**5. ReactJS-HOL**

**• Understanding the need for styling React component**

In React, styling is essential for making components visually appealing, user-friendly, and consistent across the application.

Why styling is needed:

* To improve user experience and visual clarity.
* To match branding or theme of the application.
* To organize components visually on the screen.
* To reflect dynamic states (e.g., active, disabled).

React supports multiple styling approaches:

* Traditional CSS
* Inline styles
* CSS Modules
* Styled-components (CSS-in-JS)
* SCSS and other preprocessors

**• Working with CSS Module and inline styles**

1. CSS Modules

CSS Modules help write locally scoped styles. This avoids conflicts between class names across different components.

Steps to use CSS Module:

1. Create a CSS file with .module.css extension.
2. Import the styles into your React component.
3. Use className={styles.className} syntax.

Example:

css

/\* Button.module.css \*/

.button {

background-color: blue;

color: white;

}

jsx

import styles from './Button.module.css';function Button() {

return <button className={styles.button}>Click Me</button>;

}

2. Inline Styles

Inline styles are written directly in the component using a JavaScript object.

Syntax:

* Style values must be in camelCase.
* Styles are passed using the style attribute.

Example:

jsx

function Heading() {

const styleObj = {

color: 'green',

fontSize: '24px',

textAlign: 'center'

};

return <h1 style={styleObj}>Hello, React!</h1>;

}