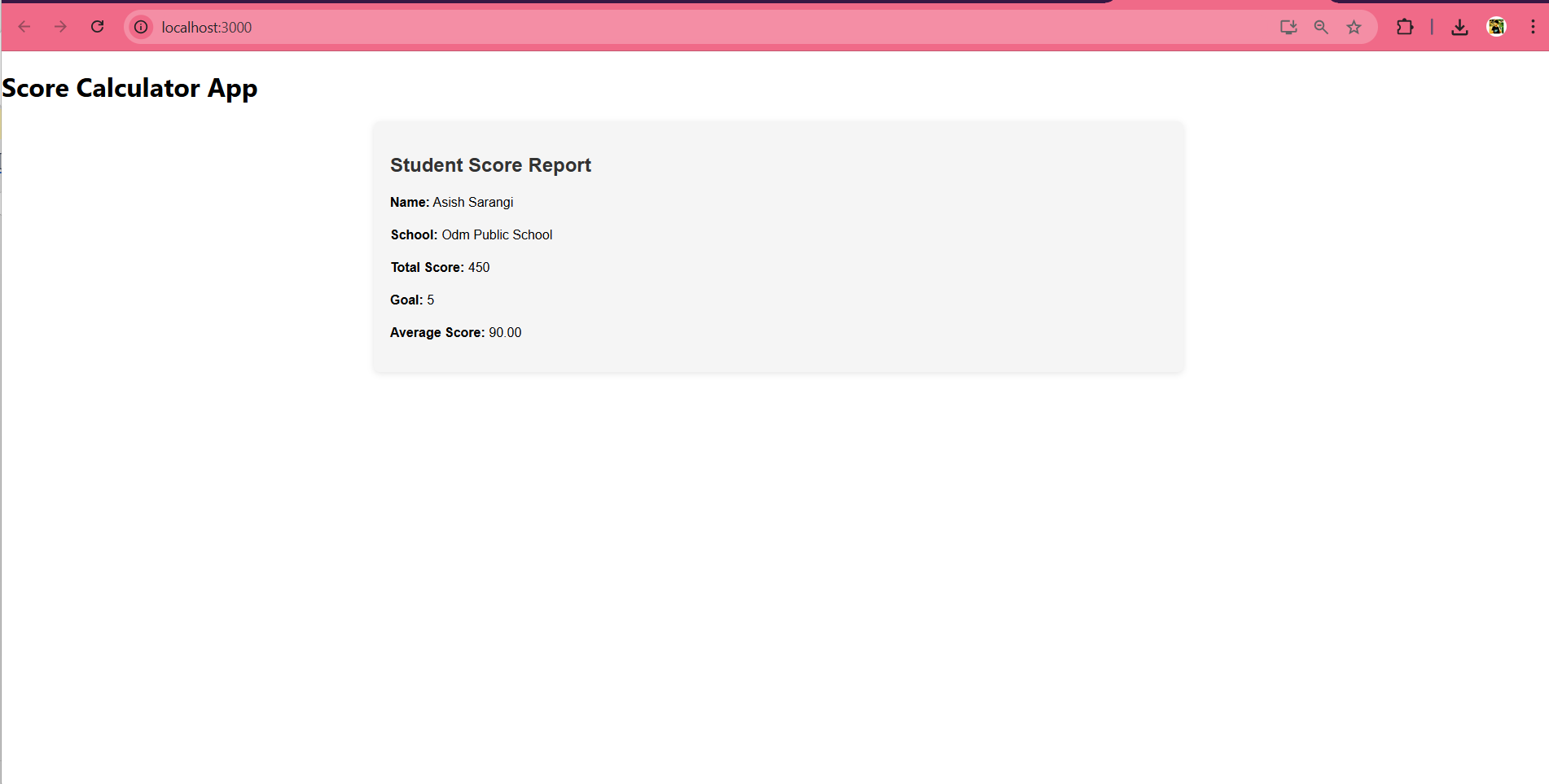
.score-box p {

  font-size: 16px;

  line-height: 1.5;

}

**Ouput:-**



# ReactJS-HOL Handson

**Objectives**

* Explain the need and Benefits of component life cycle
* Identify various life cycle hook methods
* List the sequence of steps in rendering a component

In this hands-on lab, you will learn how to:

* Implement componentDidMount() hook
* Implementing componentDidCatch() life cycle hook.

**Tools Used**

- Visual Studio Code

- Node.js and npm

- create-react-app

- Google Chrome (or any browser)

**Code:-**

**App.js**

import React from 'react';

import Posts from './Posts';

function App() {

  return (

    <div className="App">

      <h1>Welcome to BlogApp</h1>

      <Posts />

    </div>

  );

}

export default App;

**Posts.js**

import React, { Component } from 'react';

import Post from './Post';

class Posts extends Component {

  constructor(props) {

    super(props);

    this.state = {

      posts: [],

      hasError: false

    };

  }

  componentDidMount() {

    this.loadPosts();

  }

 loadPosts() {

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

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

    .then(data => {

      console.log('Fetched posts:', data);

      this.setState({ posts: data.slice(0, 5) });

    })

    .catch(error => {

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

      this.setState({ hasError: true });

    });

}

  componentDidCatch(error, info) {

    alert('Something went wrong: ' + error);

    this.setState({ hasError: true });

