Styling Components

Terms

CSS-in-JS
CSS modules
Implementation details
Interface
Inline styles
Modular
Separation of concerns
Vanilla CSS

Summary

- We have several options for styling React components, including vanilla CSS, CSS modules, CSS-in-JS, and inline styles.
- With vanilla CSS, we write our component styles in a separate CSS file and import it into the component file. However, we may encounter conflicts if the same CSS classes are defined in multiple files.
- CSS modules resolve this issue by generating unique class names during the build process.
- With CSS-in-JS, we define all the styles for a component alongside its code. Like CSS modules, this provides scoping for CSS classes and eliminates conflicts. It also makes it easier for us to change or delete a component without affecting other components.
- The separation of concerns principle suggests that we divide a program into distinct sections or modules where each section handles a specific functionality. It helps us build modular and maintainable applications.
- With this principle, the complexity and implementation details of a module are hidden behind a well-defined interface.

- Separation of concerns is not just about organizing code into files, but rather dividing areas of functionality. Therefore, CSS-in-JS does not violate the separation of concerns principle as all the complexity for a component remains hidden behind its interface.

- Although inline styles are easy to apply, they can make our code difficult to maintain over time and should only be used as a last resort.

- We can add icons to our applications using the react-icons library.

- There are several UI libraries available that can assist us in quickly building beautiful and modern applications. Some popular options include Bootstrap, Material UI, TailwindCSS, DaisyUI, ChakraUI, and more.

VANILLA CSS

```
import './ListGroup.css';

function ListGroup() {
  return <ul className="list-group"></ul>;
}
```

CSS MODULES

```
import styles from './ListGroup.module.css';

function ListGroup() {
  return <ul className={styles.listGroup}></ul>;
}
```

CSS-IN-JS

```
import styled from 'styled-components';

const List = styled.ul`
  list-style: none;
`;

function ListGroup() {
  return <List></List>;
}
```