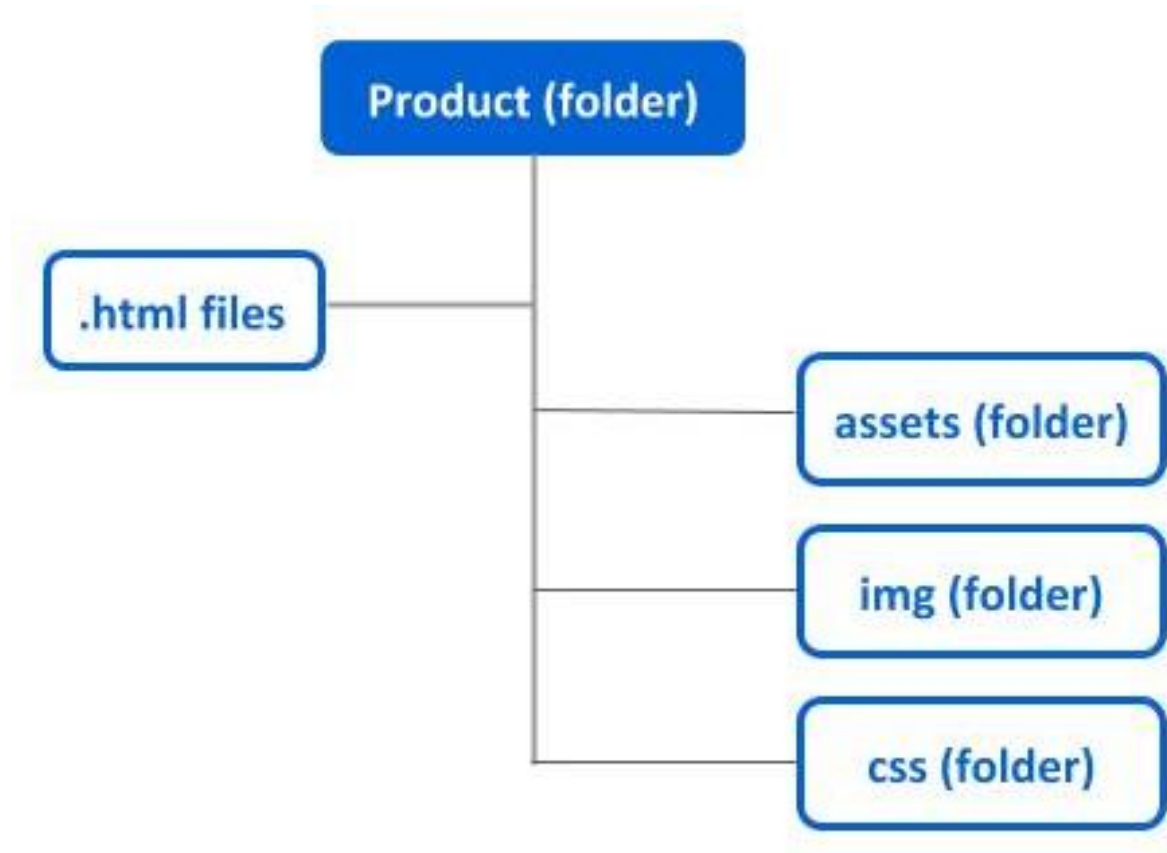


## Criterion E: Product development



My folder structure allows me to easily sort my files by their functionality, so html files are in a .html file folder, images are in an images folder, .css files are in a css folder and my documents that are being contained on the website are being held in an assets folder. The project will be hosted online using Repl.it, but will require purchasing a custom domain name, which my client has agreed to.

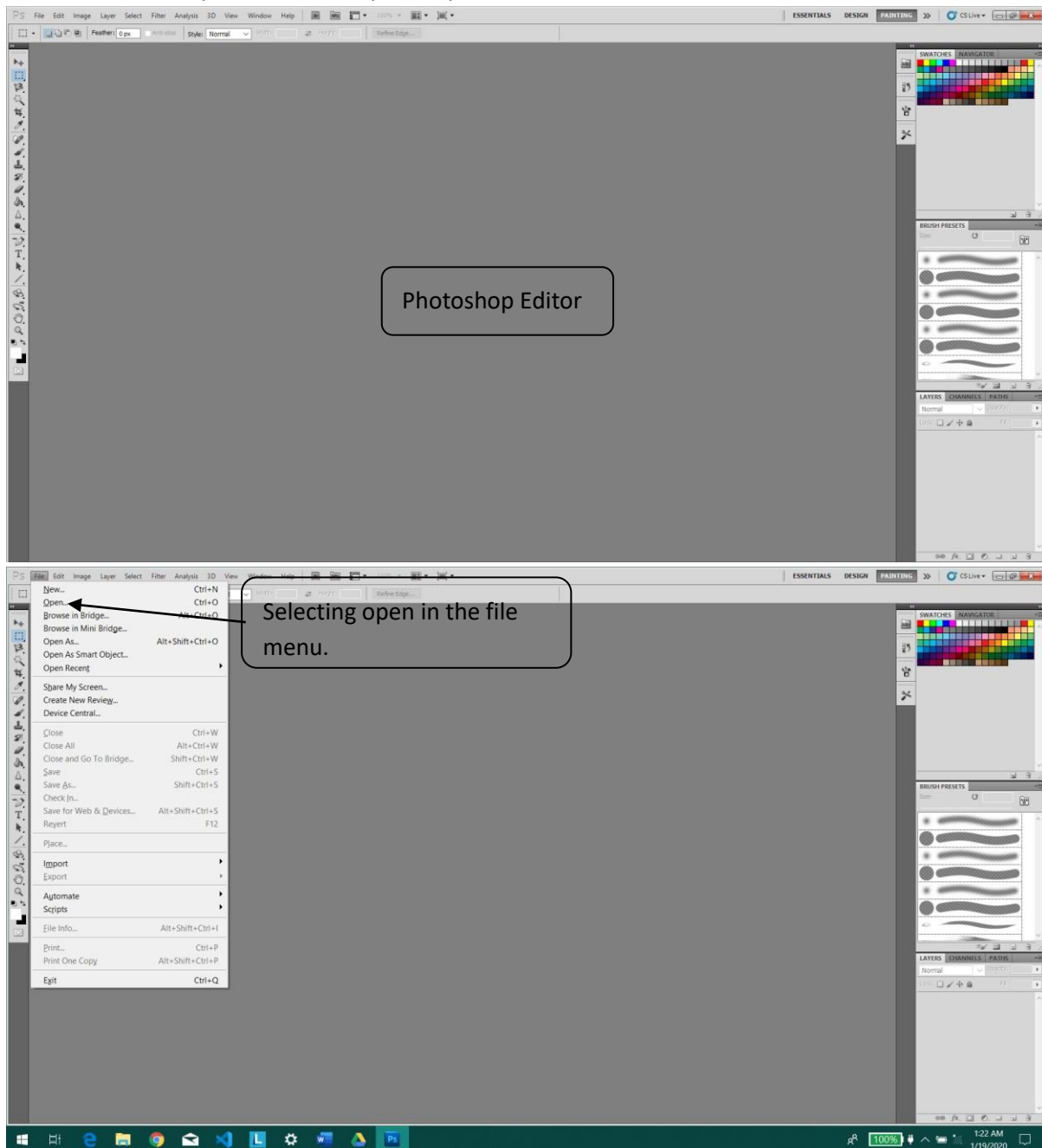
### List of complex techniques used to address the client's requirements:

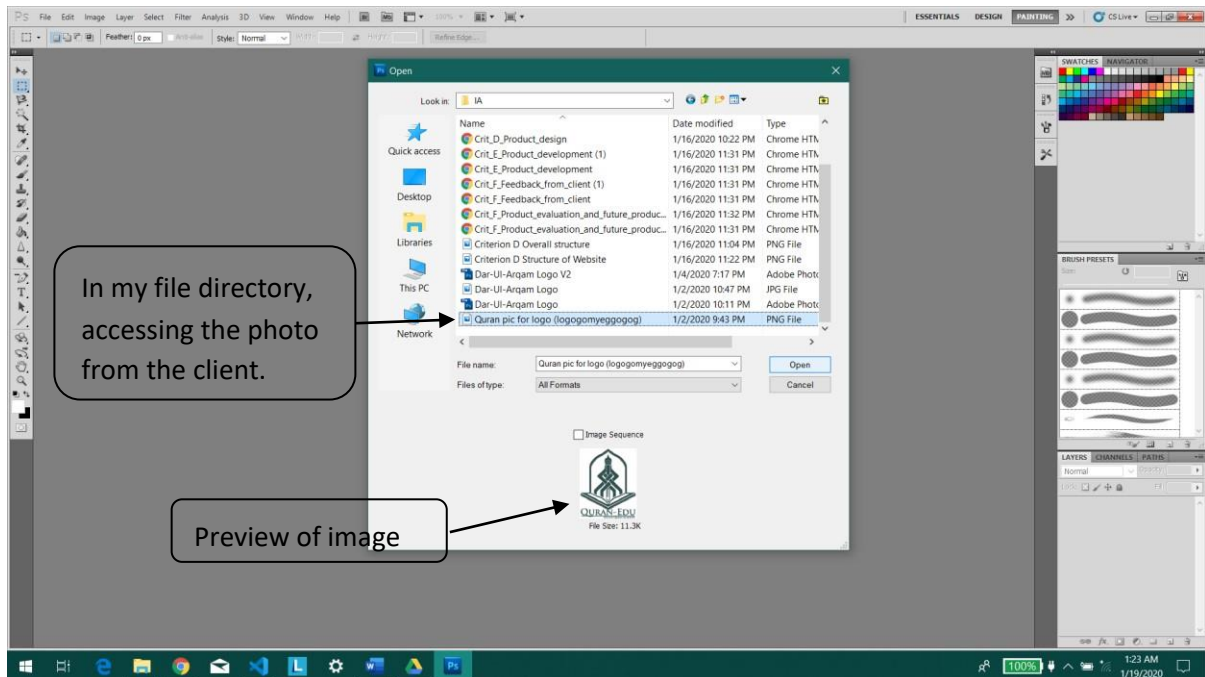
1. **Manipulation of graphics**
  - Creating a logo
2. **Cascading style sheets**
  - Adding hover effect to links
  - Styling the contact form
3. **Proficient integration of the different elements of multimedia**
  - Embedding a Google Calendar iframe
  - Embedding a Google Maps iframe
4. **Manipulation of codes such as HTML, XML, Java, JavaScript or Visual Basic to customize pages or improve functionality**
  - Creating a to-top button using HTML and JavaScript

