Millersville University

Lombardo College of Business

Department of Management and Marketing

CSS Lab – Part 5 – Tailwind

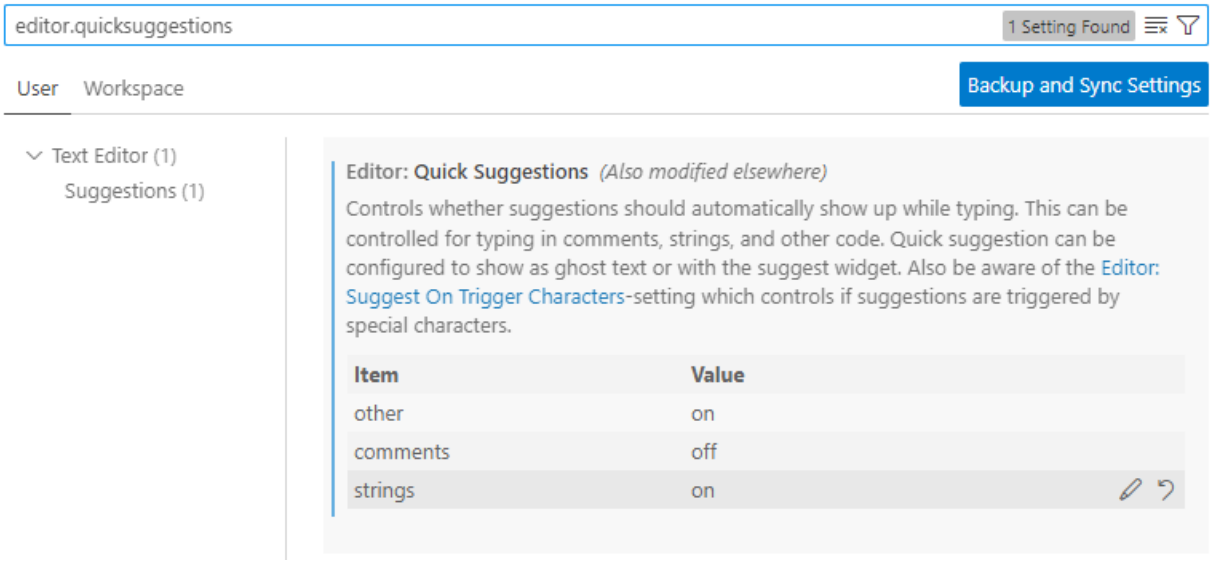
Part 1:

In this lab, we will be using the state-of-practice CSS framework named Tailwind CSS. Tailwind is a utility-first CSS framework. Bootstrap is a component framework so the most common items you need are already built. The downside is customization is rather difficult and requires recompiling the source code for the CSS. Tailwind goes the other way and rather than build premade components, it creates premade styles that you can call on any element you want simply by including the correct class name in the class argument.

To get started we want to go into Visual Studio Code and create a new HTML.

Next, we will want to install the Visual Studio Code extension to help with Tailwind. Search and install the extension Tailwind CSS IntelliSense. Once installed, there are some recommended settings you can set. At this time, we are not using Tailwind in the normal way (with Node.js) so the only one you should change now is the editor quick suggestion recommendation.

Go into settings and search for: `editor.quickSuggestions`
Change the strings to on.



Now create a new folder called Lab 5 in your CSS folder. Using the Emmet macro for !Enter and then writing Hello World in the body results in the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    Hello World
</body>
</html>
```

You can preview this code in the browser so we can see the visible change when we load Tailwind correctly.

Next add Tailwind CSS into the file. Do this by adding in the development version of Tailwind CSS into the head.

Add the following to the <head>:

```
<script src="https://cdn.tailwindcss.com"></script>
```

Change the <body> tag temporarily to the following to make sure Tailwind CSS loads:

```
<body class="text-blue-500">
```

Once it works in preview, remove the class. Some additional setup is necessary in this example. First save the current file as sauce.html. Then in the same directory create a new file called: tailwind.config.ts

This file is normally automatically created when we create a new project with Next.js, but since we are using a manual method, we need to create the file. Your Tailwind extension currently won't work correctly without this file. Remember this when you are having problem with the extension in the future. It is fine for now for it to be blank. Seriously, a blank tailwind.config.ts file is fine.

Create a new folder in your Lab 5 folder called css. Next create a new file in your css folder called sauce.css. Link this file in like any other CSS file.

