1)Explain the need for styling react component?

In React applications, styling is essential to build visually appealing and user-friendly interfaces. While React handles the structure and logic of a component, styling is needed to:

* Control layout, colors, and appearance
* Maintain consistency across components

Support responsiveness across devices

* Improve user experience (UX)

React components are reusable building blocks, and applying styles ensures they look and behave as expected in different parts of the application.

Working with CSS Modules and Inline Styles

React supports different styling methods. Two commonly used approaches are CSS Modules and Inline Styles.

2) Explain Working with CSS Module and inline styles?

CSS Modules

CSS Modules allow us to write scoped CSS — styles that apply only to a specific component. This prevents naming conflicts across large applications.

it Works:

1. Create a CSS file with the naming pattern: ComponentName.module.css.
2. Import the styles in the React component.
3. Use the styles as JavaScript objects.

Example:

Message.module.css

.success {

color: green;

font-weight: bold;

}

Message.js

import React from 'react';

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

function Message() {

return <p className={styles.success}>Operation successful!</p>;

}

export default Message;

React automatically creates unique class names to avoid global CSS conflicts.

Inline Styles

Inline styles are written directly inside the component using a JavaScript object. This method is useful for:

* Applying dynamic styles
* Styling based on props or state

Example:

import React from 'react';

function Message() {

const style = {

color: 'blue',

backgroundColor: 'lightgray',

padding: '10px'

}

return <p style={style}>This is an inline styled message.</p>;

}

export default Message;