**React Handson - 5**

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

Styling React components is essential for creating visually appealing, consistent, and user-friendly interfaces. Just like in traditional web development, styles in React control the layout, colors, typography, spacing, and overall look and feel of the application. In React, styling helps in:

• Enhancing user experience

• Establishing visual hierarchy

• Making components reusable and theme-friendly

• Ensuring responsiveness across devices

React offers multiple styling approaches such as traditional CSS, CSS Modules, inline styles, styled components, and third-party libraries, giving developers flexibility based on project needs.

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

CSS Modules and inline styles are two common methods to style components in React:

CSS Modules:

• CSS Modules are locally scoped CSS files that prevent class name collisions.

• They allow styles to be applied only to specific components by using a unique class naming convention.

• Useful for large-scale applications where component isolation is critical.

Example :

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

function Button() {

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

}

Inline Styles:

• Inline styles are defined directly within the JSX using JavaScript objects.

• They are useful for dynamic styling and when styles depend on component logic.

• CSS properties must be written in camelCase syntax.

Example :

function Heading() {

const style = {

color: 'blue',

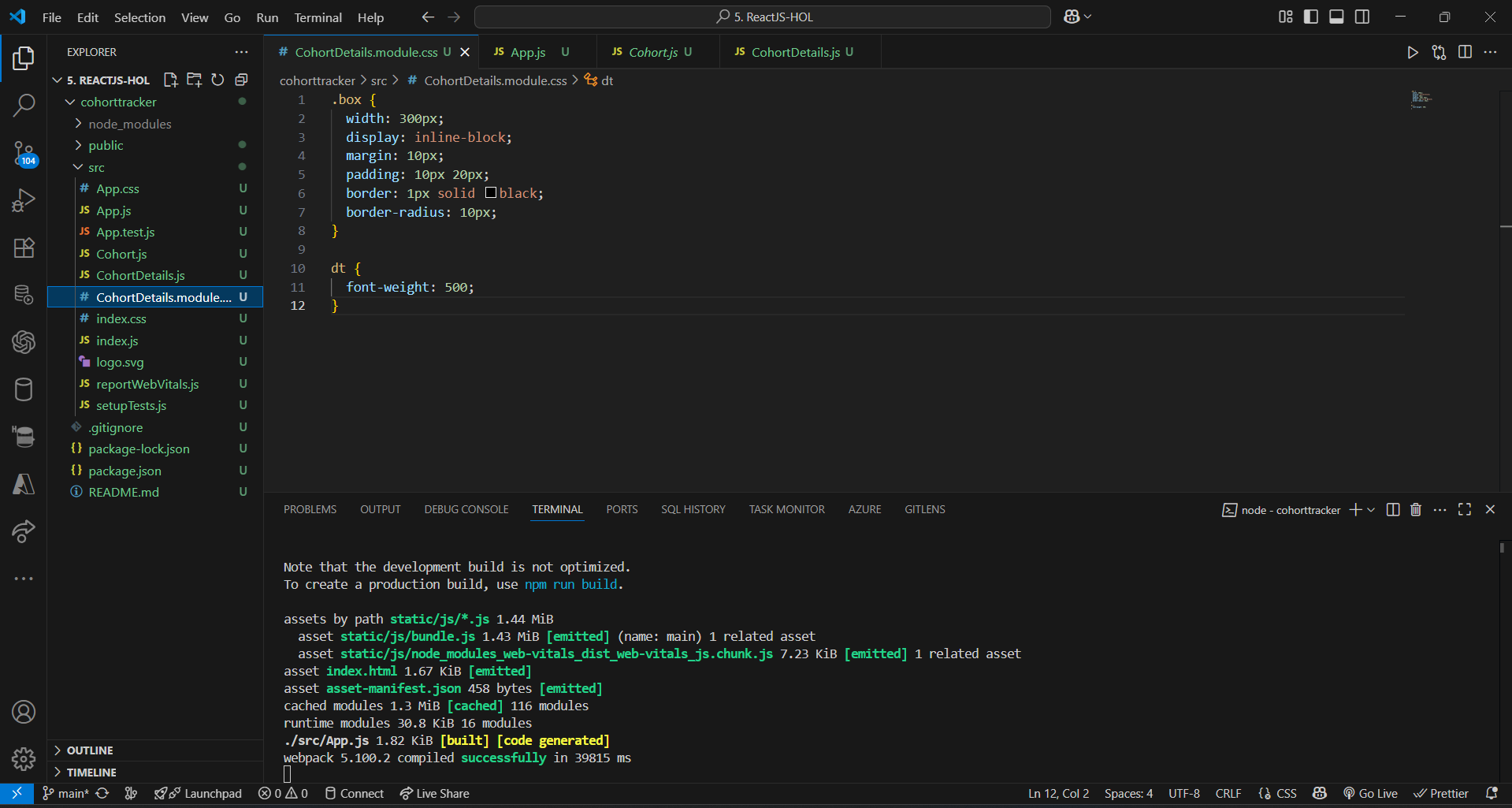
fontSize: '24px'

};

return <h1 style={style}>Welcome</h1>;

}

**3.Codes**

****

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**4.Output**

**A screenshot of a computer

AI-generated content may be incorrect.**