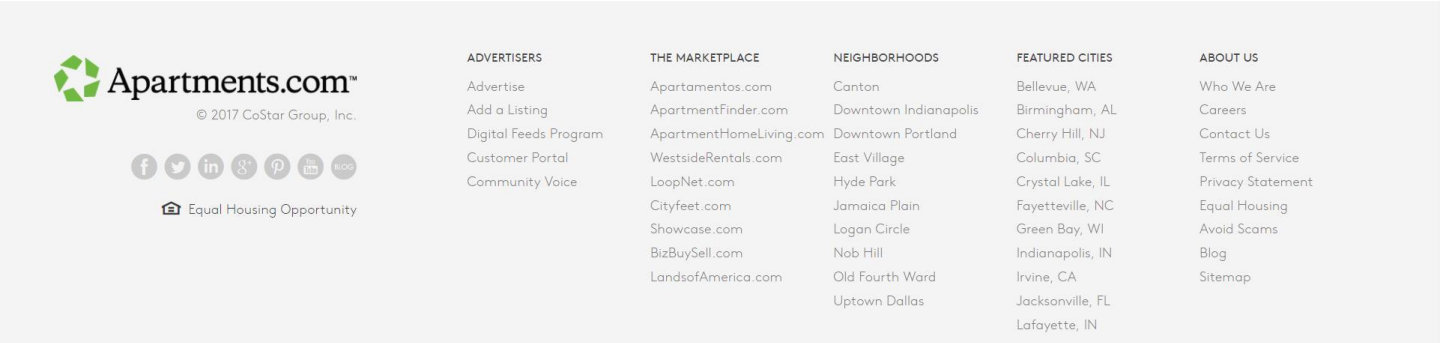
PATRICIA SCHNEPP

Front-End UI Developer

PROFESSIONAL WORK EXAMPLES

Apartments.com Homepage Main Footer Links (Tablet through Desktop)
http://www.apartments.com