  }

  render() {

    if (this.state.hasError) {

      return <h2>Something went wrong while loading posts.</h2>;

    }

    return (

      <div>

        <h2>Blog Posts</h2>

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

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

        ))}

      </div>

    );

  }

}

export default Posts;

**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;

**Output:-**



# ReactJS-HOL Handson

**Objectives**

* Understanding the need for styling react component
* Working with CSS Module and inline styles

**Tools Used**

- Visual Studio Code

- Node.js and npm

- create-react-app

- Google Chrome (or any browser)

**Code:-**

**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;

**CohortDetails.js**

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

function CohortDetails(props) {

    return (

        <div className={styles.box}>

            <h3>

                {props.cohort.cohortCode} -

                <span> {props.cohort.technology}</span>

            </h3>

            <dl>

                <dt>Started On</dt>

                <dd>{props.cohort.startDate}</dd>

                <dt>Current Status</dt>

                <dd style={{ color: props.cohort.currentStatus.toLowerCase() === 'ongoing' ? 'green' : 'blue' }}>

                    {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;

**Cohort.js**

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'),

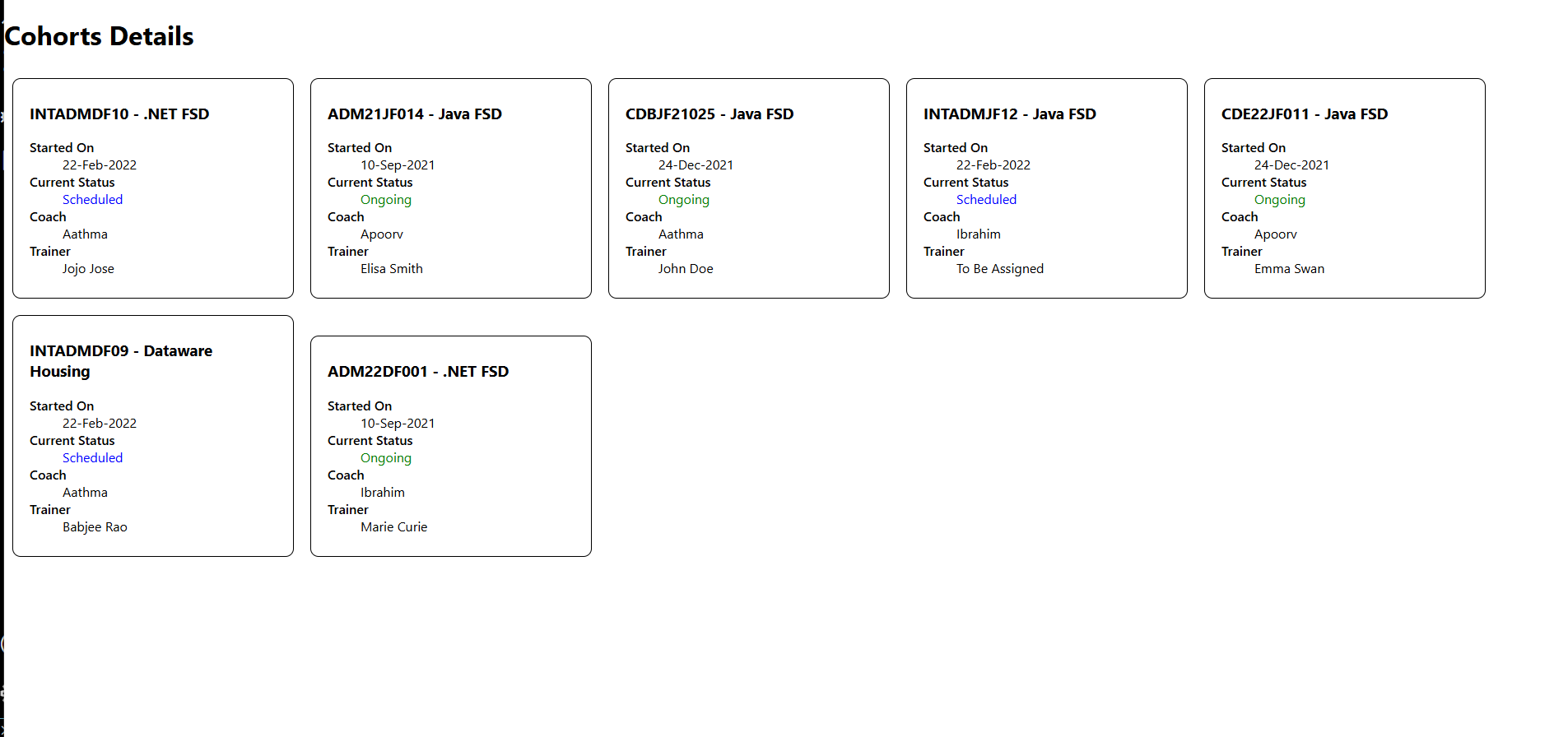
    new Cohort('INTADMDF09','22-Feb-2022', 'Dataware Housing', 'Babjee Rao','Aathma', 'Scheduled'),

    new Cohort('ADM22DF001','10-Sep-2021', '.NET FSD', 'Marie Curie','Ibrahim', 'Ongoing'),

];

export {Cohort, CohortsData};

**Output:-**

****