**Assignment 6: Styling in React**

**Style your React components using CSS.**

**Explore popular styling methodologies like CSS-in-JS.**

Styling in React can be done in various ways, and one popular approach is to use CSS. Additionally, CSS-in-JS is a methodology that involves writing styles directly in JavaScript files. Here, I'll provide a brief overview of both approaches.

Styling with CSS:

Separate CSS Files:

Create a separate CSS file for each component or a global CSS file. For example, if you have a Button component:

| **// Button.js**  **import React from 'react';**  **import './Button.css';**  **const Button = ({ text }) => {**  **return <button className="custom-button">{text}</button>;**  **};**  **export default Button;** |
| --- |

| /\* Button.css \*/  .custom-button {  background-color: #3498db;  color: #fff;  padding: 10px 20px;  border: none;  border-radius: 5px;  cursor: pointer;  } |
| --- |

Inline Styles:

You can also use inline styles directly in your components:

| const buttonStyle = {  backgroundColor: '#3498db',  color: '#fff',  padding: '10px 20px',  border: 'none',  borderRadius: '5px',  cursor: 'pointer',  };  const Button = ({ text }) => {  return <button style={buttonStyle}>{text}</button>;  }; |
| --- |

CSS-in-JS:

1. Styled-components:

Use the popular library styled-components to write CSS directly in your JavaScript files.

npm install styled-components

| // Button.js  import React from 'react';  import styled from 'styled-components';  const CustomButton = styled.button`  background-color: #3498db;  color: #fff;  padding: 10px 20px;  border: none;  border-radius: 5px;  cursor: pointer;  `;  const Button = ({ text }) => {  return <CustomButton>{text}</CustomButton>;  };  export default Button; |
| --- |

2.Emotion:

Another popular library for CSS-in-JS is emotion.

**npm install @emotion/styled @emotion/react**

| // Button.js  import React from 'react';  import { styled } from '@emotion/styled';  const CustomButton = styled.button`  background-color: #3498db;  color: #fff;  padding: 10px 20px;  border: none;  border-radius: 5px;  cursor: pointer;  `;  const Button = ({ text }) => {  return <CustomButton>{text}</CustomButton>;  };  export default Button; |
| --- |

Choose the approach that best fits your project and team preferences. Both traditional CSS and CSS-in-JS have their pros and cons, so consider factors like maintainability, scalability, and team familiarity when making a decision.