**5. ReactJS-HOL**

**CohortDetails.js**

import React from 'react';

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

function CohortDetails({ cohort }) {

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

const titleStyle = {

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

};

return (

<div className={styles.box}>

<h3 style={titleStyle}>{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;

**CohortDetails.module.css**

.box {

width: 300px;

display: inline-block;

margin: 10px;

padding: 10px 20px;

border: 1px solid #999;

border-radius: 10px;

vertical-align: top;

}

dt {

font-weight: 500;

margin-top: 10px;

}

**App.js**

import React from 'react';

import CohortDetails from './components/CohortDetails';

function App() {

const cohorts = [

{

name: "INTADMDF10 - .NET FSD",

startDate: "22-Feb-2022",

status: "Scheduled",

coach: "Aathma",

trainer: "Jojo Jose"

},

{

name: "ADM21JF014 - Java FSD",

startDate: "10-Sep-2021",

status: "Ongoing",

coach: "Apoorv",

trainer: "Elisa Smith"

},

{

name: "CDBJF21025 - Java FSD",

startDate: "24-Dec-2021",

status: "Ongoing",

coach: "Aathma",

trainer: "John Doe"

}

];

return (

<div style={{ padding: '20px' }}>

<h1>Cohorts Details</h1>

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

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

))}

</div>

);

}

export default App;

**Index.js**

import React from 'react';

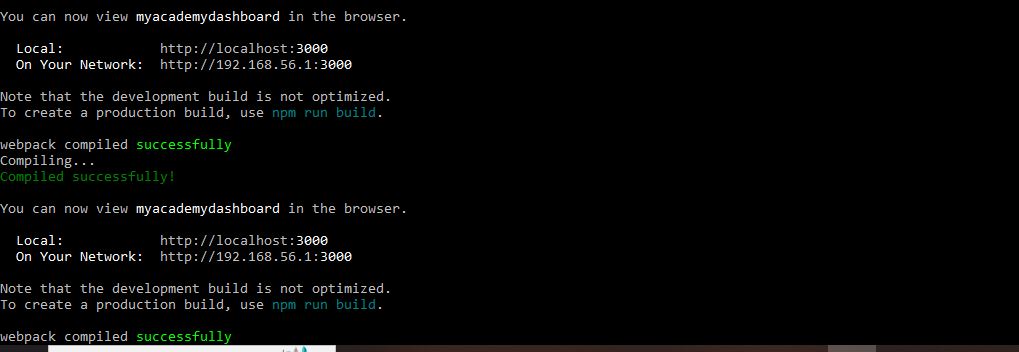
import ReactDOM from 'react-dom/client';

import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<App />);

****

****

**1. Understanding the need for styling React components**

Styling in React enhances the UI by improving readability, structure, and user experience. It helps developers:

* Visually distinguish components
* Reflect the application's brand
* Maintain modular and reusable styles using scoped CSS like CSS Modules

**2. Working with CSS Modules and Inline Styles**

**CSS Modules** allow defining scoped styles. They avoid global CSS conflicts.

**Inline styles** are applied directly in JSX using JavaScript objects. Useful for dynamic styling (e.g., based on a condition).