### Manipulation of graphics

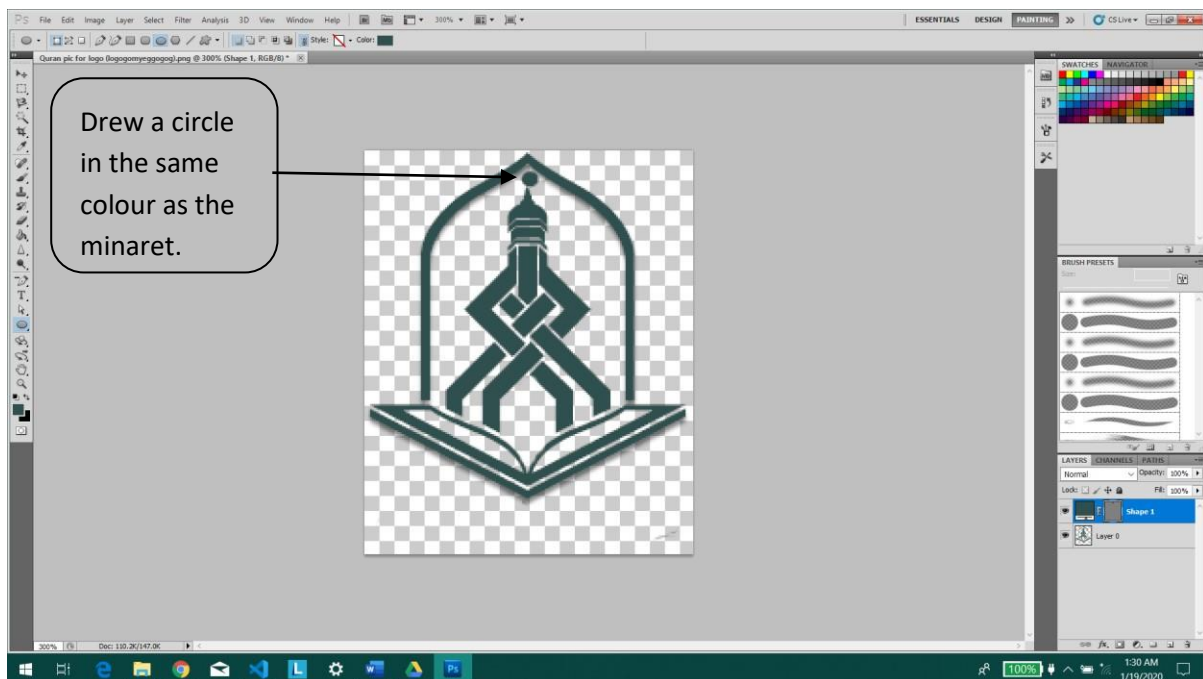
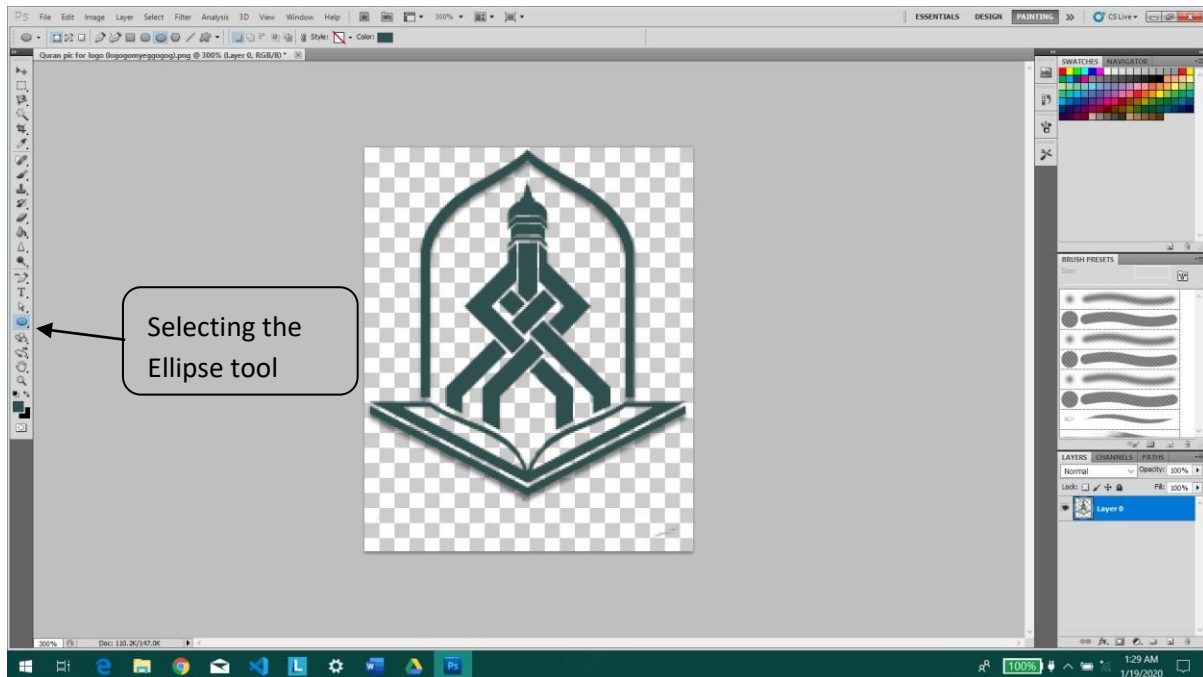
### Creating a logo

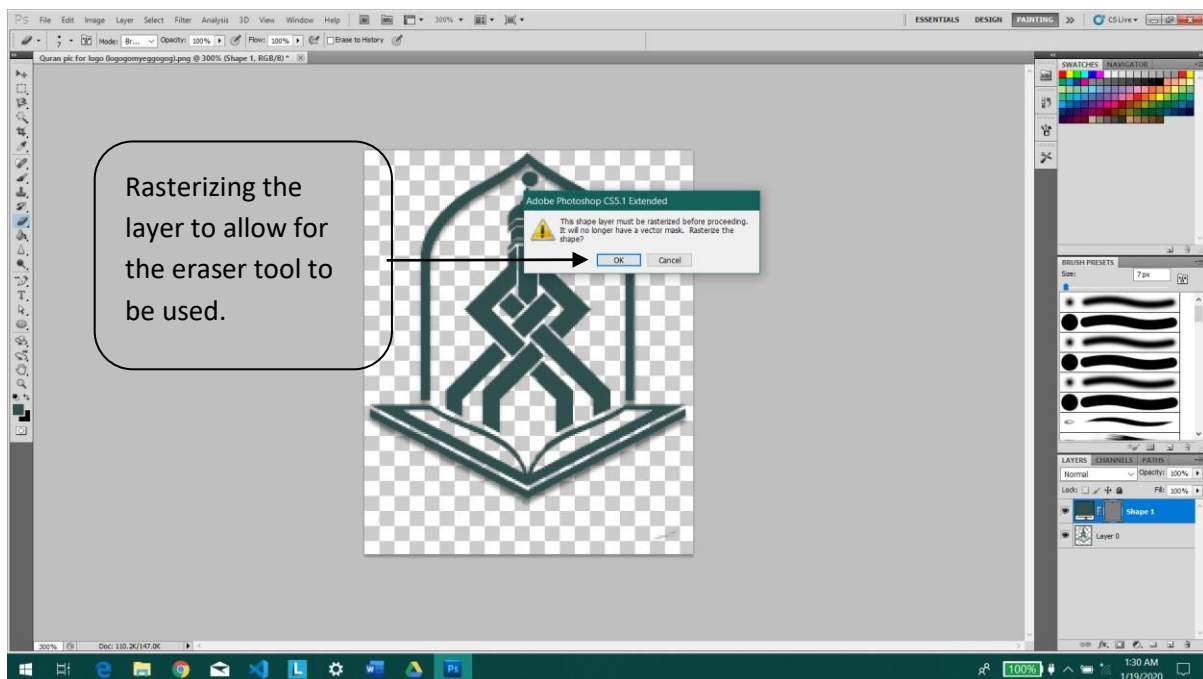
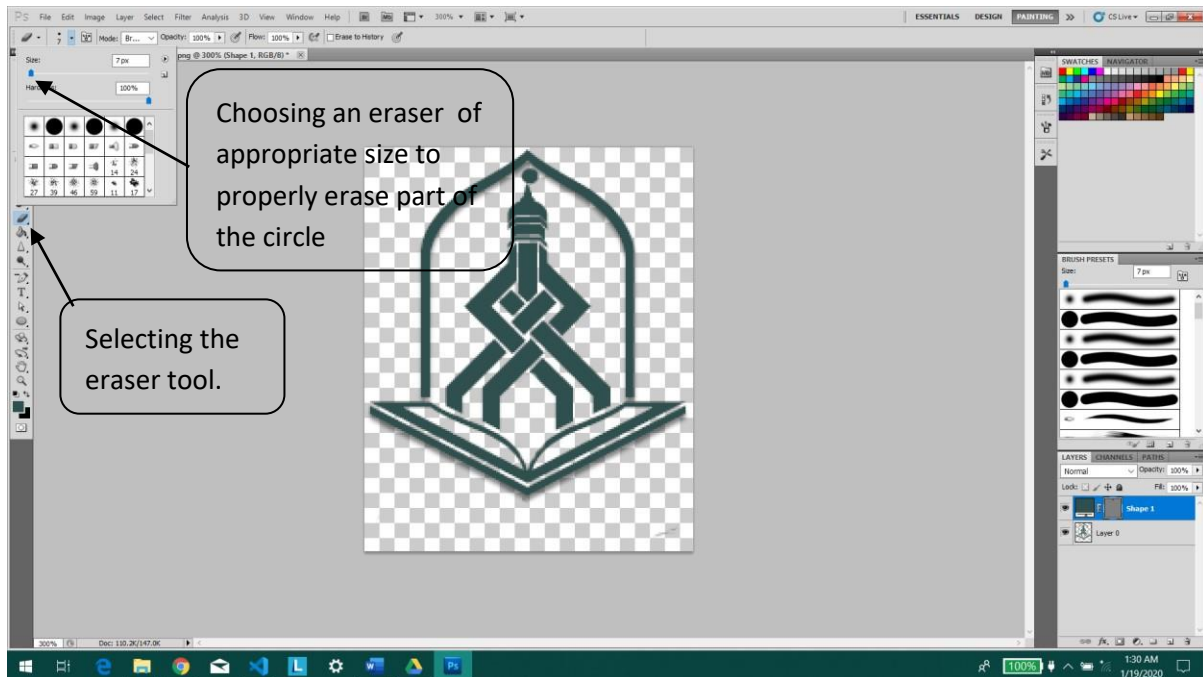
My client required a logo to represent her company, so I created a logo using some of her specified resources. First, I opened Photoshop and opened the file.