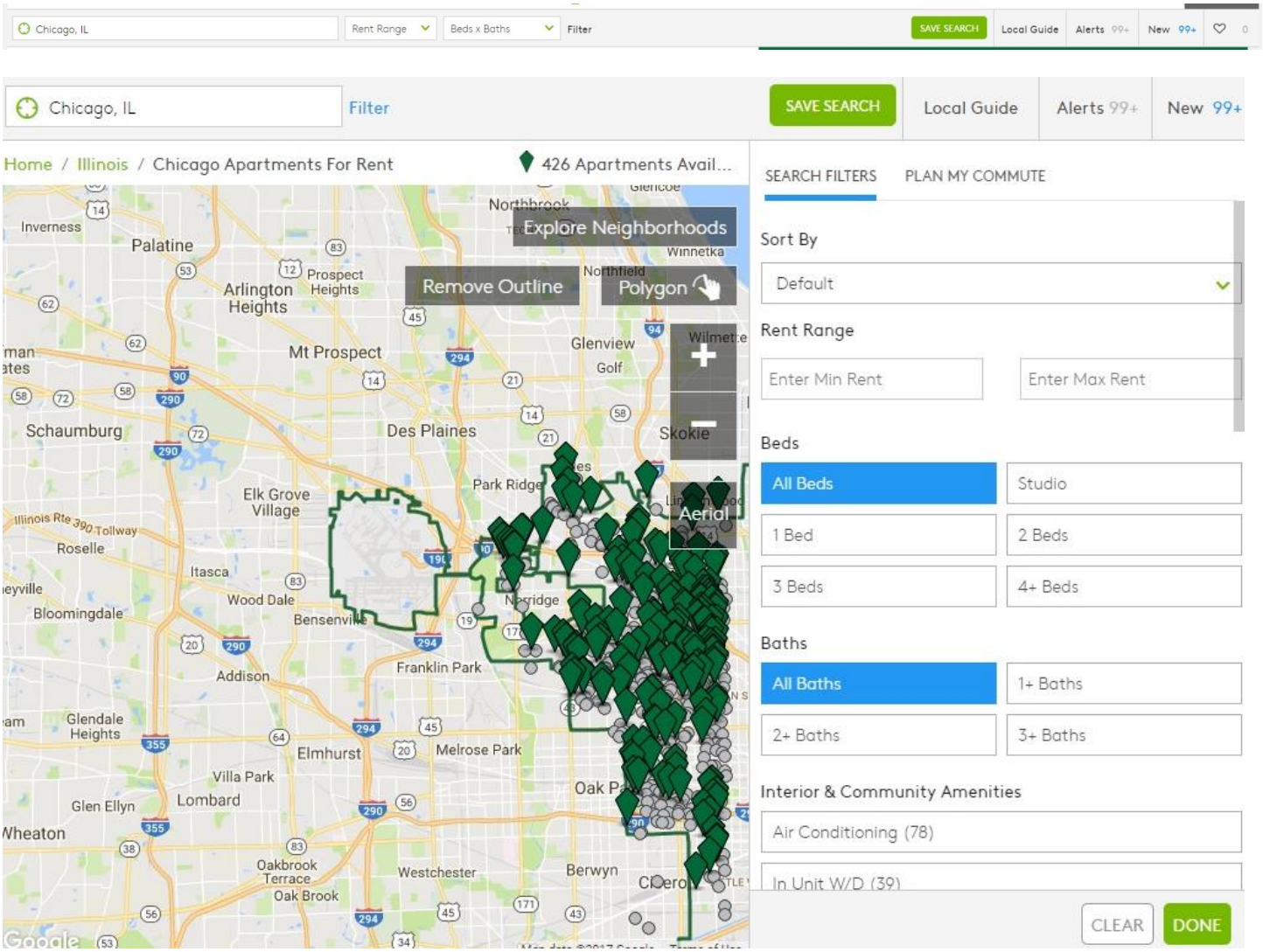
The footer section above was always a problematic area for the website especially when a user views it at smaller resolutions. The text in the columns, if too long, would edge up against the text in the next column. Last March, the new CSS Grid spec launched and this was something I had learned about at the CSS Dev Conference I attended in October 2016. This was the perfect opportunity to use it in a production environment. This was the perfect candidate as this layout always has one row and a fixed number of columns where I could utilize grid properties for Chrome, Firefox, Safari and use the IE prefixes for IE 10, IE 11, and Edge. Using the grid specs, I created a single row with five columns that would equally adjust the width based on the available space. When using the IE grid, you must define the number of columns and also define which row and what column number the column is to be placed.

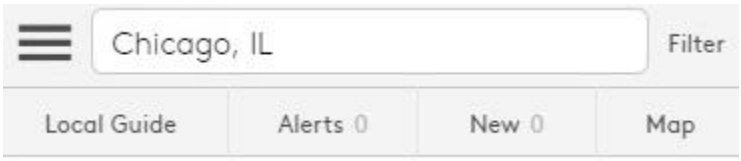
CSS Snippet:

```
.footerLinksWrapper {
    display: -ms-grid;
    display: grid;
    -ms-grid-columns: 1fr 1fr 1fr 1fr 1fr;
    grid-template-columns: repeat(5, 1fr);
}
```

```
.footerLinks {
        display: inline-block;
        margin-right: 5px;

        /* Items below are needed for IE Grid */
        -ms-grid-row: 1;
       &:nth-of-type(1) {
            -ms-grid-column: 1;

       }
       &:nth-of-type(2) {
            -ms-grid-column: 2;
       }
       &:nth-of-type(3) {
            -ms-grid-column: 3;
       }
       &:nth-of-type(4) {
            -ms-grid-column: 4;
       }
       &:nth-of-type(5) {
            -ms-grid-column: 5;
            margin: 0;
       }
```

