**Week-6**

**React**

**Exercise 1: Create First React App**

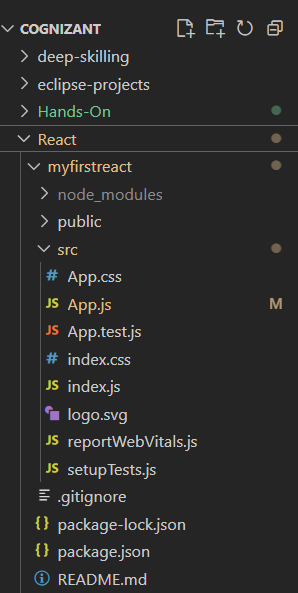
**Steps:**

1. **Create a New React Application:**
   * Install Node.js and use npx create-react-app to generate a new React project named myfirstreact. Then, navigate into the project folder.
2. **Modify the Default Component**
   * Open the project in Visual Studio Code, go to the main component file (App.js), and replace the default content with a heading message like "Welcome to the first session of React".
3. **Run and View the Application:**
   * Start the development server using npm start. Then, open a browser and go to localhost:3000 to view the React application running with your custom heading.

React is a **JavaScript library** developed by Facebook for building **user interfaces**, especially **Single-Page Applications (SPAs)**. It allows developers to build web apps using **components**, which are reusable and manage their own state.

A Single-Page Application is a web application that **loads a single HTML page** and dynamically updates the content without refreshing the entire page. This provides a faster and smoother user experience.

**Folder Structure:**



**App.js**

// App.js

function App() {

  return (

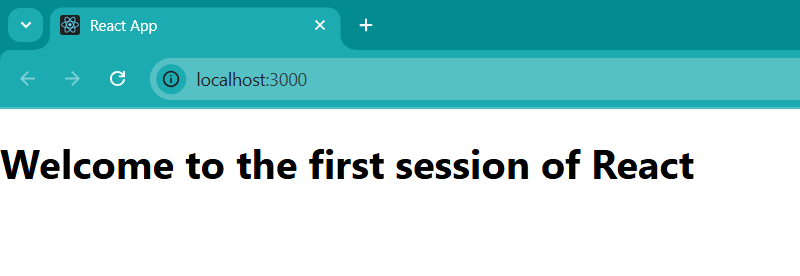
    <h1>Welcome to the first session of React</h1>

  );

}

export default App;

**Output:**

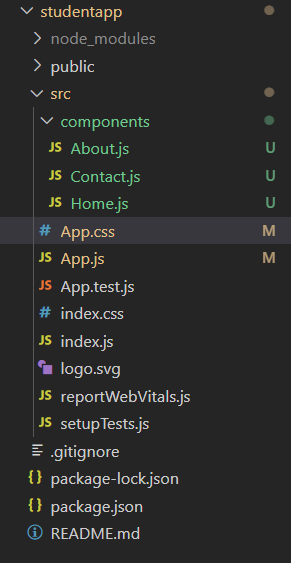


**Exercise 2: Student App**

**Steps:**

1. **Create a New React Application:**
   * Install Node.js and use npx create-react-app to generate a new React project named studentapp. Then, navigate into the project folder.
2. **Modify and add new files:**
   * Open the project and create files likes home.js, about.js, contact.js and modify app.js add app.css.
3. **Run and View the Application:**
   * Start the development server using npm start. Then, open a browser and go to localhost:3000 to view the React application running with your custom heading.

**Folder Structure:**



**App.js**

// 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='Container'>

      <Home/>

      <About/>

      <Contact/>

    </div>

  );

}

export default App;

**App.css**

// App.css

.Container

{

  display: flex;

  flex-direction: column;

  align-items: center;

  justify-content: center;

  height: 100vh;

  background-color: #f0f0f0;

}

**Home.js**

// Home.js

import React,{Component} from "react";

class Home extends Component {

    render() {

        return (

            <div className="home">

                <h1>Welcome to the Student App</h1>

                <p>This is the home page where you can find various resources and information.</p>

            </div>

        );

    }

}

export default Home;

**About.js**

// About.js

import React,{Component} from "react";

class About extends Component {

    render() {

        return (

            <div className="about">

                <h1>About Us</h1>

                <p>This application is designed to help students manage their studies and resources effectively.</p>

            </div>

        );

    }

}

export default About;