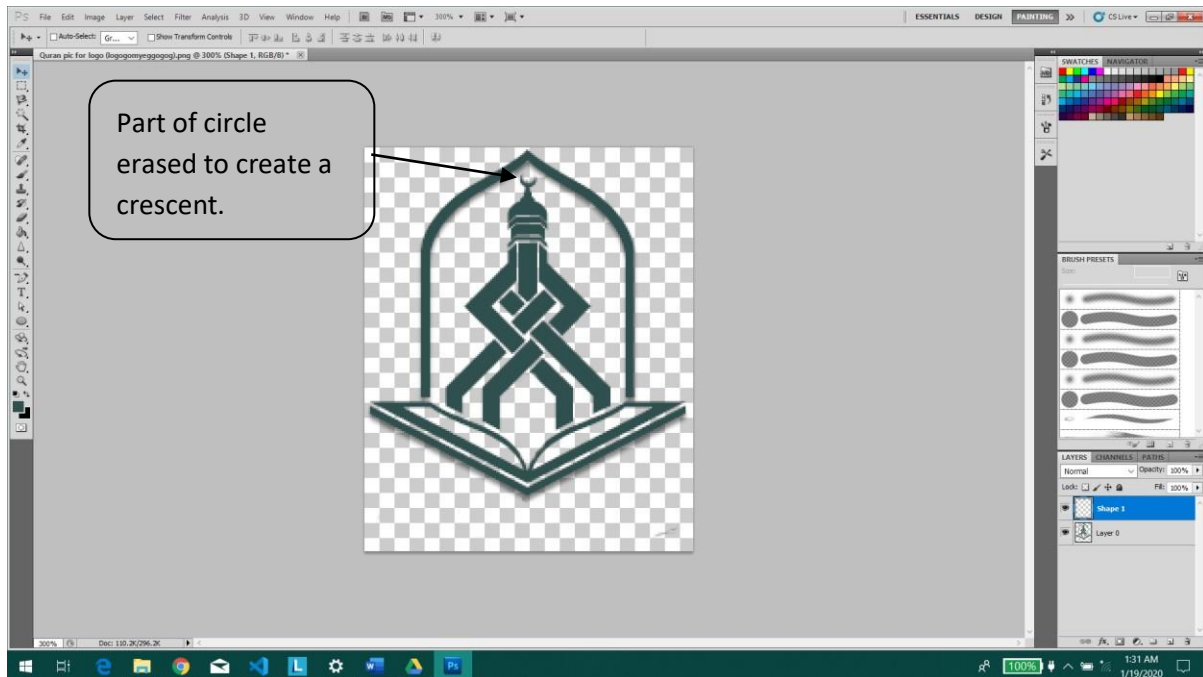




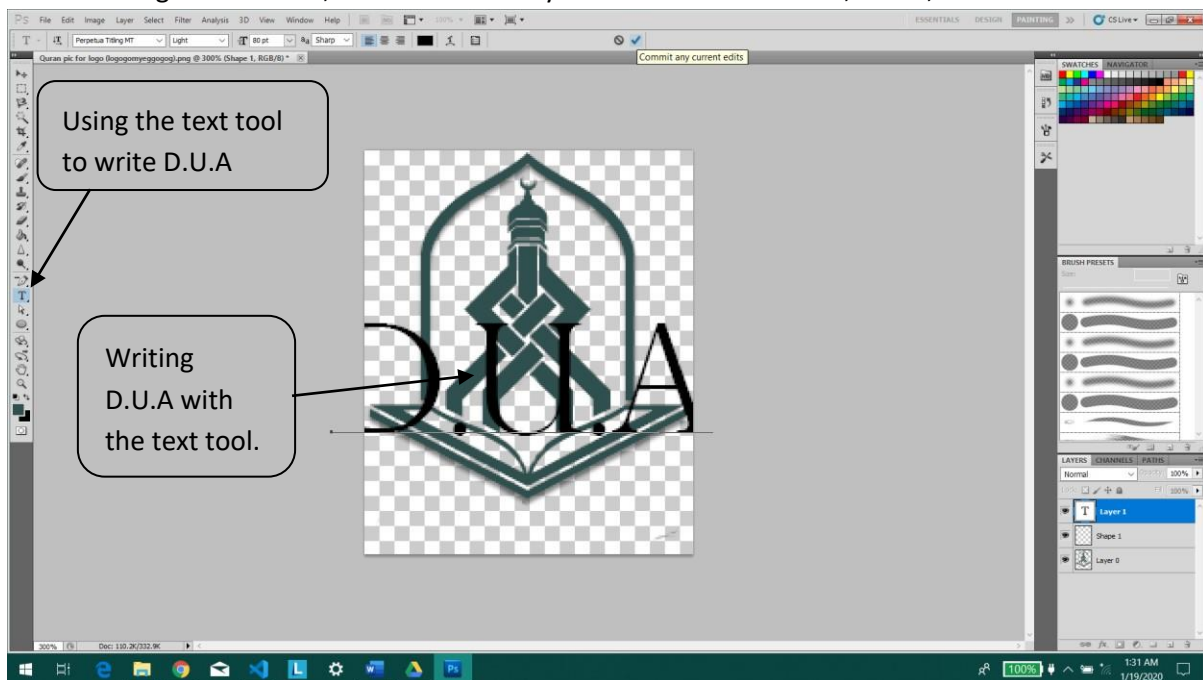
Next, I added a round circle that I would erase to create a crescent.