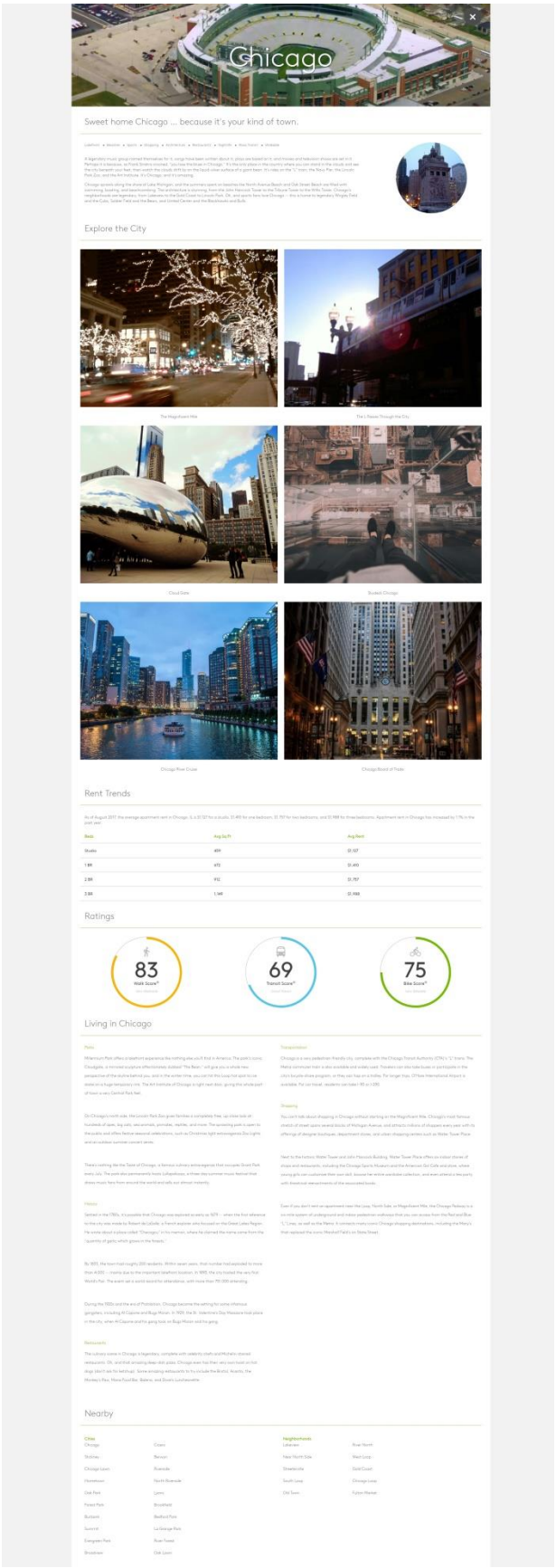
Apartments.com Search Bar Enhancements (Mobile, Tablet, and Desktop)
https://www.apartments.com/chicago-il/ | https://www.apartments.com/es/chicago-il/





Search bar enhancements are always a challenge because as a user gets down to smaller resolutions the real estate to add new components becomes very small. When adding the Local/Campus Guide Button, Alerts Button, New Button, and Favorite Button, I had to work closely with our designer to get everything to fit especially when translated to Spanish. As this search bar is a highly visible piece of functionality to our users we had to ensure that it worked across all resolutions including any breakpoints that aren't our main breakpoints. In the original design, the rent range and bed bath dropdowns were included in the mock ups until the 1024 resolution, but when the HTML/CSS was coded there was not enough real estate for all the components. I had suggested that we drop rent range, beds, baths, and favorites button at 1024. While testing all resolutions, I also came to the conclusion we had to drop the items from the search bar even sooner at an odd breakpoint because the design was breaking. For the mobile version, I used Flexbox to ensure the flexibility in the widths to adjust based on available space.

Apartments.com Local/Campus Guide Redesign (Mobile, Tablet, and Desktop)
https://www.apartments.com/chicago-il/#guide



For this project, our team had to convert our stand alone city/neighborhood pages into a full screen modal for SEO purposes. With the conversion of these pages came a complete redesign for the full screen modal for all resolutions. One of the challenges for this project was ensuring that all images kept their aspect ratio while fitting into their available space.

