# SSC Website Redesign

## Introduction

<screenshot of original SSC index page>

The purpose of this project is to clean up and modernize the look of the existing SSC website (<http://www.ssc-inc.com/index.php>). I also wanted to improve navigation so that the entire site would be accessible from the index page.

<screenshot of index.html>

New and improved index.htm

## Design and Implementation

<screenshot of original design>

My wireframe for index.html

<screenshot of index.html>

New and improved index.html

(upload index.html and styles.scss)

Predictably, my design changed during the transition from wireframe to code. The main structural changes were as follows.

* I added “About SSC”, “Careers”, and “Contact Us” links to the top of the jumbotron. Thematically, all of these links related to the company as a whole, and that’s the subject of the jumbotron.
* After spending some time playing with carousel features, I finally deleted the thing because I learned it was considered bad design to use carousels for announcements. Oh well.
* I removed the button for “How Do I Dispose Of…” I thought about replacing some of the side links I was working on the navbar, I started wondering about the use of buttons vs links and after a little research realized that it would be a better convention to simply link to new resources and reserve buttons for toggling the interface to do things like trigger popup menus, video players, etc.
* Finally, at the bottom of the page, I reserved the footer for legalese and reduced the amount of it.

I wanted to incorporate fancy features, but I finally decided to keep it really simple because SSC is a company where they’re obviously having problems maintaining their web site, and it’s clear they don’t have the bandwidth to maintain anything complicated.

Dealing with the chicken-and-egg problem: Should I start with the index.html page, code in vanilla html/Bootstrap, and then create the scss file? Or should I make the css file first? (At the time, I didn’t realize it was better to just go straight to scss.) I went back and forth between index.html and got myself thoroughly messed up.

Structuring the SCSS file: \_variables.scss and the other Bootstrap scss files to

Debugging:

Aesthetics:

Managing scope creep: figuring out where to pull the plug because I really needed to get back to JavaScript and general programming concepts.

something like a Udemy Bootstrap course because there is just too much I don’t know.

The index page includes the following features.

|  |  |
| --- | --- |
| Feature | Description |
| Meta tags | I tried to write the meta tags to support SEO.  <title>Sanitary Services Company, your waste management company in Whatcom County, Washington</title>  Waste and Recycling Collection in Bellingham, Blaine, and Ferndale  <meta name="description" content="Sanitary Services Company, your waste management company in Whatcom County, Washington, provides garbage collection, recycling, and sanitary services for both residential and commercial customers." />  Sanitary Service Company: Waste and Recycling Collection in Bellingham, Blaine, and Ferndale |
| Logo | I dug up a plain SSC logo on Google because the logo on the existing SSC website had company name text I didn’t want to use in my design. I couldn’t fix the antialiasing and didn’t want to burn a lot of time installing and learning drawing tools right now, so I just worked with what I had. |
| Jumbotron | I kept the jumbotron white to work around the logo antialiasing problem. |
| Carousel | Human interest, why we’re a great company. Decided to move this under the jumbotron to make the page more interesting, it’s an attention-grabber for major announcements like weather closures. |
| Navbar | Greatest single difficulty in coding the index page was figuring out how to control text color for the links. I eventually learned that this problem is a infamous Bootstrap hazing ritual. |
| Search button | Shortcut for users who know exactly what they want. |
| Buttons for Recycling Guide and Drop-off Locations | Wanted to make these more prominent than links. Whenever I’ve gone to the actual SSC web site, these are the things I’ve been looking for, things a return visitor would want to find immediately. |
| Accordion cards for Residential, Commercial, Education, and News & Events | Major function areas are accessible from accordion cards (less cumbersome than menu dropdowns, especially for mobile users).  Happened to see a web site that agreed with me on bad usability for menu dropdowns: <https://www.nngroup.com/articles/does-user-annoyance-matter/>. |
|  |  |
| Footer | Legalese is at the bottom of the page. |

For styles.scss,

Decided to keep most of the default whites and grays in \_variables.scss, seemed like it would be too much of a PITA to change them.

It took me a lot of thrashing to figure out how to override the default Bootstrap colors and other formatting.

Better to overload a stylesheet with variables I may use in the future, or not (slows downloads)?

, and among other things I wasted too much time doing CSS formatting in the styles.css file. What I should have done earlier was work straight in styles.scss and copy/paste default formatting over from \_variables.scss and other Bootstrap .scss files and then figure out exactly what I wanted to override from the standard features.

Basic things I’ve learned: how to insert fonts, use a color palette generator, how to set navbar text colors, how to extract portions of \_variables.scss and override the colors using variables, how to use Chrome dev tool for CSS debugging, how to incorporate some basic Bootstrap features.

Wish I could have done something with ARIA.

Don’t know how good page load time is?

What threw me off from doing this was the (IMHO, counterproductive) way that Nucamp teaches CSS and Bootstrap. In the nucampsite styles.css, we were creating these random styles (kind of doing the stylesheet equivalent of inline formatting). In the real world, people using Bootstrap probably don’t do this anymore. (Anybody with half a brain or who isn’t working on legacy code, that is.)

You make a skeleton web page, just accept the default Bootstrap formatting.

rather than styles.scss/overriding \_variables.scss. The reason I did that was because I thought that if I put all my formatting into styles.scss, I would have to rebuild the entire project every time I wanted to see my changes in the browser. I didn’t now all I had to do was one simple command (npm run scss). As usual, this was probably something everyone else in the class probably clued into instantly, but my fear of

* Use styles to easily format your Word documents in no time. For example, this text uses the List Bullet style.
* On the Home tab of the ribbon, check out Styles to apply the formatting you want with just a tap.

|  |  |  |
| --- | --- | --- |
|  | Column Heading | Column Heading |
| Row Heading | Text | 123.45 |
| Row Heading | Text | 123.45 |
|  |  |  |