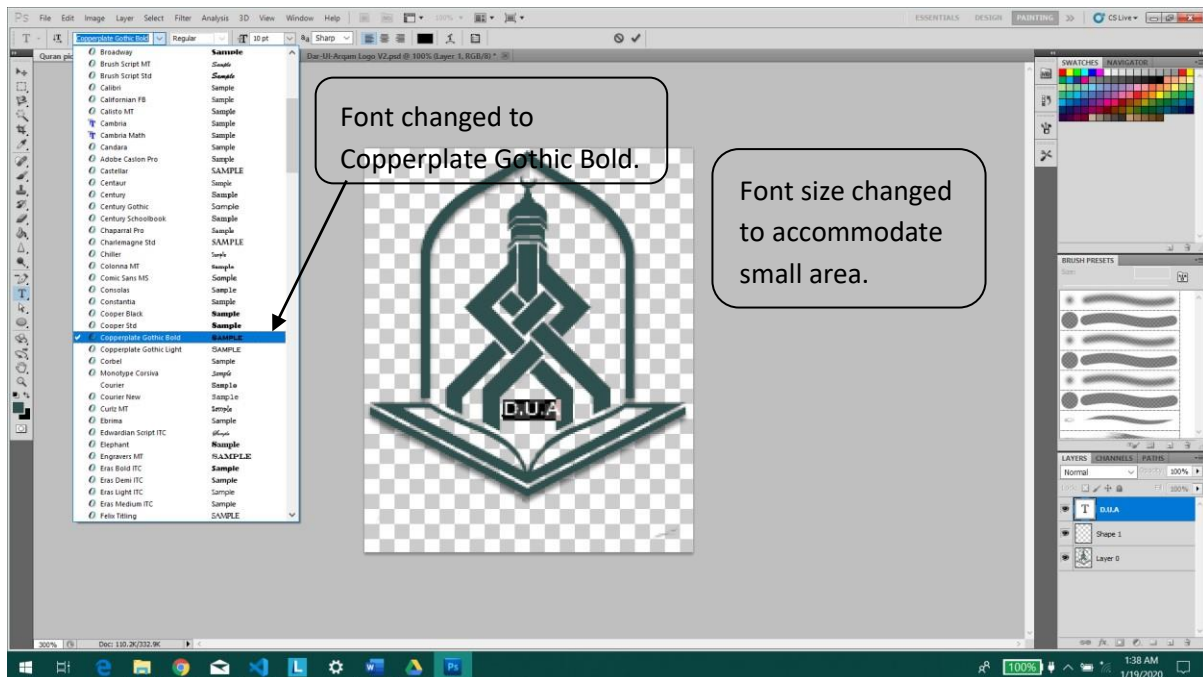
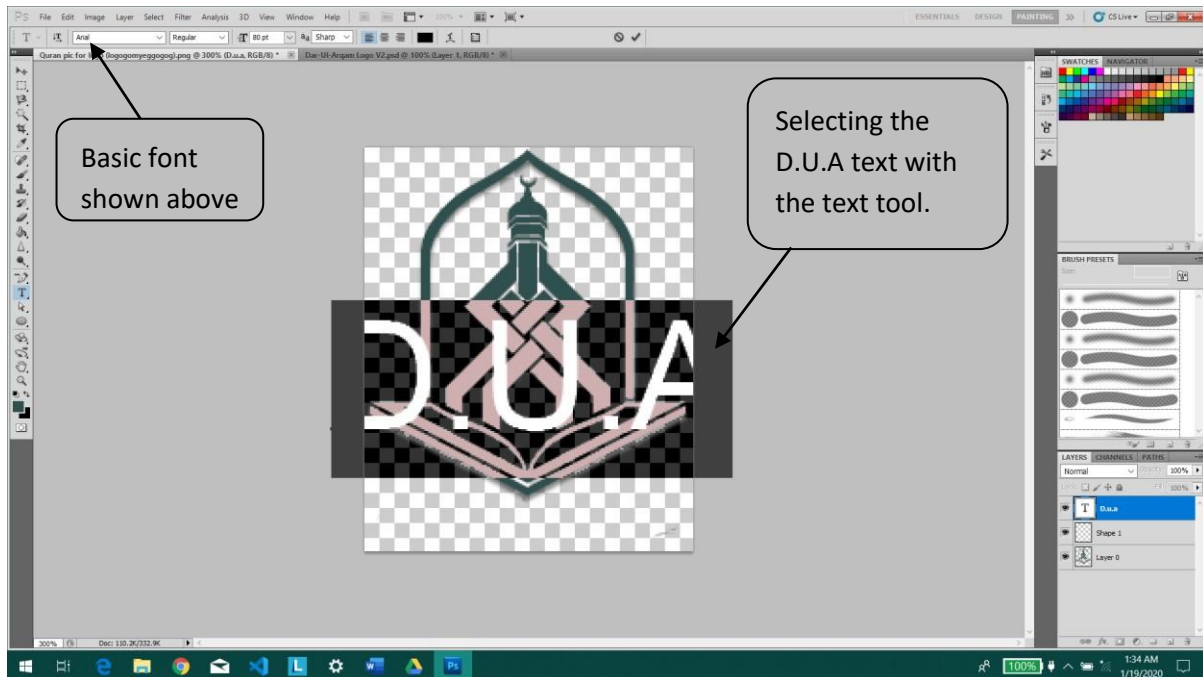




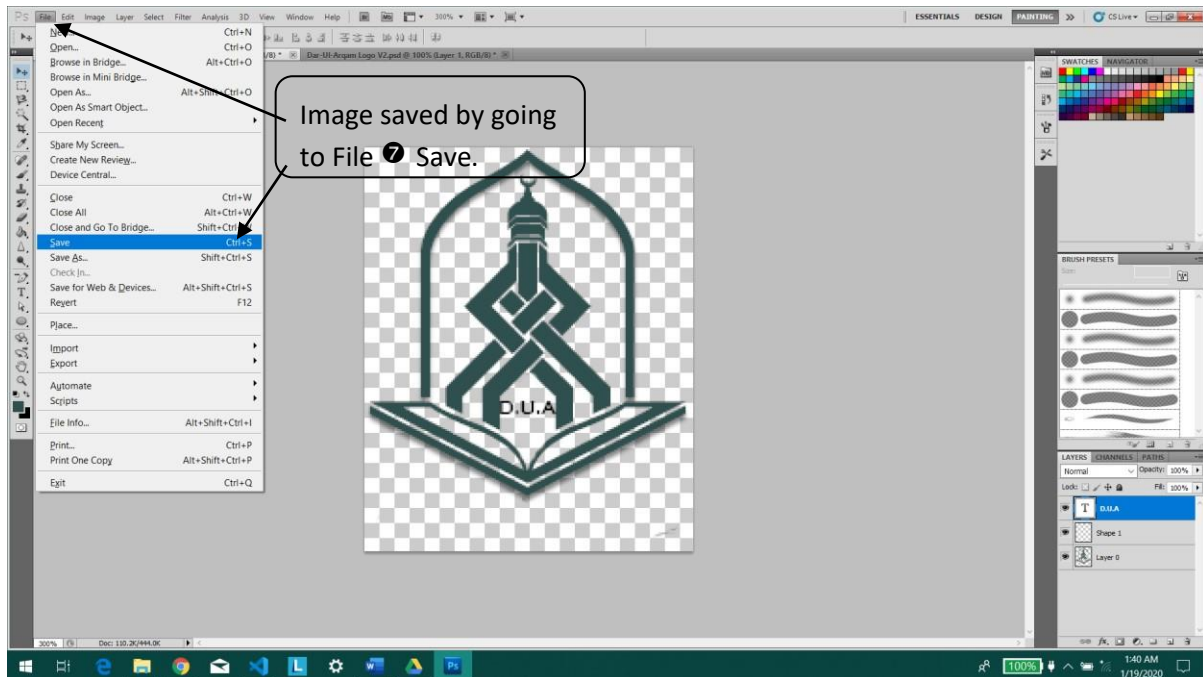
After creating the crescent, I wrote the acronym for the school's name, D.U.A, inside the minaret.



