**5.ReactJs-Hol**

**Objectives**

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

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

* Style a react component
* Define styles using the CSSModule
* Apply styles to components using className and style properties

## **Prerequisites**

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

## **Notes**

Estimated time to complete this lab: **30 minutes.**

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.
5. Final result should look similar to the below image



Figure 2: Final Result

CohortDetails.module.css

/\* CohortDetails.module.css \*/

/\* This is a CSS Module.

  Class names defined here (like .box) are scoped locally to the component that imports this file.

  This prevents style conflicts with other components.

\*/

/\* Style for the main container div.

  - width: Sets a fixed width for the card.

  - display: inline-block allows multiple cards to sit side-by-side.

  - margin: Provides space around the entire card.

  - padding: Creates space between the content and the border.

  - border: Adds a thin, black border.

  - border-radius: Rounds the corners of the card.

\*/

.box {

  width: 300px;

  display: inline-block;

  margin: 10px;

  padding: 10px 20px; /\* 10px for top/bottom, 20px for left/right \*/

  border: 1px solid black;

  border-radius: 10px;

  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1); /\* Added a subtle shadow for better visuals \*/

  vertical-align: top; /\* Ensures alignment when heights differ \*/

}

/\*

  This is a global tag selector.

  By default, tag selectors in CSS Modules are also locally scoped.

  To make them global, you would use :global(dt).

  However, for this assignment, a local scope is fine.

  It targets all <dt> (Definition Term) elements within the component.

\*/

dt {

  font-weight: 500;

}

CohortDetails.js

// src/components/CohortDetails.js

import React from 'react';

// Step 1: Import the CSS Module.

// 'styles' will be an object where keys are the class names from the CSS file.

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

// This component receives cohort data via props.

const CohortDetails = ({ cohort }) => {

  // Step 2: Define the conditional style for the h3 element.

  // This is an inline style object. React uses camelCase for CSS property names.

  // We check the 'status' property of the cohort object.

  const headerStyle = {

    color: cohort.status === 'Ongoing' ? 'green' : 'blue'

  };

  return (

    // Step 3: Apply the 'box' class from the CSS module to the container div.

    // We access the class name as a property of the imported 'styles' object.

    // This becomes a unique class name in the browser like 'CohortDetails\_box\_\_12345'

    <div className={styles.box}>

      {/\* Step 4: Apply the conditional inline style to the h3 element. \*/}

      <h3 style={headerStyle}>

        {cohort.id} - {cohort.name}

      </h3>

      {/\* The rest of the component structure \*/}

      <dl>

        <dt>Started On</dt>

        <dd>{cohort.startedOn}</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;

/\*

  NOTE: This component assumes it receives a 'cohort' object as a prop.

  For this to work, the parent component (e.g., App.js) would look something like this:

  import CohortDetails from './components/CohortDetails';

  function App() {

    const cohorts = [

      { id: 'INTADMF10', name: '.NET FSD', startedOn: '22-Feb-2022', status: 'Scheduled', coach: 'Aathma', trainer: 'Jojo Jose' },

      { id: 'ADM21JF014', name: 'Java FSD', startedOn: '10-Sep-2021', status: 'Ongoing', coach: 'Apoorv', trainer: 'Elisa Smith' },

      { id: 'CDBJF21025', name: 'Java FSD', startedOn: '24-Dec-2021', status: 'Ongoing', coach: 'Aathma', trainer: 'John Doe' }

    ];

    return (

      <div>

        <h1>Cohorts Details</h1>

        {cohorts.map(c => <CohortDetails key={c.id} cohort={c} />)}

      </div>

    );

  }

  export default App;

\*/