**Contact.js**

// Contact.js

import React,{Component} from "react";

class Contact extends Component {

    render() {

        return (

            <div className="contact">

                <h1>Contact Us</h1>

                <p>If you have any questions or need assistance, please reach out to us at:</p>

                <p>Email: cognizant@gmail.com  </p>

                <p>Phone: +91 1234567890</p>

                <p>We are here to help you with any inquiries you may have.</p>

            </div>

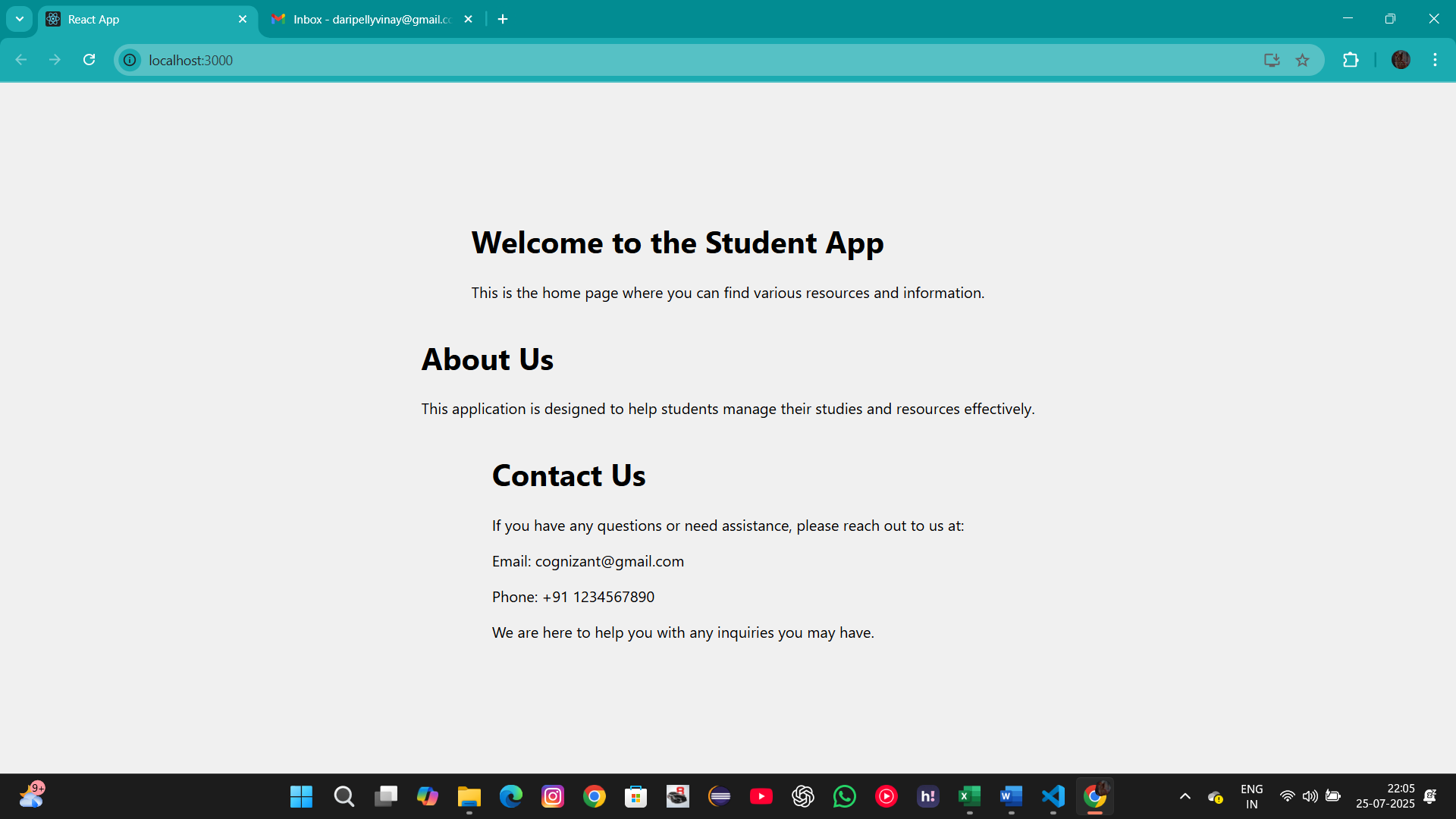
        );

