# All practice is done in the 15.React Router Fast pizza project

TailWind CSS

# What is Tailwind?

* Tailwind is a utility first, CSS framework, packed with utility classes like *flex, text-center, rotate-90 + many more* that can be composed to build any design, directly into the Markup (HTML or JSX)
* **Utility first CSS**
  + Writing tiny classes with one single purpose
  + Combine them to build entire Layouts
* With Tailwind, we are not creating any new CSS, we are just using something that is prebuilt

# Pros and Cons

## Pros

* The classes are already named, so we don’t have to think about the naming
* No jumping between the files (code straight into HTML or JSX)
* Can be seen as a design system with **consistent UI decisions**
  + Spaces
  + Fonts
  + Colors
  + **It’s looking good out of the box**
* Documentation is WORLD CLASS

## Cons

* Markup (HTML or JSX) is unreadable at the beginning
* Learn the classes TailWind provides
* Install and setup tailwind on each project
* Give up on ‘vanilla CSS’

# Setting up TailWind

* We are using the **Framework Guides** so that we can set it up with **Vite**
  + <https://tailwindcss.com/docs/guides/vite>
* Install the Tailwind VSCode extension
  + This will show use exactly what CSS is written using those classes
* Install Tailwind Prettier Extension
  + <https://github.com/tailwindlabs/prettier-plugin-tailwindcss>
  + npm install -D prettier prettier-plugin-tailwindcss
  + Add this code to the prettier configuration file
    - Create a new file .prettierrc
    - Add this code
      * "plugins": ["prettier-plugin-tailwindcss"]
  + This extension will **automatically sort the classes by name**

# Colors

* https://tailwindcss.com/docs/customizing-colors
* Text-red-600
* Bg-green-100

# Styling Text

* Font Size 🡪 Text-xs … Text-sm … Text-lg …. Text-9xl
* Font Weight 🡪 font-thin …. Font-medium … font-extrabold
* Text Transform 🡪 uppercase … lowercase … normal-case

# The Box Model

* Margin 🡪 mb- … mt- … ml- … mx- … my- …
* Padding 🡪 p- … pt- … py- … px- … pt- … pb- …
* Border 🡪 border-b … border-x … border-y ….
* Spacing
  + We can add spacing without flexbox
  + We use **space-x-3** (space-y-4 …. Etc.)
* Display 🡪 block … flex … inline … grid … hidden

# Responsive Web Design

* https://tailwindcss.com/docs/responsive-design
* By default, Tailwind comes with 5 breakpoints
  + Sm
  + Md
  + Lg
  + Xl
  + 2xl
* These are based on the @media (min-width) media queries
* These are **mobile first** 
  + When working with Tailwind, we **should implement mobile design first**
* For this to work, we have to add the **base** rule and also the rules for the **responsive design**
* The property is applied **from the screen size and larger**
  + Md: hidden 🡺 from Medium screen size to Larger
  + Sm: border-b 🡺 from small screen to larger

A screenshot of a computer

Description automatically generated

* Default **display** is **hidden**
* sm:block 🡺 From **small** screen size to larger is **block**

# Flexbox

* Flex
* Align Items 🡺 Items-center … items-top … etc
* Justify Item 🡺 justify-start … justify-between … justify-center … etc

# Grid

* We want to build the layout using Grid
  + We set 3 Rows (header, main, footer) so we can push the footer all the way down
* We want to create 3 **custom** rows
  + We use the [] to define our own **Arbitrary Values**
  + className=’grid grid-rows-[auto\_1fr\_auto]”
  + and the h-screen 🡺 height to 100 view height

# Buttons

## Hover

* we use the **hover:**  prefix to add styling when we **hover** items
  + hover:text-blue-700
* Border Radius 🡪 rounded-lg … rounded-sm … rounded-full

## Transition

* Transition 🡪 transition-all … transition-colors …

## Duration

* Duration 🡪 duration-0 … duration-75 … duration-100

# Form Elements

* We can reuse styles with **@apply**
* Using **@apply**
  + We create an class in the .css file
  + @layer components{
* @layer components {
* .input {
* @apply w-full rounded-full border border-stone-200 px-4 py-2;
* }
* }
* Now we use the **@apply new class**

# Reusing Styles with React Components

* We are creating a new react component (**Button**)
* And add the styles to that react component
* We are creating it as an API

# Z and Absolute Positioning (Spinner)

* We want to add the Spinner in the middle of the page and add some BLUR
  + We wrap the spinner in a parent element
  + Position the parent on the whole page
  + Center the loader in the parent
  + Blur the parent