```
<link href="css/sauce.css" rel="stylesheet" />
```

In this sauce.css file insert the following:

```
@tailwind base; /* Preflight will be injected here */
@tailwind components;
@tailwind utilities;
```

This is probably the strangest code we will write in this lab. It won't make much sense until latest in the semester. These lines allow for us to the power of Tailwind CSS. But for now, the base is a version of normalize or reset (reboot). Tailwind calls the their CSS reset/normalize Preflight.

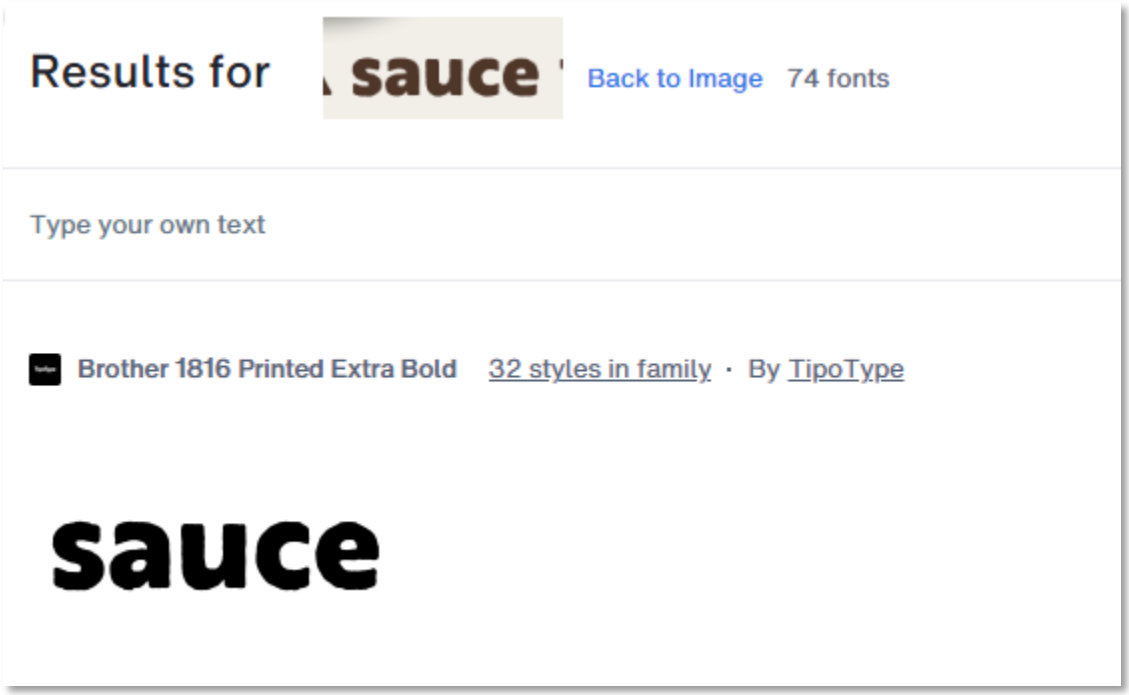
The lab will implement the following image in HTML and CSS using Tailwind:

A sauce for every craving

Stuck on which **made-from-scratch sauce** to try next? We can help. Whichever you land on is sure to pair perfectly with our **never-ending first course** that's always on us.*

DECISIONS, DECISIONS

Font: Using the image on a website like whatthefont? from https://www.myfonts.com/pages/whatthefont gives us some idea of what font is used in the image.



The website wants to locate the font for you to purchase a copy. We don't need to purchase the font, but it is important to see that Brother 18116 Printed Extra Bold is suggested as the font picked. I believe that it probably was so we will design our page with this font.

There are a few ways to get fonts for your webpage. Google Fonts is a great resource for free fonts that you can link easily. There are currently 1601 font families available. Each font family can have many styles, for example the Brother 1816 Printed is the name of the font family, but the style is called Extra Bold.

The advantage of Google Fonts is that it is free. Another option is Adobe Fonts. Adobe Fonts are usually not free but are the most popular fonts used throughout time. As part of my university membership to Adobe Creative Cloud, I can use Adobe Fonts for free as long as my subscription is active. I can pick fonts and then install them on my computer or have Adobe create a custom CSS file that I can access. Adobe is currently using TypeKit to implement this service. So, the next step is to link my personal style sheet that includes Brother 1816 Printed.

Put the following in your head:

```
<link rel="stylesheet" href="https://use.typekit.net/fsa3yhp.css" />
```

As another example, look at the following image for Olive Garden:

If you try to identify this font, one suggestion is Eldwin-Script. I have added this font into my Type Kit CSS.

Now we must create CSS to use the loaded fonts from the TypeKit. Go to the sauce.css and add this at the bottom of the page:

```
.title {
    font-family: "brother-1816-printed", sans-serif;
    font-weight: 800;
    font-style: normal;
  }

  .normaltext {
    font-family: "brother-1816-printed", sans-serif;
    font-weight: 500;
    font-style: normal;
  }

  .olivegarden {
    font-family: "eldwin-script", sans-serif;
    font-weight: 400;
    font-style: normal;
  }
```

The title class will be dark/bold as the font-weight is 800. Normal text will be lighter at 500. Finally, olivegarden will be at 400. Tailwind use a font-weight of 300 to be regular. So, the olivegarden text will be slightly bold.

Font Awesome: A popular icon library is called font awesome. The newest version requires you to create a custom kit like I did with the Adobe fonts, but version 5 allows everyone to link like a normal custom font through CSS. Add the following to the <head>

```
    <link
      href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.4/css/all.min.css"
      rel="stylesheet"
    />
```

Then try putting the following in the code body to see that it works. Once it works delete it.

```
<span class="fas fa-x fa-smile-beam"></span>
```

Font Awesome is great, but now the company is trying to make money on new fonts or icons with different weights(bold). But most likely you will find everything you need for free.

Google Font Icons:
Google Font Icons are an alternative to Font Awesome if you find you are limited to the available fonts that are free. If you just need a few icons, you can download the svg format from Google Font Icons as svg and then enter the svg code as normal content.

Try putting the following somewhere to test it and then delete it.

```
<svg xmlns="http://www.w3.org/2000/svg" height="24" viewBox="0 -960 960 960"
width="24"><path d="m612-550 141-142-28-28-113 113-57-57-28 29 85 85ZM120-160v-
80h480v80H120Zm520-280q-83 0-141.5-58.5T440-640q0-83 58.5-141.5T640-840q83 0
141.5 58.5T840-640q0 83-58.5 141.5T640-440Zm-520-40v-80h252q7 22 16 42t22
38H120Zm0 160v-80h376q23 14 49 23.5t55 13.5v43H120Z"/></svg>
```

Colors:

You can use a color picker to grab the colors from the image we want to implement. There is a brownish color and a greenish color that we need to get the color information.

There are many color pickers. You can install one in Visual Studio Code, you can use one in your operating system or application, or can use a web-based color picker. For this simple case, use https://imagecolorpicker.com/en as your color picker. Upload the Olive Garden picture (take a screen shot and upload the screen shot) and identify the brown and green color. If you look at the background you will see it isn't pure white. The brown and green colors don't look good on pure white. Grab the background color too.

The hex code or rbg(a) is fine to grab and is essentially the same. HSL is now the preferred method to represent colors, but it is not beginner friendly.

When dealing with RBG, remember that there are three lights, a red, blue, and green. When all three lights are off you get black, when all three lights are on you get white. The fourth number is alpha

channel or how transparent it is. 255 is not transparent at all, where 0 is completely transparent and will not show up.

I get the following when using the color picker:

olivegardenbackground: rgba(242,239,233,255)
olivegartdenbrown: #4e3629
olivegardengreen: #a8ad00

What can we do with this? We can the colors directly when we need in our code or CSS. We can create CSS variables with --. But for this lab, we are going to use colors the Tailwind way.

Go to the following site and play with the colors and the code that is created:
https://tailwind-colors.meidev.co/

The code is generated for JavaScript (we created a TypeScript file, so it doesn't work directly).

Since we are in a manual method or using Tailwind, we need to put the following in the <head> instead of the configuration file.