    }

}

export default Contact;

**Output:**

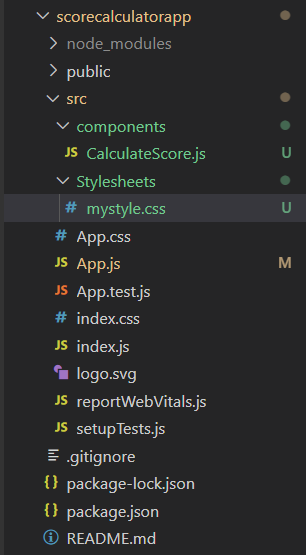


**Exercise 3: Score Calculator App**

**Steps:**

1. **Create a New React Application:**
   * Install Node.js and use npx create-react-app to generate a new React project named scorecalculatorapp. Then, navigate into the project folder.
2. **Modify and add new files:**
   * Open the project and create files likes app.js, calculatorscore.js and mystyle.css.
3. **Run and View the Application:**
   * Start the development server using npm start. Then, open a browser and go to localhost:3000 to view the React application running with your custom heading.

**Folder Structure:**



**App.js**

// App.js

import './App.css';

import React from 'react';

import { CalculateScore } from './components/CalculateScore';

function App() {

  return (

    <div className="App">

      <CalculateScore

        Name="Arikanti Ruthvika"

        School="VNR VJIET"

        total={98}

        goal={100}

      />

    </div>

  );

}

export default App;

**Calculatescore.js**

// Calculatescore.js

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

const percentToDecimal= (decimal) => { return (decimal.toFixed(2) + '%')}

const calcScore = (total, goal) => { return percentToDecimal(total/goal)}

export const CalculateScore = ({Name, School, total, goal}) => (

<div className="formatstyle">

    <h1><font color="Brown">Student Details:</font></h1>

    <div className="Name">

        <b> <span> Name: </span> </b>

        <span>{Name}</span>

    </div>

    <div className="School">

        <b> <span> School: </span> </b>

         <span>{School}</span>

    </div>

    <div className="Total">

        <b><span>Total:</span> </b>

        <span>{total}</span>

        <span>Marks</span>

    </div>

    <div className="Score">

    <b>Score:</b>

    <span>

        {calcScore(

            total,

            goal

        )}

    </span>

    </div>

</div>)

**mystyle.css**

// mystyle.css

.Name

{

font-weight:300;

color:blue;

}

School

{

color:crimson;

}

.Total

{

color:darkmagenta;

}

.formatstyle

{

text-align:center;

font-size: large;

