**Module 10 – Single Page Application framework – React**

**5.ReactJS-HOL**

**Understanding the need for styling React components**

React components manage UI, and styling them is essential to:

* Make the UI user-friendly and visually appealing.
* Highlight important information (e.g., changing color based on status).
* Maintain consistency across components.
* Improve user experience with responsive and readable layouts.

In this lab, we styled components like CohortDetails to present data in a clean, readable format using both CSS Modules and inline styles.

**Working with CSS Modules and inline styles**

CSS Modules and inline styles help manage component-specific styles:

* **CSS Modules** ensure:
  + Class names are scoped locally (avoid conflicts).
  + Reusable and maintainable styling structure.
* **Inline styles** allow:
  + Dynamic styling based on component logic.
  + Quick styling changes inside JS files (e.g., conditional color based on cohort status).

**Style a React component**

I learned how to apply styles to a React component (CohortDetails) by:

* Importing a .module.css file.
* Creating a clean layout using div, h3, and dl elements.

**Define styles using the CSS Module**

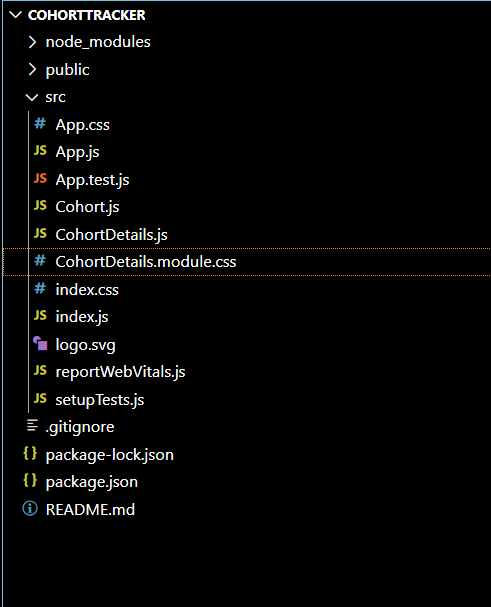
I created a file CohortDetails.module.css and defined the .box class for:

* Setting width, padding, margin, border, and border-radius.
* Styling <dt> elements with bold text.

**Apply styles using className and style properties**

* Used className={styles.box} to apply styles from the CSS module.
* Used inline style like style={{ color: status === 'Ongoing' ? 'green' : 'blue' }} for dynamic color based on status.

**Structure:**



**App.css:**

.App {

  text-align: center;

}

.App-logo {

  height: 40vmin;

  pointer-events: none;

}

@media (prefers-reduced-motion: no-preference) {

  .App-logo {

    animation: App-logo-spin infinite 20s linear;

  }

}

.App-header {

  background-color: #282c34;

  min-height: 100vh;

  display: flex;

  flex-direction: column;

  align-items: center;

  justify-content: center;

  font-size: calc(10px + 2vmin);

  color: white;

}

.App-link {

  color: #61dafb;

}

@keyframes App-logo-spin {

  from {

    transform: rotate(0deg);

  }

  to {

    transform: rotate(360deg);

  }

}

**App.js:**

import React from "react";

import cohortData from "./Cohort";

import CohortDetails from "./CohortDetails";

function App() {

  return (

    <div style={{ padding: "20px", border: "1px solid black", margin: "30px" }}>

      <h2><b>Cohorts Details</b></h2>

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

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

      ))}

    </div>

  );

}

**App.test.js:**

export default App;

import { render, screen } from '@testing-library/react';

import App from './App';

test('renders learn react link', () => {

  render(<App />);

  const linkElement = screen.getByText(/learn react/i);

  expect(linkElement).toBeInTheDocument();

});

**Cohort.js:**

const cohortData = [

  {

    id: "INTADMDF10",

    name: "-.NET FSD",

    startDate: "22-Feb-2022",

    status: "Scheduled",

    coach: "Aathma",

    trainer: "Jojo Jose"

  },

  {

    id: "ADM21JF014",

    name: "-Java FSD",

    startDate: "10-Sep-2021",

    status: "Ongoing",

    coach: "Apoorv",

    trainer: "Elisa Smith"

  },

  {

    id: "CDBJF21025",

    name: "-Java FSD",

    startDate: "24-Dec-2021",

    status: "Ongoing",

    coach: "Aathma",

    trainer: "John Doe"

  }

];

export default cohortData;

**CohortDetails.js:**

import React from "react";

import styles from "./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;

}

const CohortDetails = ({ cohort }) => {

  const { id, name, startDate, status, coach, trainer } = cohort;

  const titleStyle = {

    color: status.toLowerCase() === "ongoing" ? "green" : "blue",

    fontWeight: "bold"

  };

  return (

    <div className={styles.box}>

      <h3 style={titleStyle}>

        {id} {name}

      </h3>

      <dl>

        <dt>Started On</dt>

        <dd>{startDate}</dd>

        <dt>Current Status</dt>

        <dd>{status}</dd>

        <dt>Coach</dt>

        <dd>{coach}</dd>

        <dt>Trainer</dt>

        <dd>{trainer}</dd>

      </dl>

    </div>

  );

};

export default CohortDetails;

**index.css:**

body {

  margin: 0;

  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',

    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',

    sans-serif;

  -webkit-font-smoothing: antialiased;

  -moz-osx-font-smoothing: grayscale;

}

code {

  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',

    monospace;

}

**index.js:**

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import App from './App';

import reportWebVitals from './reportWebVitals';

ReactDOM.render(

  <React.StrictMode>

    <App />

  </React.StrictMode>,

  document.getElementById('root')

);

reportWebVitals();

**reportWebVitals.js:**

const reportWebVitals = onPerfEntry => {

  if (onPerfEntry && onPerfEntry instanceof Function) {

    import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {

      getCLS(onPerfEntry);

      getFID(onPerfEntry);

      getFCP(onPerfEntry);

      getLCP(onPerfEntry);

      getTTFB(onPerfEntry);

    });

  }

};

export default reportWebVitals;

**setupTests.js:**

import '@testing-library/jest-dom';

**Output:**

