### **1. Understanding the Need for Styling React Components**

* Styling makes the UI attractive and user‑friendly.
* Each React component is reusable, so styles must be modular and isolated.
* Prevents style conflicts between components.
* Improves readability, layout, and user experience.
* Supports multiple methods: CSS Modules, Inline Styles, CSS‑in‑JS.

### **2. Working with CSS Module and Inline Styles**

**CSS Modules**

* Provides scoped styles only to the component that imports it.
* Prevents class name conflicts in large projects.
* Uses .module.css files imported in the component.
* Example: className={styles.className}

**Inline Styles**

* Styles are written directly inside the component using the style attribute.
* Written as a JavaScript object with camelCase properties.
* Useful for dynamic styling (style based on state or props).
* Example: style={{ backgroundColor: 'red', color: 'white' }}