```
    <script>
        tailwind.config = {
          theme: {
            extend: {
              colors: {
                /* Here are three custom created colors for Olive Garden */
                olivegardenbackground:
                  "rgba(242,239,233,255)" /* Use can use rbg, hsl or #color code
*/,
                olivegartdenbrown: "#4e3629",
                olivegardengreen: "#a8ad00",
              },
            },
          },
        };
      </script>
```

We now have a starting file to implement the design with all the necessary resources loaded, now it is time to code the actual page.

Start with this file:

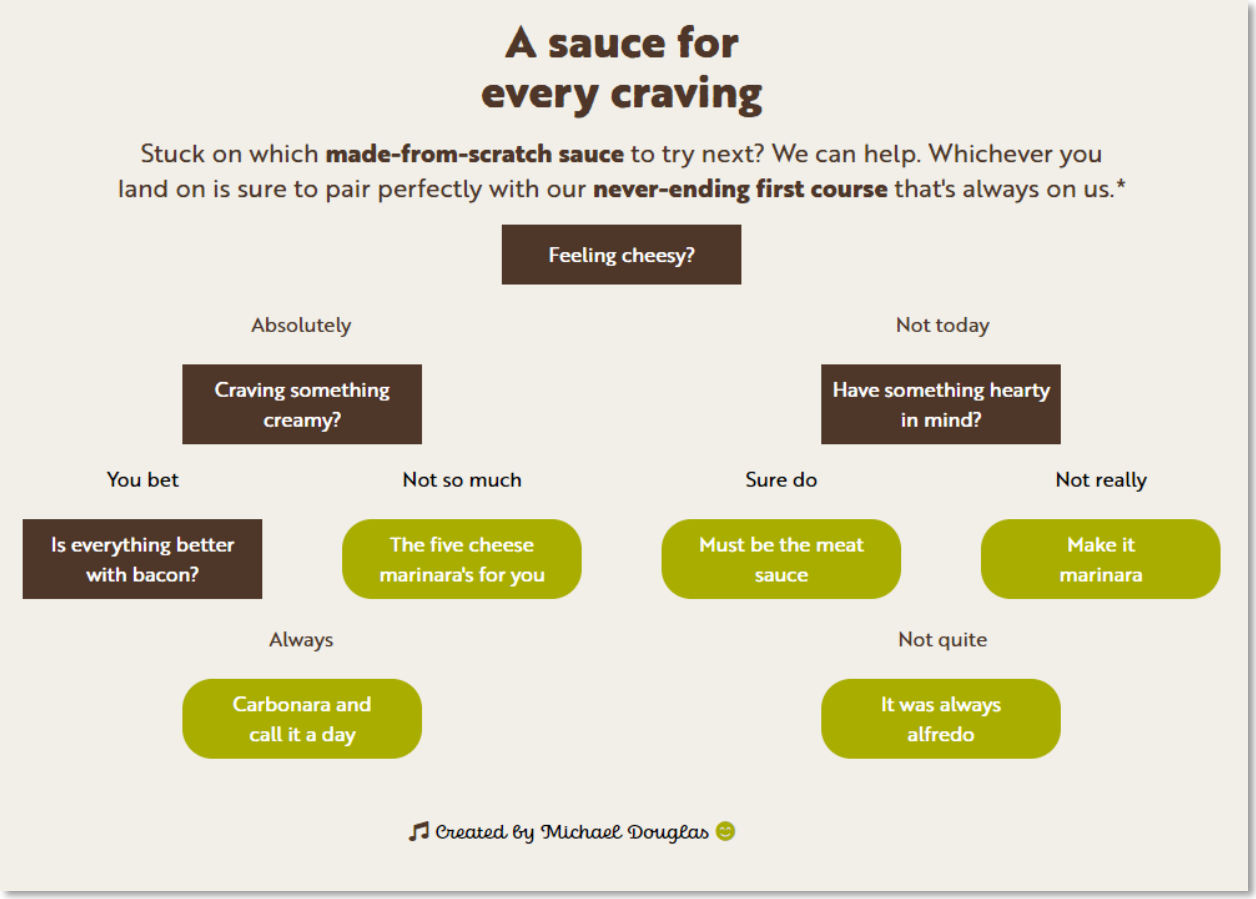
```
<!DOCTYPE html>
<!--- Load the normal html5 with emmet !enter -->
<html lang="en">
  <head>
    <meta charset="UTF-8" />
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link
      href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.4/css/all.min.css"
      rel="stylesheet"
    />
    <!-- this loads font awesome -- view pictures at
https://fontawesome.com/v5/search?o=r&m=free -->
    <title>Olive Garden</title>
    <script src="https://cdn.tailwindcss.com"></script>
    <!-- this loads the developer/preview  (non-producion, non-node.js) of
tailwind -->

