**Hands-on in this document**

**Objectives**

**1. Understanding the need for styling react component**

Visual Appeal: Makes the UI look clean and professional.

Better UX: Improves readability, spacing, and interaction.

Organized Layout: Separates and structures content clearly.

Scoped Styling: Avoids CSS conflicts using modules or inline styles.

Dynamic Styling: Allows style changes based on data (e.g., green if status is "Ongoing").

**2. Working with CSS Module and inline styles**

**CSS Modules:**

* Local scoped styles (class names don’t conflict globally).
* File ends with .module.css.
* Import like:

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

* Use in JSX:

<div className={styles.box}>...</div>

**Inline Styles:**

* Styles written directly in JSX as JavaScript objects.
* Example:

<h3 style={{ color: cohort.status === 'Ongoing' ? 'green' : 'blue' }}>

{cohort.name}

</h3>

**Code :**

**-- 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**

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

function CohortDetails(props) {

    const headingStyle = {

        color: props.cohort.currentStatus === 'ongoing' ? 'green' : 'blue',

    };

    return (

        <div className={styles.box}>

            <h3 style={headingStyle}>

                {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 :**