}

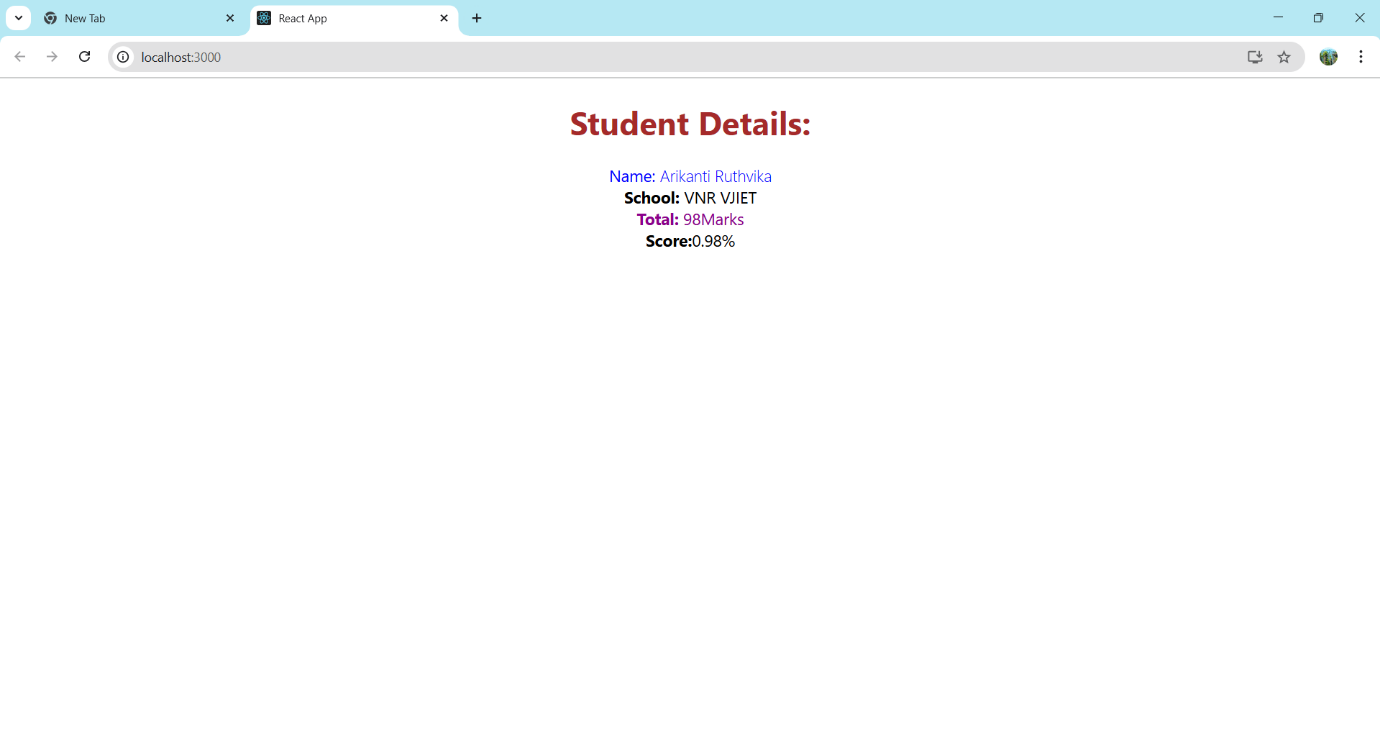
Score

{

color :forestgreen;

}

**Output:**

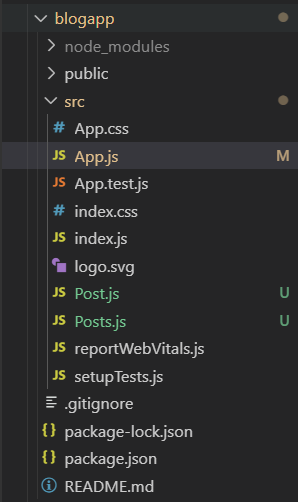


**Exercise 4: Blog App**

**Steps:**

1. **Create a New React Application:**
   * Install Node.js and use npx create-react-app to generate a new React project named blogapp. Then, navigate into the project folder.
2. **Modify and add new files:**
   * Open the project and create files likes app.js, post.js and posts.js.
3. **Run and View the Application:**
   * Start the development server using npm start. Then, open a browser and go to localhost:3000 to view the React application running with your custom heading.

**Folder Structure:**



**App.js**

// App.js

import './App.css';

import Posts from './Posts';

function App() {

  return (

    <div className="App">

      <Posts />

    </div>

  );

}

export default App;

**Post.js**

// Post.js

class Post{

    constructor(id,title, body) {

        this.id = id;

        this.title = title;

        this.body = body;

    }

}

export default Post;

**Posts.js**

// Posts.js

import React, { Component } from 'react';

import Post from './Post';

class Posts extends Component {

  constructor(props) {

    super(props);

    this.state = {

      posts: []

    };

  }

  loadPosts = () => {

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

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

      .then(data => {

        const postList = data.map(item => new Post(item.id, item.title, item.body));

        this.setState({ posts: postList });

      })

      .catch(error => {

        alert("Failed to load posts: " + error.message);

      });

  };

  componentDidMount() {

    this.loadPosts();

  }

  componentDidCatch(error, info) {

    alert("An error occurred: " + error.toString());

