



TAILWIND



SHEERAZ MEHBOOB



Contents

Requirements:.....	2
Styles:	2
Custom Builds / Break Points:	2
Adding Styles:	3

TailWind CSS

- Uses an approach to build Web Apps called Utility First approach.
 - You can create a unique Web app without using CSS
 - Unlike Bootstrap Tailwind has no classes, it only has a responsive grid and flex box classes
- ➔ You can either download GitHub repository or setup the environment by creating a new folder tailwind that contains “index.html” file.

Requirements:

- NPM
- NPX (use to run modules without actually install them but install temporary copy of any package)
- Git
- Install Tailwind
 - ❖ Tailwind JIT CLI
 - Method -1:
 1. Command: `npx tailwindcss -o ./css/styles.css --jit --purge "/*/**/*.html" --watch`
 2. `--purge` (automatically find all classes in html files)
 3. JIT (Just in Time Compiler)
 4. `--watch` (lookup for changes in files)
 - Method -2:
 1. `npm init -y`
 2. `npm install -D @tailwindcss/jit tailwindcss postcss`
 3. `npx tailwindcss init -p`
 4. customize `tailwind.config.css` and `package.json` accordingly
 5. run `npm start`
 - ❖ CDN (not preferred when working on project as it is not fully customizable)
 - No Customization
 - No Plugins
 - No Tree Shaking (No automatic detection whether a part of code is removed or added)
 - No JIT

Styles:

Custom Builds / Break Points:

By default, styles are defined like break points but we can customize them or add new ones. We can do this by extend the portion of “config” file by adding additional elements to existing object.

- New:

```
theme: {
  extend: {
    screens: {
      medium: '1200px'
    }
  },
}
```

- Customize:

```
theme: {
  extend: {
  },
screens: {
  sm: '640px',
  md: '768px',
  lg: '1024px',
  xl: '1280px',
  '2xl': '1536px'
}
```

Adding Styles:

- Tailwind CSS cause crazy amount of classes but we can create our own custom classes.
- Add a new file 'styles.css' into 'src' folder. The styles.css is input file that is preprocessing for tailwind i.e. Tailwind will look into this input file to override the existing tailwind classes and output the styles based on that.

→ First of all we will incorporate following

```
@tailwind base: (Series of styles Tailwind need to reset how things looks on browser).
@tailwind components: This directive in styles.css tells Tailwind CSS to include all of the styles for its built-in components in your compiled CSS.
@tailwind utilities: Has the styles that Tailwind CSS needs to output certain utilities.
```

In order to do all this we need two commands **@layer** and **@apply**. **@layer** allows us to choose which layer to target like headline etc and to add components just simply put those in components. **@apply** is used when you want to apply a specific Tailwind CSS style.

→ Modify your package.json file as:

```
"scripts": {
  "start": "tailwindcss -i ./src/styles.css -o ./build/css/styles.css --watch"
},
```

- Stop the server using “Ctrl + C” command and restart it by “npm start”.
- Now just simply modify your input file and see the changes.

- We can even write simple css to create our own custom classes or customize the elements.

