

Student's Name

Instructor's Name

Course

Date

React Styling with Material UI

Material UI is based on Flexbox. Therefore, all the components are grid-based. They have built-in styles that can be extended or overwritten by other props (Stirrup). Styling a React app with Material UI makes it easy to override its theme and classes and prioritize the CSS rules of the styled components.

The first necessary step before using Material UI is installation. It can be installed through the NPM package or Yarn. The commands are shown below, respectively.

```
npm install @material-ui/core/
```

```
yarn add @material-ui/core
```

The Material UI theme feature enables one to create an attractive base for styling the application. It provides default style rules for the components, such as color and shadow. The code snippet below shows changing all the buttons' background color into green.

```

src > js theme.js > [🔍] default > 🔗 overrides > 🔗 MuiButton
1   import { createMuiTheme } from '@material-ui/core/styles';
2
3   export default createMuiTheme({
4     overrides: {
5       MuiButton: {
6         root: {
7           backgroundColor: 'green',
8           margin: '10px',
9         },
10      },
11    },
12  });
13

```

We then provide this theme to the app using the `MuiThemeProvider`, which takes in a theme as a prop and injects it into the application. This process is shown below.

```

src > js App.js > 📦 App
1   import Button from '@material-ui/core/Button';
2   import MuiThemeProvider from '@material-ui/core/styles/MuiThemeProvider';
3
4   function App() {
5     return (
6       <MuiThemeProvider>
7         <Button variant='contained' color='primary'>
8           Hello World
9         </Button>
10      </MuiThemeProvider>
11    );
12  }
13
14  export default App;
15

```

The styled method is then used to style the Material UI components. This method takes in a React component as its input and outputs another React component with a varying style

(Stirrup). This method is useful and convenient as it styles components using the same syntax with CSS stylesheet.

```
src > Js Button.js > ...
1   import Button from '@material-ui/core/Button';
2   import styled from 'styled-components';
3
4   export default styled(Button)`
5     color: white;
6     background-color: blue;
7     height: 80px;
8 `;
9
```

Material UI has a large and active community. This implies that it is easy to get help and fix an issue when it rises. Its explicit documentation exhausts its functionalities when styling a React app. Besides, the code for styling the components is minimal as it focuses on the logical structure. It also provides Sass support, making it easy for developers to understand styles with reduced code. Material UI's several advantages make it have a broad user base.

Work Cited

Stirrup, Benjamin. "Why Material UI and Styled-Components?" *Sipios*, 17 Oct. 2019,
[https://www.sipios.com/blog-tech/how-to-use-styled-components-with-material-ui-in-a-re
act-app](https://www.sipios.com/blog-tech/how-to-use-styled-components-with-material-ui-in-a-react-app). Accessed 25 Feb. 2022.