For the hero image on desktop and tablet we had to keep the 4.2 aspect ratio. The width was set to 100% then to calculate the height we took 100vw and divided that by 4.2 and for anything above 1920 we took the same calculation and subtracted five percent to reduce the extra white spacing for anything below 1920.

CSS:

```
    @aspectRatio1920: 4.2666;

    &.heroImage {
        width: 100%;
        height: calc(100vw / @aspectRatio1920);
        overflow: hidden;

        @media only screen and (min-width: 1921px) {
            height: calc((100vw / @aspectRatio1920) - 5%);
        }
     }
```

For the Explore the City section used a different approach that we use on the property placard photos and the photos on the property profile page. First we take the div that contains the background image and image and set a before property with content set to blank and the padding top to 80% to create the square image. Then I create an aspect ratio div that is absolutely positioned and the div that contains the image and background image is set to 100% with background properties are set.

HTML/CSS:

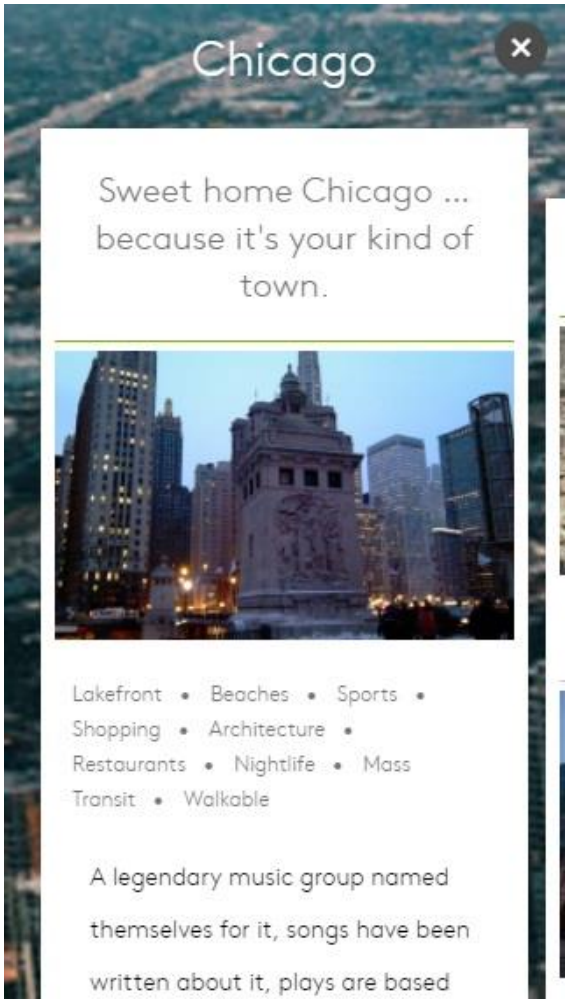
```
<div class="exploreInfoImageWrapper">
        <div class="exploreInfoAspectRatio">
               <!-- ko if: $parent.isLocalGuideVisible() -->
              <div class="exploreInfo" data-bind="style: {backgroundImage: BackgroundImage}">
               <img data-bind=" attr: {src: Uri, alt: Title, title: Title}" />
              </div>
              <!-- /ko -->
         </div>
</div>
.exploreInfoImageWrapper {
        position: relative;

        &::before {
        padding-top: 80%;
              content: ' ';
              display: block;
       }
        .exploreInfoAspectRatio {
               position: absolute;
              top: 0;
              right: 0;
              left: 0;
              bottom: 0;
        }
         .exploreInfo {
                background-position: center center;
              background-size: cover;
              background-repeat: no-repeat;
              height: 100%;
              img {
               display: none;
              }
         }
}
```