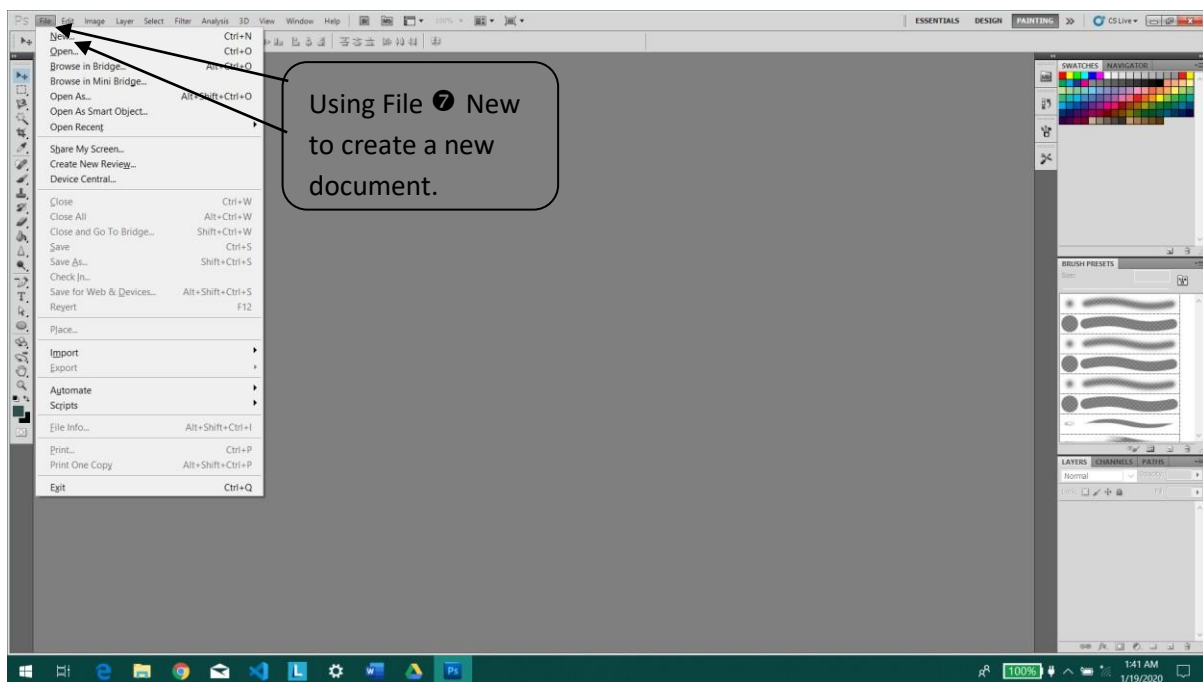




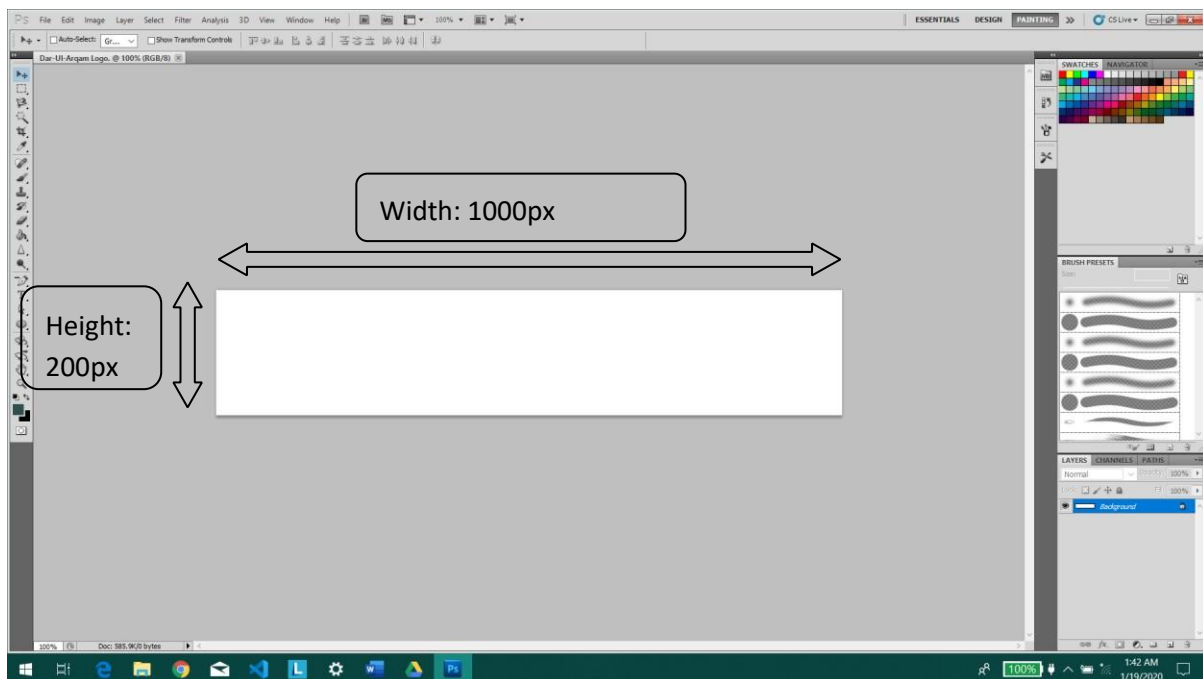
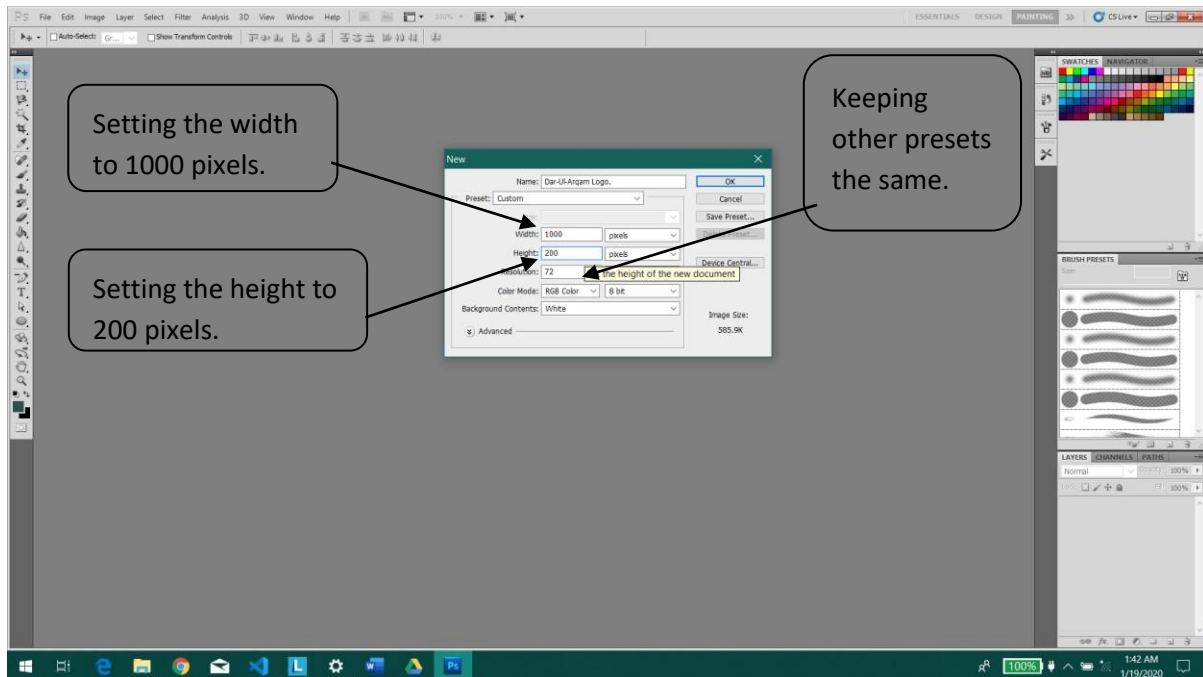
Finally, I saved the image.



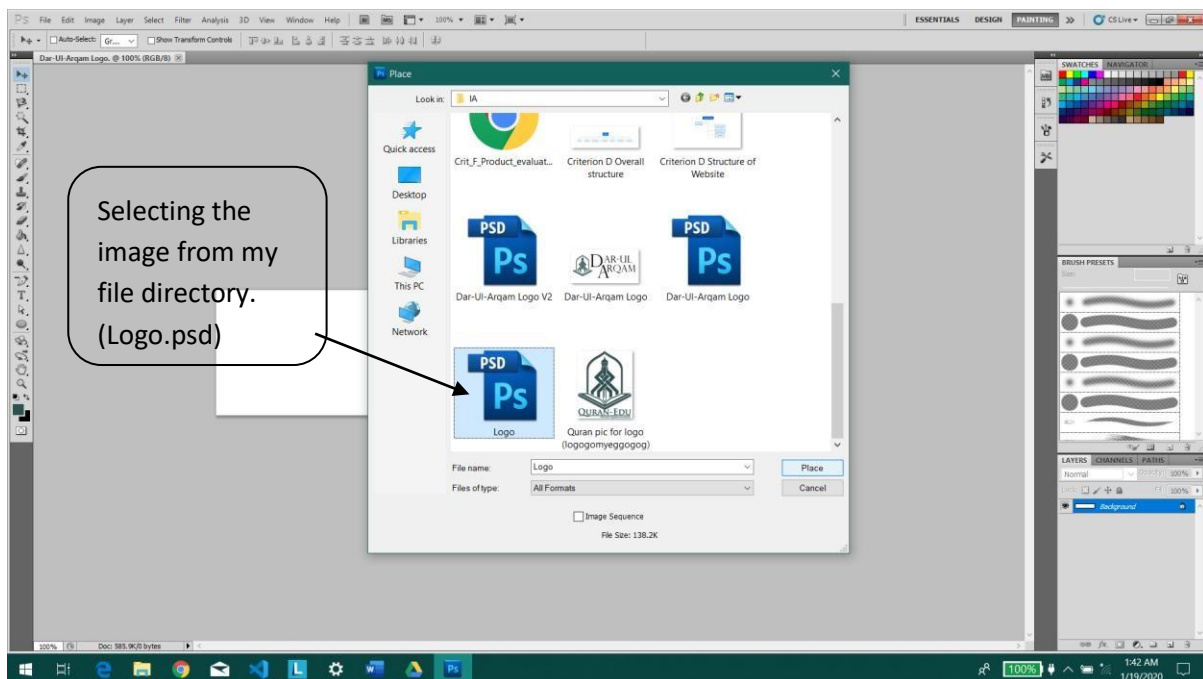
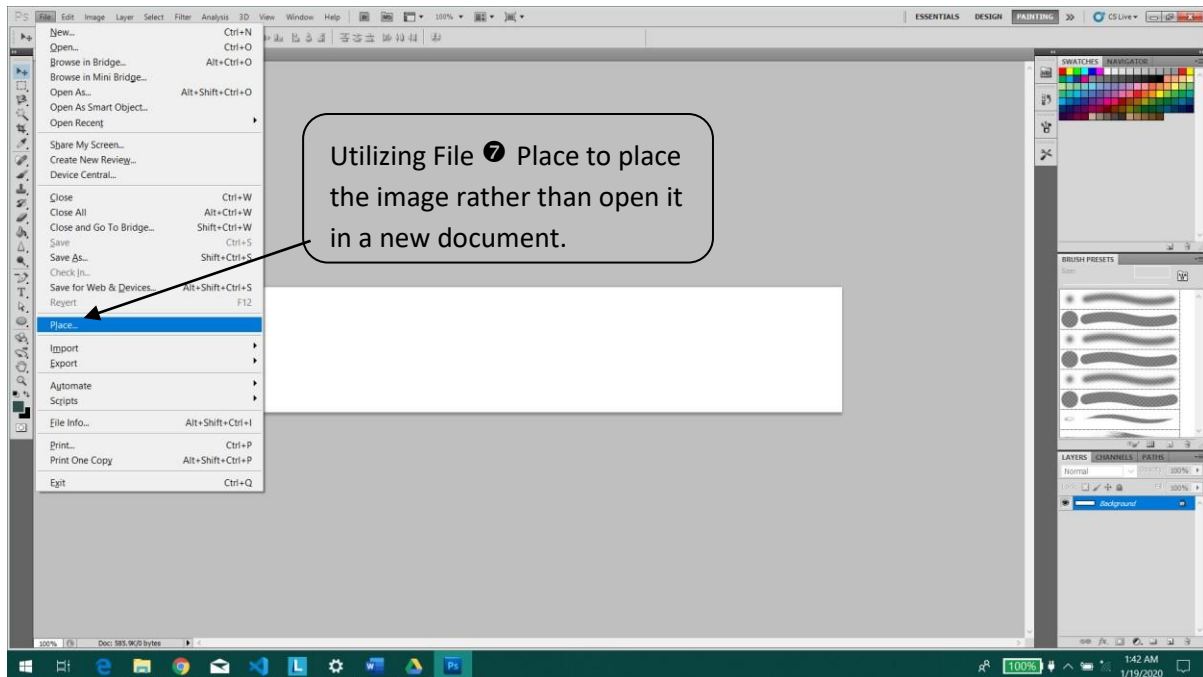
Next, I created a new blank document, 1000px in width by 200px in height. I kept all other presets the same.