    <!--- this loads the required tailwind config, normally in this goes
      into tailwind.config.ts when developing with node.js -->
    <script>
      tailwind.config = {
        theme: {
          extend: {
            colors: {
              /* Here are three custom created colors for Olive Garden */
              olivegardenbackground:
                "rgba(242,239,233,255)" /* Use can use rbg, hsl or #color code
*/,
              olivegartdenbrown: "#4e3629",
              olivegardengreen: "#a8ad00",
            },
          },
        },
      };
    </script>
    <link rel="stylesheet" href="https://use.typekit.net/fsa3yhp.css" />
    <!-- this is my own personal typesheet, has fonts used -->
    <link href="css/sauce.css" rel="stylesheet" />
    <!-- this loads the css for the webpage -->
  </head>
  <body class="bg-olivegardenbackground normaltext">
    <!-- sets the background color to offwhite, sets the correct font -->
    <main class="container mx-auto">
      <!-- All responsive code needs a container, mx-auto will auto-center on x
axis -->
      <div>
        <div
          class="title text-4xl text-olivegartdenbrown w-1/4 my-4 text-center mx-
auto"
```

```
  >
    A sauce for every craving
  </div>
  <div
    class="text-xl text-olivegartdenbrown w-10/12 my-4 text-center mx-auto"
  >
    Stuck on which <span class="title">made-from-scratch sauce</span> to
    try next? We can help. Whichever you<br />
    land on is sure to pair perfectly with our
    <span class="title">never-ending first course</span> that's always on
    us.*
  </div>
</div>

<div
  class="bg-olivegartdenbrown text-white w-48 mx-auto py-3 text-center"
>
  Feeling cheesy?
</div>
<div class="grid grid-cols-2 mt-5 gap-x-1">
  <button class="text-olivegartdenbrown">Absolutely</button>
  <button class="text-olivegartdenbrown">Not today</button>
</div>

<div class="grid grid-cols-2 justify-items-center">
  <div class="bg-olivegartdenbrown text-white w-48 mt-5 text-center p-2">
    Craving something creamy?
  </div>
  <div class="bg-olivegartdenbrown text-white w-48 mt-5 text-center p-2">
    Have something hearty in mind?
  </div>
</div>

<div class="grid grid-cols-4 my-4">
  <button>You bet</button>
  <button>Not so much</button>

  <button>Sure do</button>
  <button>Not really</button>
</div>

<div class="grid grid-cols-4 justify-items-center my-5">
  <div class="bg-olivegartdenbrown text-white w-48 text-center p-2">
    Is everything better with bacon?
  </div>
```

```
        <div
          class="bg-olivegardengreen text-white w-48 text-center rounded-3xl p-2"
        >
          The five cheese marinara's for you
        </div>
        <div
          class="bg-olivegardengreen text-white w-48 text-center rounded-3xl p-2"
        >
          Must be the meat sauce
        </div>
        <div
          class="bg-olivegardengreen text-white w-48 text-center rounded-3xl p-2"
        >
          Make it <br />marinara
        </div>
      </div>