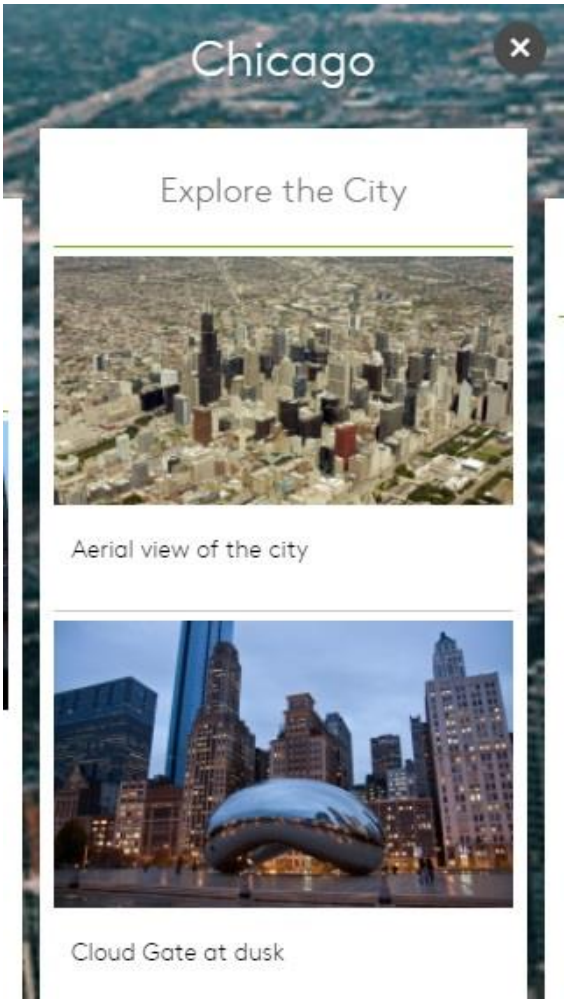
Another section worth pointing out for the Desktop/Tablet version of the Local Guide modal is the Ratings section. Again, here I was able to utilize the new CSS Grid property. The way this grid was implemented is a bit different than the homepage footer links grid is because not all Local Guides will have all three types of ratings. In this situation, I was not able to utilize the IE prefixes as we won't always know how many columns will be present. The other way to utilize the new CSS grid property is to use the @supports query. This @supports query will only run the CSS styling only if the browser supports that CSS property and is added to override after the fallback CSS definition.

```
.ratingsInfo {
     .displayFlex(); //fallback for Grid if no support
     text-align: center;
}

.ratingsCol {
     .flex(0, 0, 33.33%); //fallback for Grid if no
support
}
```

```
/* Use CSS Grid if Supported */
@supports (display:grid) {
     .localGuideContentWrapper {
          .ratings {
               .ratingsInfo {
                    display: grid;
                    grid-gap: 10px;
                    grid-template-columns: repeat(3, 1fr);
               }

               .ratingsCol {
                    flex: none;
               }
          }
     }
}
```