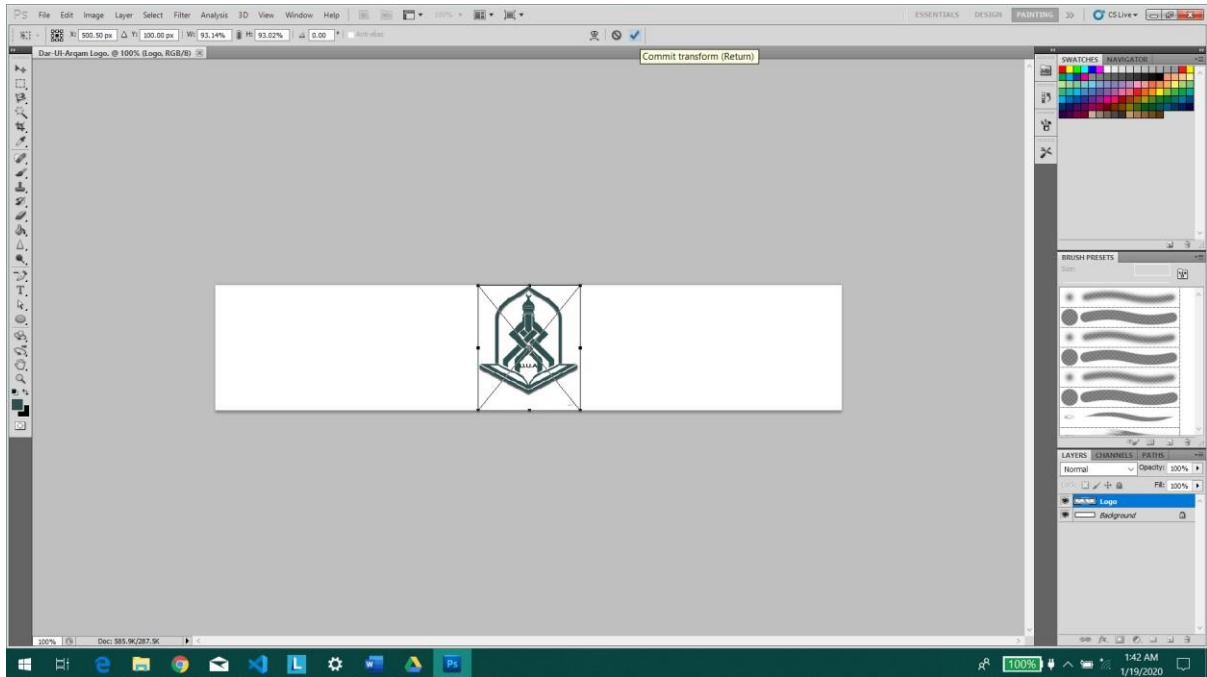




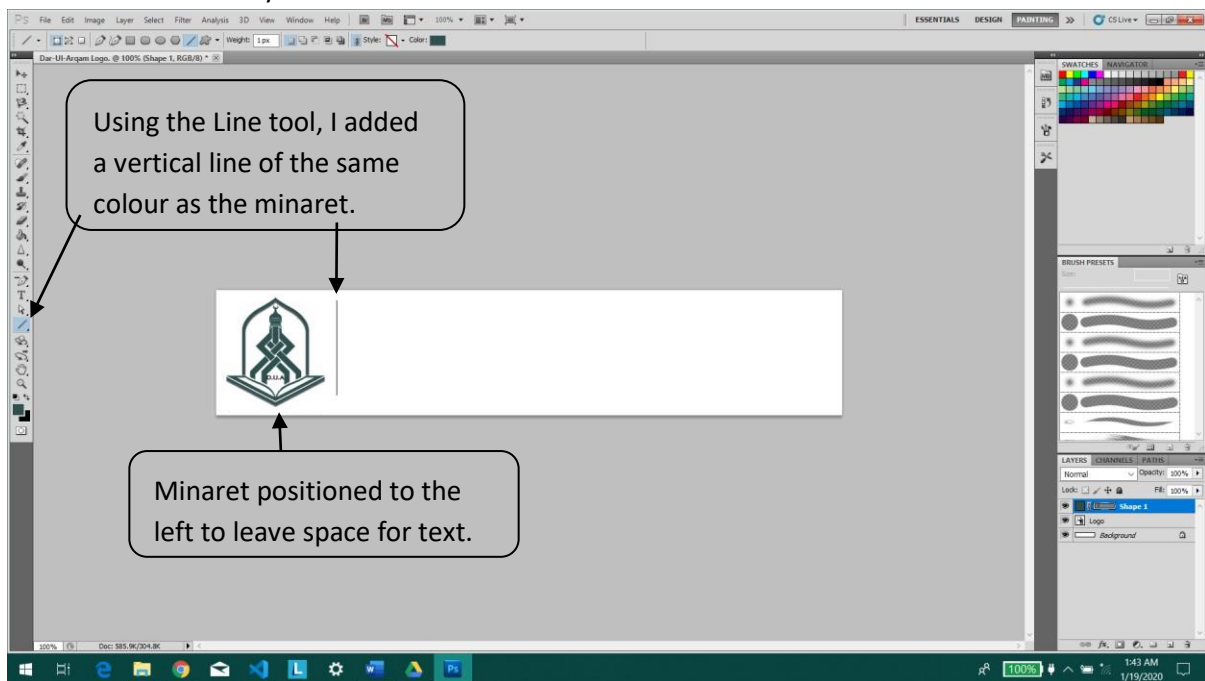


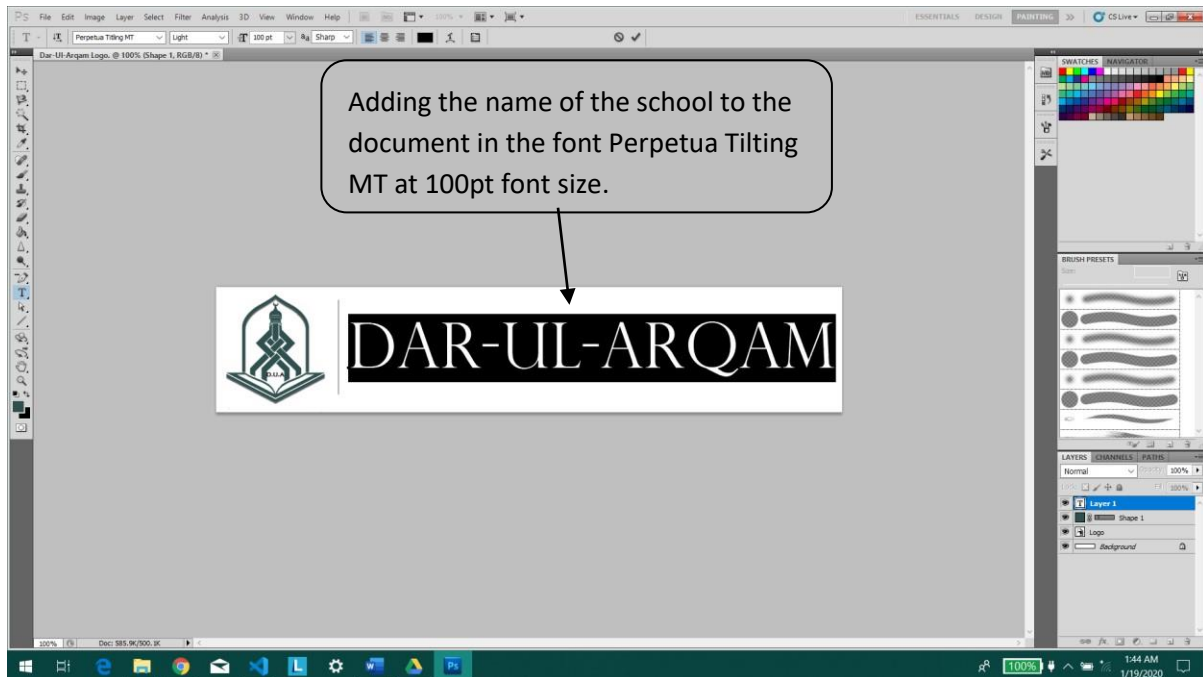
I subsequently added the previously designed minaret to the white document.



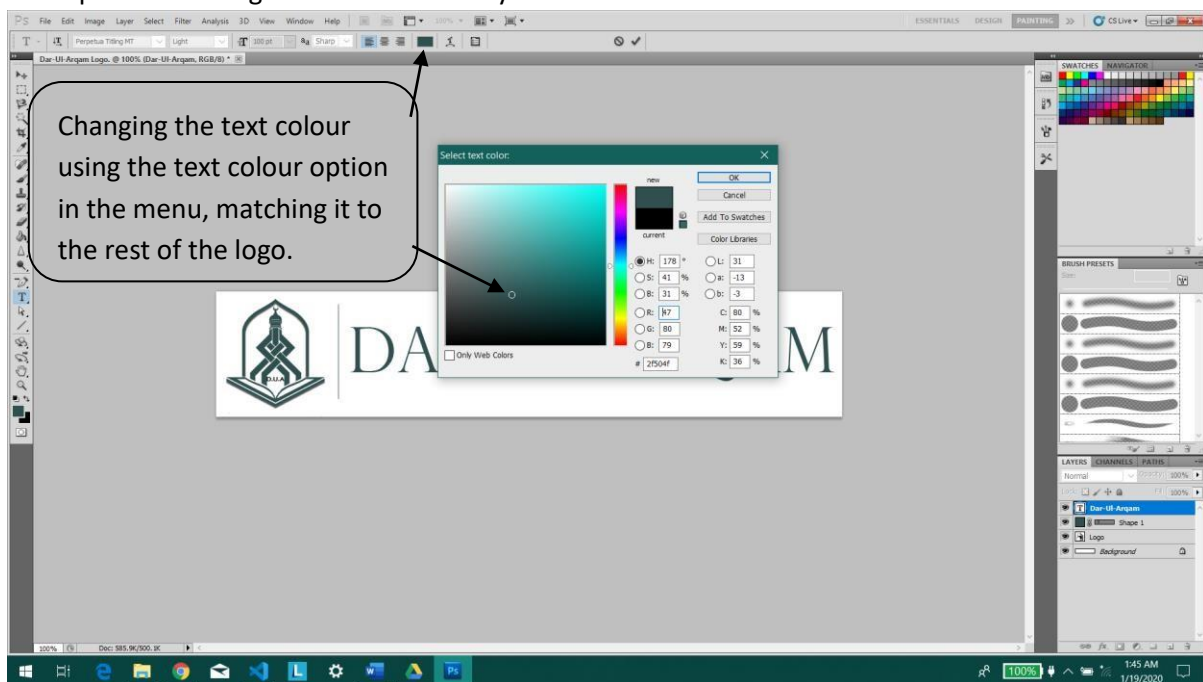


Next, I positioned the minaret to the left side, added a bar to separate the different sections, and added the name of my client's school.

