**React Material UI Notes**

**Introduction**

* MUI is a UI component library for react that provides us with components to quickly build user interfaces
* MUI is an implementation of Google’s material design specification. Material design is a design philosophy by google which bundles together best practices/principles for designs.

**Setup**

* First, we need to create a react app, getting rid of boiler plate files/code.
* Then, we need to install the MUI package. Go to <https://mui.com/material-ui/getting-started/installation/> and copy the npm installation command: ‘npm install @mui/material @emotion/react @emotion/styled’
* Notice we installed mui/material which is the core material library that provides us with ready to use components
* Notice we installed emotion which is a library designed for writing css styles with js and is the default styling solution

**Default Theme**

* Go to <https://mui.com/material-ui/customization/default-theme/>, then toggle expand all
* Here we see the default theme for MUI components
* For example, we can see the default value for grid points:

**Typography**

* Use typography to present your design and content as clearly and efficiently as possible.
* To use the Typography component, import it from @mui/material
* Text

  Description automatically generated
* Graphical user interface, text, application, chat or text message

  Description automatically generated
* The ‘variant’ property specifies the styling. For example, the above component has a ‘variant’ of ‘h1’ so it’s font weight and size is that of an ‘h1’ HTML tag. Note that body1 is the default ‘variant’ of the Typography component.
  + h1 to h6
    - Underlying HTML element of h1 to h6
  + subtitle1, subtitle2:
    - Underlying HTML element is h6
    - These are variants of the h6 tags that don’t stand out as much as the h6 variant.
    - Subtitle1 has a larger font size and lesser font weight
    - Subtitle2 has a lesser font size and larger font weight
  + body1, body2
    - Underlying HTML element is p
    - These are used for body text such as paragraphs of text that we need to display
* The ‘component’ property specifies the underlying HTML element. For example, the above component has a ‘component’ of ‘div’ so in the HTML, it says a ‘div’ element is rendered. If there was no component property, the underlying HTML element would be that of the ‘variant’ prop. So in this example, the underlying HTML element if we didn’t specify a ‘component’ property would be ‘h1’. Note that changing the ‘component’ prop does not affect the styling, even if there is no ‘variant’ prop.
* The gutterBottom prop adds a margin at the bottom. Large variants will have more margin at the bottom.

**Button**

* Text

  Description automatically generated A picture containing text

  Description automatically generated

**Button Group**

* The ButtonGroup component can be used to group related buttons.
* Text

  Description automatically generated A picture containing text

  Description automatically generated

**Toggle Button**

* Toggle buttons can be used to group related options.
* To emphasize groups of related Toggle buttons, a group should share a common container. The ToggleButtonGroup controls the selected state of its child buttons when given its own value prop
* Text

  Description automatically generated
* Text

  Description automatically generated with medium confidence
* In the above example, both ‘bold’ and ‘underlined’ are selected. To only be able to select one, add the ‘exclusive’ prop to the ToggleButtonGroup.
* Text

  Description automatically generated Graphical user interface, text, application

  Description automatically generated

**Text Field**

* Text fields let users enter and edit text.
* Text

  Description automatically generated Graphical user interface

  Description automatically generated

Box

* By default, the Box component is exactly the same as a div
* However, we can also pass in the sx prop
* Basically we use Box component whenever we need to use a div, and we can pass in the sx prop for quick css changes

Theme

* By default, every component follows the default theme (which we can override)
* For example, inside the default theme object, there is a ‘palette’ property which allows us to set our colours.
* To make a theme, we use the createTheme function which we need to import. This function takes in a object as its parameter. This object represents all the properties of the default theme that we want to override. This function returns an object which represents the new theme our app will use.
* To allow our app to use this theme, we use the ThemeProvider component. Any children compoennts will gain access to the theme provided to the ThemeProvider component. Thus, we pass our theme object (the return value for he createTheme function) and pass it as the value of the ‘theme’ prop inside the ThemeProvider component.
* Text

  Description automatically generated
* Text

  Description automatically generated
* A picture containing text, clock

  Description automatically generated
* Mui is built in with some colours

Paper

* Paper is MUI’s background component
* We want to use it whenever we need a background for something
* Text

  Description automatically generated
* 

Sx prop

* Provides inline styling
* By default, the defulalt spacing is 4 px. When we set p:2 we say the padding is 2\*4px = 8px
* <https://mui.com/system/properties/>
* Its meant for small changes

For reusable styles, use the styled function

they deprecated the makeStyles shit