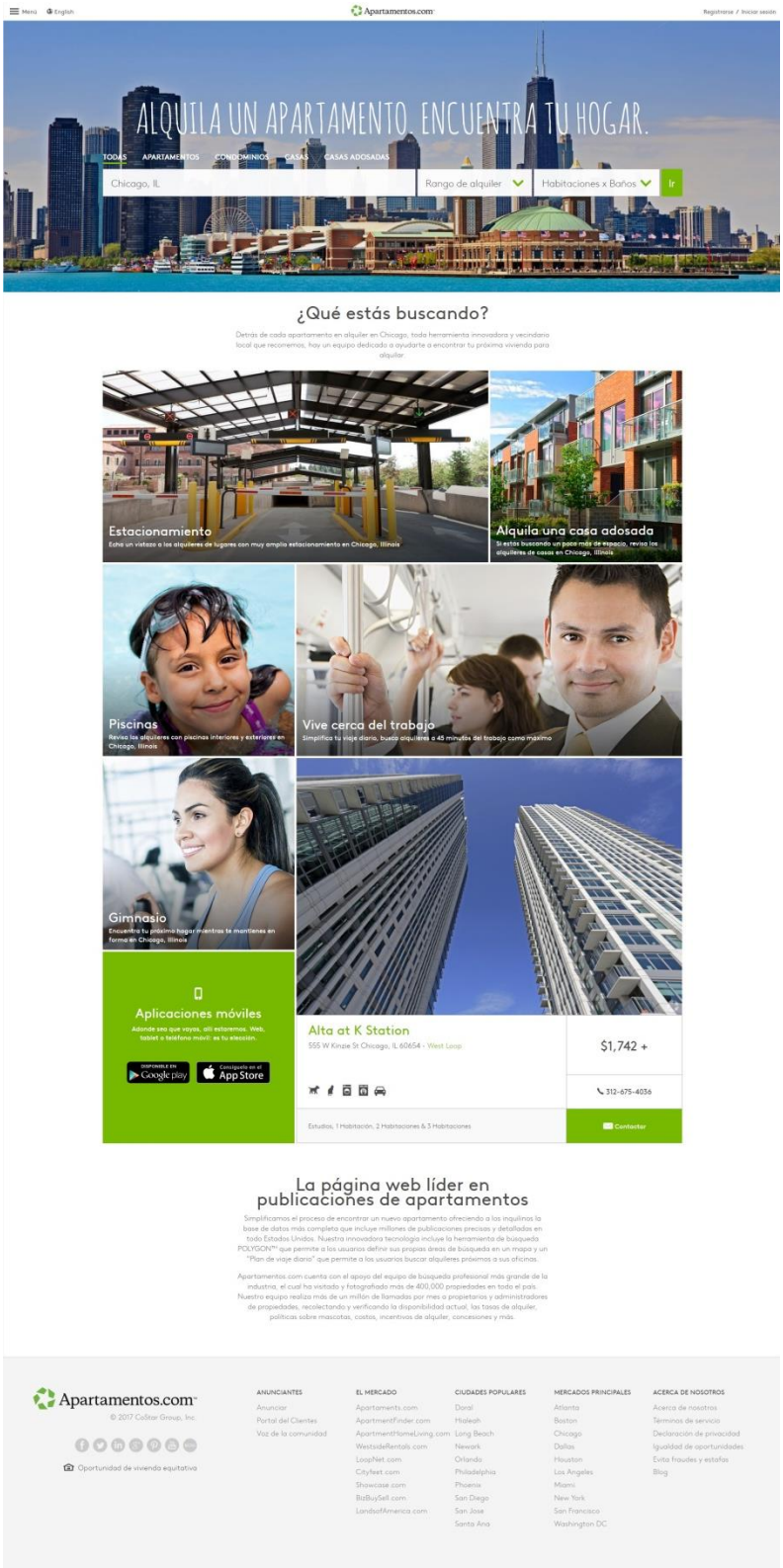
Lastly, the mobile version of the Local Guide was a fun challenge as it was setup up to look like cards that you swipe to the left and right for each section of the Local Guide.





This was a fun challenge as there the City/Neighborhood header title was fixed and also each of the cards were somewhat fixed and the information within the cards had to be scrollable. The main goal for me when developing was to ensure that the cards were scrollable left and right by setting them to inline-block even before the Javascript wire up. The cards content overflow was set to scrollable to allow users to scroll within the card to see all the information. Once I achieved this, I was able to pass it off to the Javascript developer on the team to implement the Slick Javascript plugin to help with positioning and behavior of the swiping the cards left and right.

Apartamentos.com (Mobile, Tablet, and Desktop)
http://www.apartamentos.com



This project we translated all of Apartments.com into Spanish. The challenge here from a design standpoint was making sure all translations would fit into the current design. If translations didn't fit I would either get a shorter translation from the translator or if that still didn't work I would work with our designer to try and come up with an alternative design that didn't sway too far from the original design. Usually in the case where there wasn't a shorter translation, we would play around with font sizes and spacing to get it to fit in the available space. In these instances, I created a separate Spanish LESS file to only override the styling needed for the Spanish site.