Website Resources

* LibGuides CMS is built on Bootstrap (<http://getbootstrap.com/>). The Website has a lot of information about how to manipulate the code. Use Bootply ([http://www.bootply.com/new#](http://www.bootply.com/new)) to text code in real time. You can test html, css, and javascript in Bootply. Elements of the catalog box was created using Bootply.
* For colors, I like to use color-hex (<http://www.color-hex.com/>). It’s a great resource in that not only does it give you the colors in hexagon code but it also gives you shades and tints of the color on the same page. You’re also given triadic colors, monochromatic, analogus, and the complimentary color. I highly recommend this site!!
* All of the icons on our page (toolbar home, ask a librarian, directory, citations, etc.) are from Font Awesome (<http://fontawesome.io/icons/>). Font Awesome has a TON of icons to choose from and it’s very simple to use. You can also choose to increase the size, color, orientation and etc. of the icon. It’s just a matter of tweaking the icon code (ex. <i class="fa fa-home" aria-hidden="true">). No need to download an image file.
* Since the CMS is through LibGuides, the best resources to use are through Springshare. The resources (<http://support.springshare.com/friendly.php?s=libguides/cmswebsite/resources>) page is full of great information. Such as, other color websites and the actual code for the header, footer, and some general CSS tweaks.
* When in need of ideas for design, I did some Google searches of different University Libraries. Certain parts of our website come from the VCU libraries page (<https://www.library.vcu.edu/>) and Furman University (<http://libguides.furman.edu/library/home>). Since Furman uses LibGuides CMS, it’s a good website to use as a template or look at their code. Springshare’s examples of Academic Library websites is good to look at (<http://buzz.springshare.com/producthighlights/libguidescms-as-website/academicexamples>). This way you can see what all can be done with the CMS.
* Creating the custom header with navigation bar was just a matter of following the directions on the Springshare page (<http://support.springshare.com/libguides/cmswebsite/advanced/navbar>). The navbar just takes advantage of the Bootstrap and allows us to add the toolbar we want.
* For any of the fancy html/CSS changes, the videos found on this site (<http://support.springshare.com/libguides/training/admin#s-lg-box-1676134>) are extremely helpful. The Springshare personnel do a great job at explaining how to use the CMS and bend or break the code so that the website looks the way we want it to.
* For the media queries, I did a number of things. One, I took a class explaining how media queries worked and completed various examples. Both Code School and Codecademy are great for learning code in a hands on environment. I highly recommend both of them for understanding front-end web development.
* Second, I read various articles explaining it (<https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries>), (<http://bradfrost.com/blog/post/7-habits-of-highly-effective-media-queries/>), (<https://responsivedesign.is/strategy/page-layout/defining-breakpoints>). After reading these articles, I was curious about which library websites were responsive and how they rendered their website into a mobile and tablet form. So, I began searching various libraries (much like I did at the design level) and decided which breakdowns I liked best and worked on applying those changes to our site.
* When I was working on the media queries for our site, I tested the site not only by adjusting the size of my browser screen, but also by testing it in one of these sites (<http://responsivetest.net/#u=http://www.zootemplate.com|1024|768|1>, <https://www.responsinator.com/>, <https://www.browserstack.com/responsive>). These are great in seeing how the site renders on different operating systems(Apple, Samsung, etc) and devices (iPhone, iPad, phones, tablets, laptops, and widescreen desktops).
* Hint: When sizing things, use flexible widths (ex. 10%) NOT fixed widths (ex. 10px). This allows the content to adjust depending on the size of the screen. This makes it a lot easier on the responsive side.
* Hint: Start with mobile design! A huge mistake I made in making our site was that I started with the desktop site instead of working from the smallest screen size and working up. This caused HUGE problems when it came time to do media queries. Once I went back into the code and resized the boxes to percentages, it helped a lot. But, the next time when you begin to CSS the site and make certain changes (like sizes and placement) start with the mobile site. This will make things easier in the long run.