      <div class="grid grid-cols-2 mt-5 gap-x-1">
        <button class="text-olivegartdenbrown">Always</button>
        <button class="text-olivegartdenbrown">Not quite</button>
      </div>

      <div class="grid grid-cols-2 justify-items-center my-5">
        <div
          class="bg-olivegardengreen text-white w-48 text-center rounded-3xl p-2"
        >
          Carbonara and<br />call it a day
        </div>
        <div
          class="bg-olivegardengreen text-white w-48 text-center rounded-3xl p-2"
        >
          It was always<br />alfredo
        </div>
      </div>
      <p class="olivegarden mx-auto w-1/3 mt-12">
        <span class="fas fa-music fa-x text-olivegartdenbrown"></span> Created
        by Michael Douglas
        <span class="fas fa-x fa-smile-beam text-olivegardengreen"></span>
      </p>
    </main>
  </body>
</html>
```

If everything worked, we should have the following:



Part 2 - Assignment:

For this assignment you are to create two more files. Make sure you save all your files. Make a copy of sauce.html and call it sauceflex.html. Make another copy of the sauce.html and call it saucegrid.html. The sauce.html file has specific issues in it to allow for you to improve the file in 2 ways. I want you to understand all the lines of code in this assignment so far to do the next steps.

Flexbox

For the sauceflex.html you are to implement the design by using only Flexbox. Remove all the grid and grid* items in your classes. You need to make the design look as close as possible to design that was created at a medium screen. Learning how to use CSS Flexbox is an important skill to learn in CSS. When trying to make it look as close as possible, you need to make three different changes. First you need to change the two icons to something different from the Font Awesome collection. Second you need to change my name to your name in the Created by code. Make sure you use the Tailwind CSS classes for Flexbox and not the raw Flexbox CSS. Third, you need to work through the CSS tutorial for Flexbox called Flexbox Froggy at https://flexboxfroggy.com/. At the bottom of your webpage have an image that is linked to your image folder. This image should be a screen shot of the screen once you "win" the flexbox froggy tutorial.

Grid

Once you won the Flexbox Froggy game, you can probably guess what is next: Grid Garden. Use the saucegrid.html and this time remove all but one grid classes. Your main class should setup the entire grid for the project. Also do not use any Flexbox. Use the grid commands to create the layout as in sauce.html. Note in sauce.html I purposely didn't use grids correctly as I used grids where I should have used flexbox, and I created many smaller grids versus 1 main grid. Make sure you change my name to your name in Created by and pick 2 new icons that haven't been used in either sauce.html or sauceflex.html. Win the Grid Garden and take a screenshot of your carrots at the end on the win screen and create an <img> tag that has this screenshot linked to the picture in the image folder.

For submission:
Your root folder should be called CSS Lab 5 – Tailwind. You will have to rename your folder from Lab 5.

In this folder should be two folders:
css
images

Also included in the root folder should be:
sauce.html
sauceflex.html
saucegrid.html
tailwind.config.ts

In your css folder you should have:
sauce.css

In your images folder you should have:

screenshot-win-flex.{jpg|png}
screenshot-win-grid.{jpg|png}

Zip your CSS Lab 5 – Tailwind folder and submit to D2L. This file will be run through the auto-tester (Playwright) later in the semester so do a good job. Do not submit these files to GitHub. You don't have the license for the TypeKit. If you want to post it on GitHub, remove the line:

```
<link rel="stylesheet" href="https://use.typekit.net/fsa3yhp.css" />
```

Daisy UI:
The last thing we need to have the best modern CSS is to use a component framework that is a plug in for Tailwind. We can get the benefits of a framework like Bootstrap with the customization of Tailwind. I don't recommend installing Daisy UI in the manual mode we are using because the CSS files can get very large. Once we move into Node.js, the compiler will get rid of all unused CSS and make the ending CSS file as small as possible. Check out https://daisyui.com/ . We will come back to this later in the semester.