  }

  render() {

    return (

      <div>

        <h1>Posts</h1>

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

          <div key={post.id}>

            <h2>{post.title}</h2>

            <p>{post.body}</p>

          </div>

        ))}

      </div>

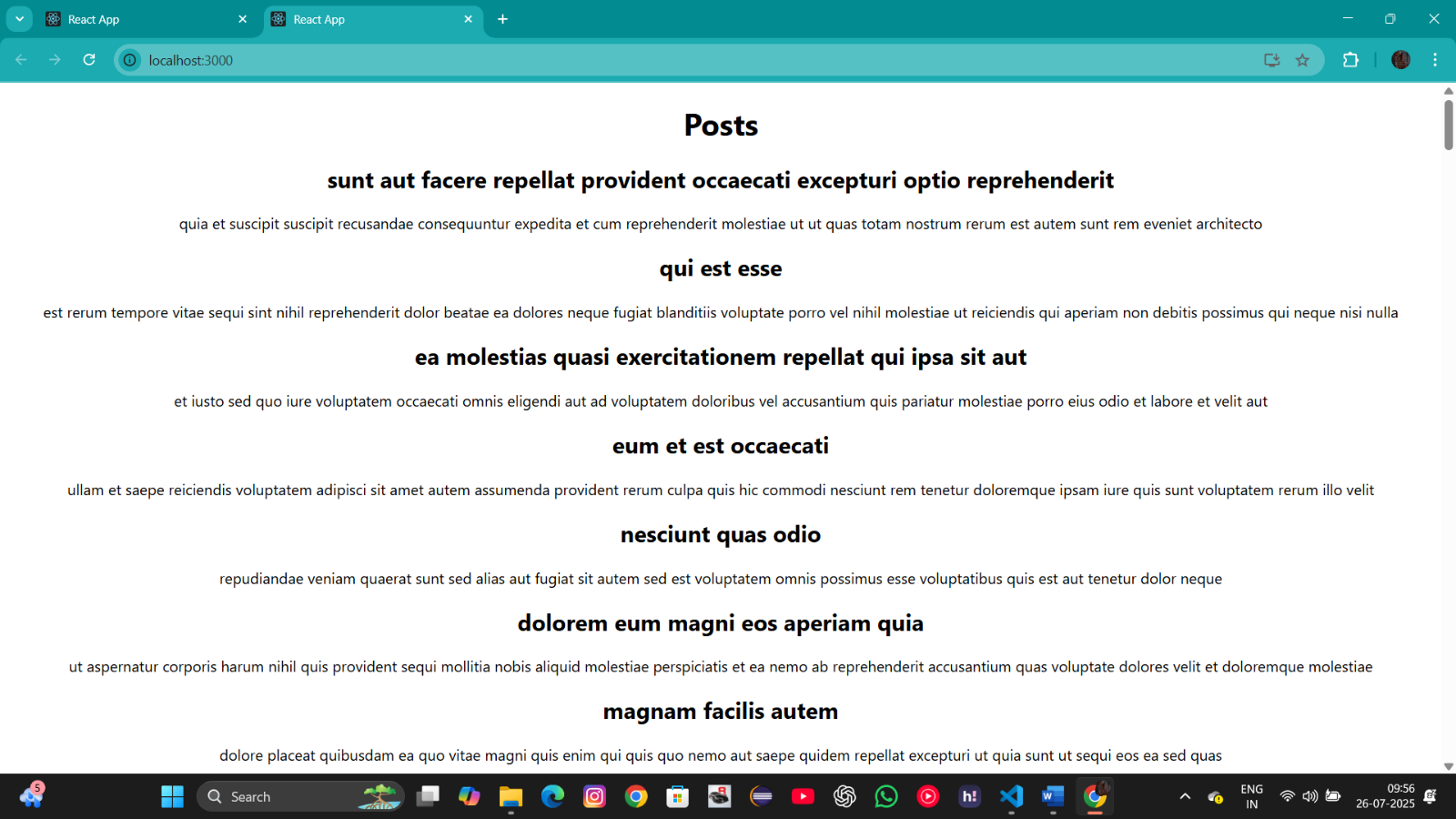
    );

  }

}

export default Posts;

**Output:**

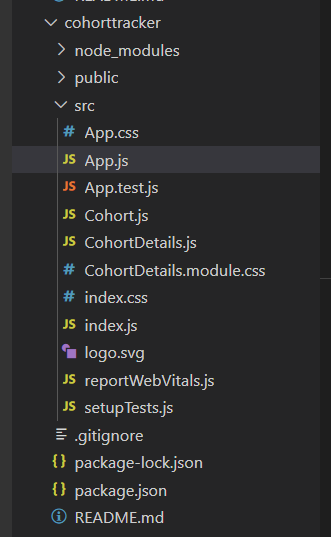


**Exercise 5: Cohort Tracker App**

**Steps:**

1. **Extract a the React Application:**
   * Extract the react app and make modifications to the files to the app cohorttracker.
2. **Modify the new files:**
   * Open the project files likes app.js, CohortDetails.js and CohortDetails.module.css.
3. **Run and View the Application:**
   * Start the development server using npm start. Then, open a browser and go to localhost:3000 to view the React application running with your custom heading.

**Folder Structure:**



**App.js**

// App.js

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.module.css**

//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;

}

**CohortDetails.js**

//CohortDetails.js

import React from 'react';

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

function CohortDetails(props) {

    const status = props.cohort.currentStatus.toLowerCase();

    const titleColor = status === 'ongoing' ? 'green' : 'blue';

    return (

        <div className={styles.box}>

            <h3 style={{ color: titleColor }}>

                {props.cohort.cohortCode} -

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

            </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>

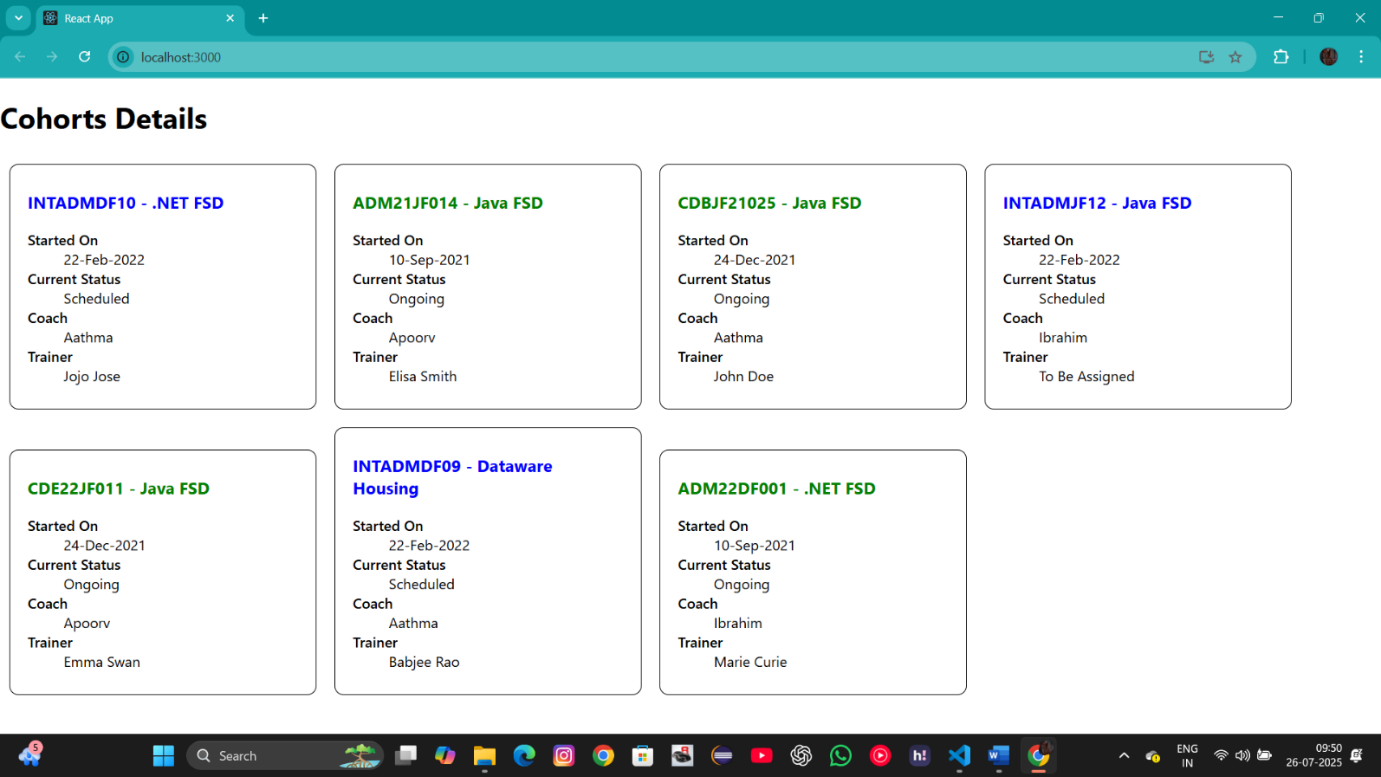
        </div>

    );

}

export default CohortDetails;

**Output:**



**Conclusion:**

In conclusion, These exercises introduced core React and JavaScript concepts for building interactive UIs. They demonstrated how components, state, and styling work together to create responsive and modular web applications.

**-- THE END --**