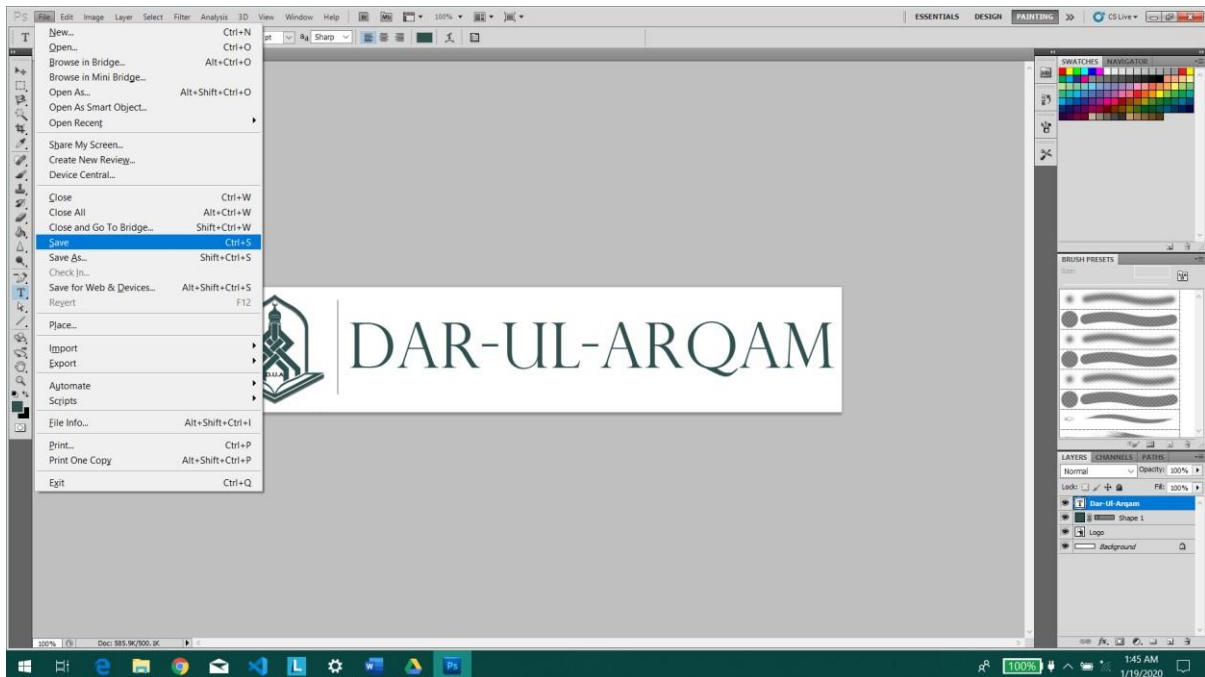




Next, I positioned the font so that it was centered in its area and changed the font-colour so that all the aspects of the logo meshed cohesively.



Finally, I saved the file as a JPEG.

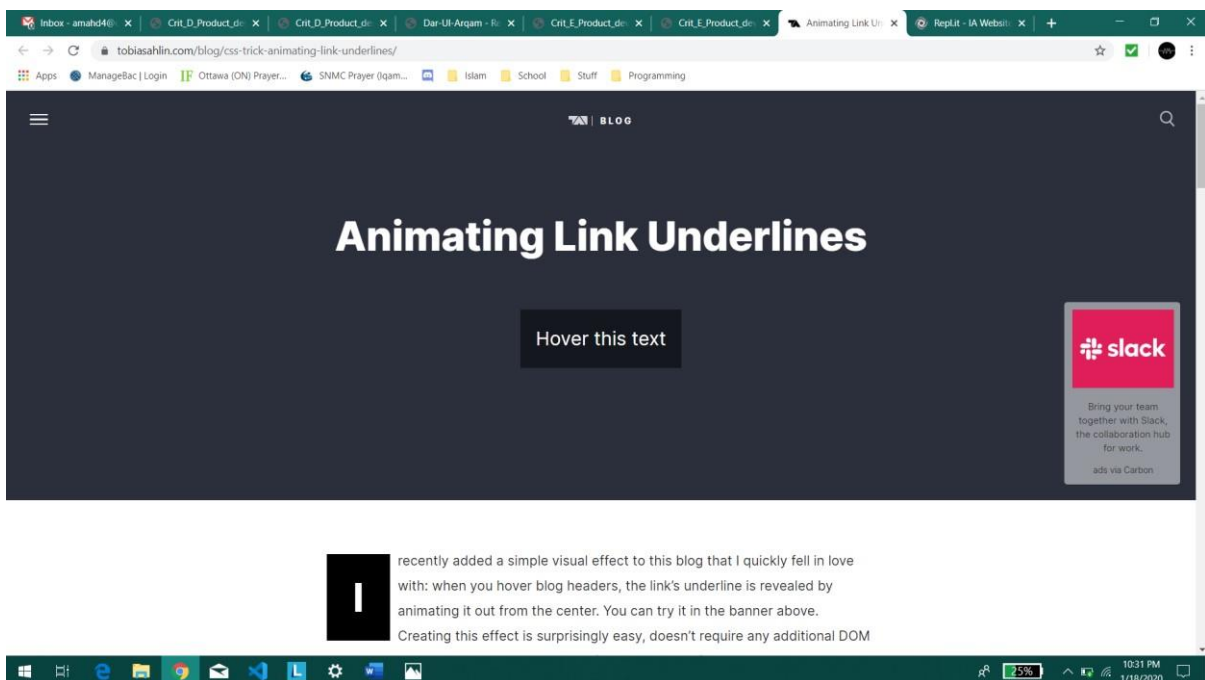


## Cascading Style Sheets

### Adding hover effect to links

My client wanted a cohesive design and aesthetically pleasing design for the website. To do that, I decided to find an aesthetically pleasing CSS hover effect and then apply it to all links, emphasizing the cohesion.

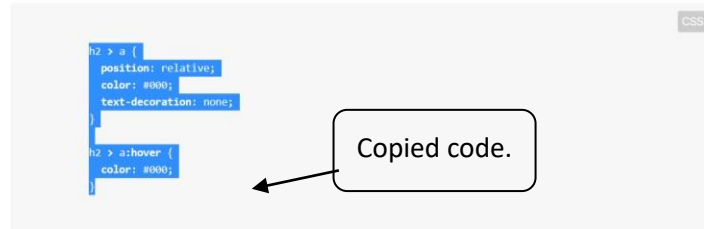
First, I found the CSS code at <https://tobiasahlin.com/blog/css-trick-animating-link-underlines/>, who distributes CSS code freely.



I copied the code and then pasted it into my code editor.

elements to be added through HTML, and falls back nicely for browsers that don't support CSS animations (it will just show up as a regular underline).

The first thing we need to do is turn off `text-decoration`, and set the link's `position` to `relative`. For simplicity's sake, we'll also make sure the link doesn't change color on hover. Here we're applying the effect to all link elements inside `h2`s:



Next, we want to add the border, and hide it through a transformation. We do this by inserting it with `:before`, and setting its X scale to `0`. As a fallback, we hide it with `visibility: hidden` for browsers that don't support CSS animations.



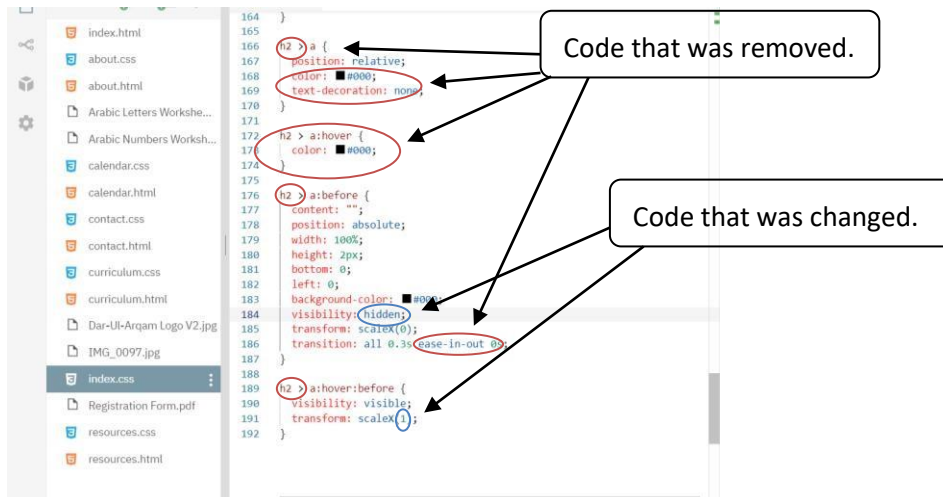
At the very bottom we tell the element to animate all changes applied to it, with a duration of `0.3` seconds. For the animation to appear, now we just need to make the element visible again on `hover`, and set its X scale back to `1`:



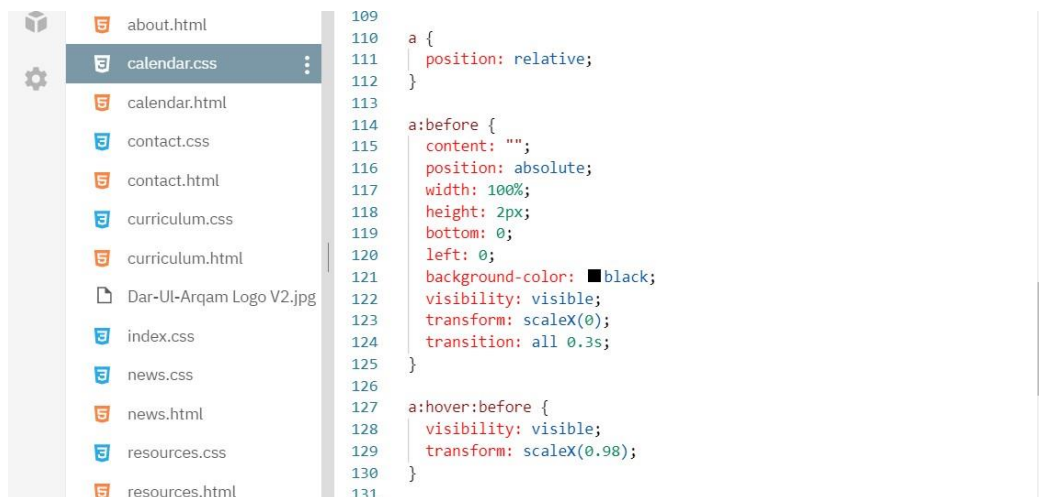
From here I edited the code so that it would function as I needed. I wanted the code to apply to all link elements, which are called "a" elements in CSS. Thus I changed the "`h2 > a`" to simply "`a`" to



apply to all links. Additionally, I removed the lines 168-169 and 172-173 as they were redundant for my links, which all already had those CSS attributes. I changed “visibility: hidden” on line 184 to “visibility: visible” because I was having issues with it on different browsers. Similarly I removed the “ease-in-out” on line 186, to accommodate other browsers. Finally, I changed “transform: scaleX(1);” on line 191 to “transform: scaleX(0.98);” because I found that for some links, it almost seemed as though the line was going over the length of the word, and 0.98 for the scaleX() value seemed to look better.



Here is the code after the changes.

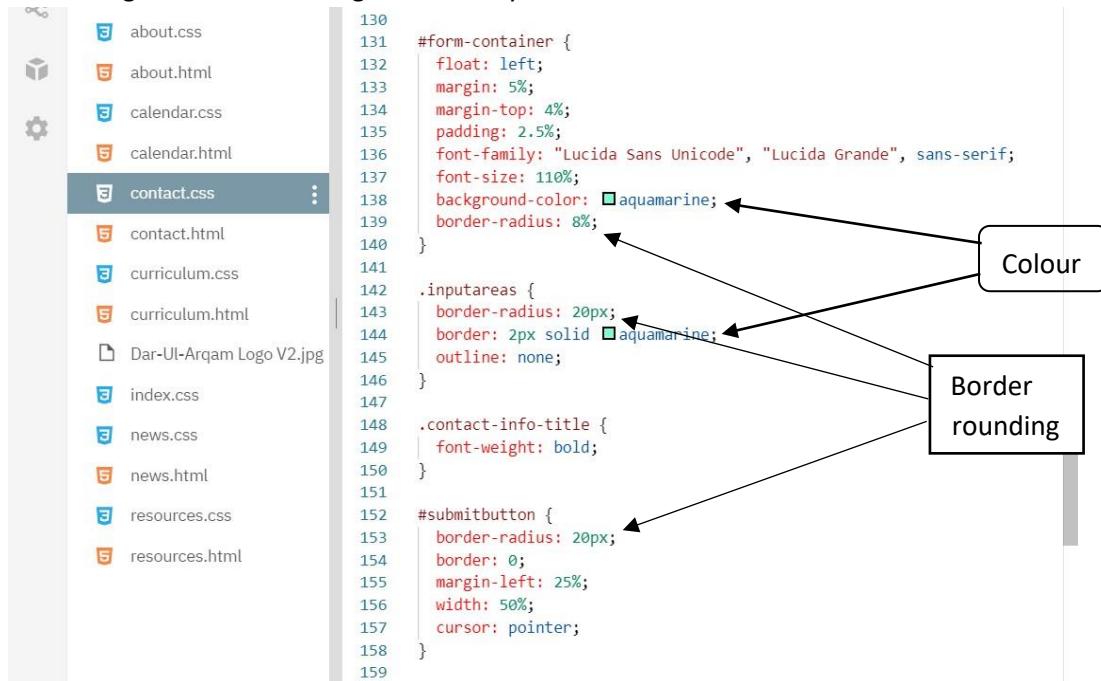


Here is a look at the effect of this code. It gives an underline to links that are hovered over that grows and completely underlines the link in 0.3s.

**OUT CURRICULUM CAL**

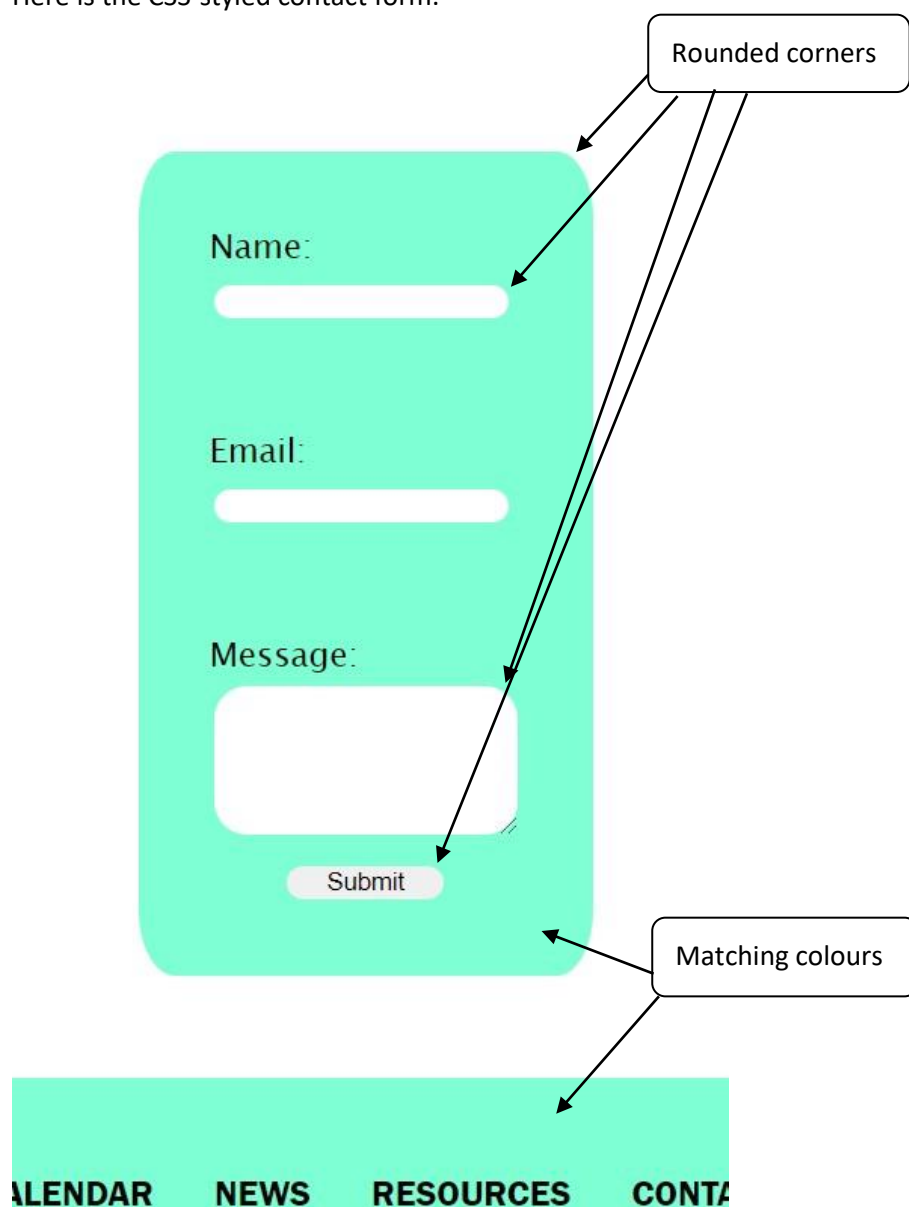
### Styling the contact form

My client wanted an aesthetically pleasing design and colour palette, that was also cohesive and matched the rest of the website. To achieve this effect, I used CSS to create a modern and interesting contact form design. Here is my code for the contact form.



The form container id on line 131 references a HTML <div> element that is the container for the form. The inputareas class on line 142 references 3 HTML <input> elements that are able to take in input. Finally, the id on line 152 references an HTML <button> element, which is a clickable button. To establish cohesive design, I matched the background colour of the form-container to the footer, and to the border of the ".inputareas" class, which are both aquamarine(line 138 & line 144). I also made all the elements in the form rounded with the CSS "border-radius" attribute. To make the form a little more intuitive, for the <button> element I made the "cursor: pointer" (line 157). This makes the cursor indicate that the button is clickable. I also turned the outline off for the ".inputareas" class as the outline was posing a problem to the design. Finally, I added some general properties to the "#submitbutton" and "#form-container" ids, such as height, width, padding, margin, font-family and font-size.

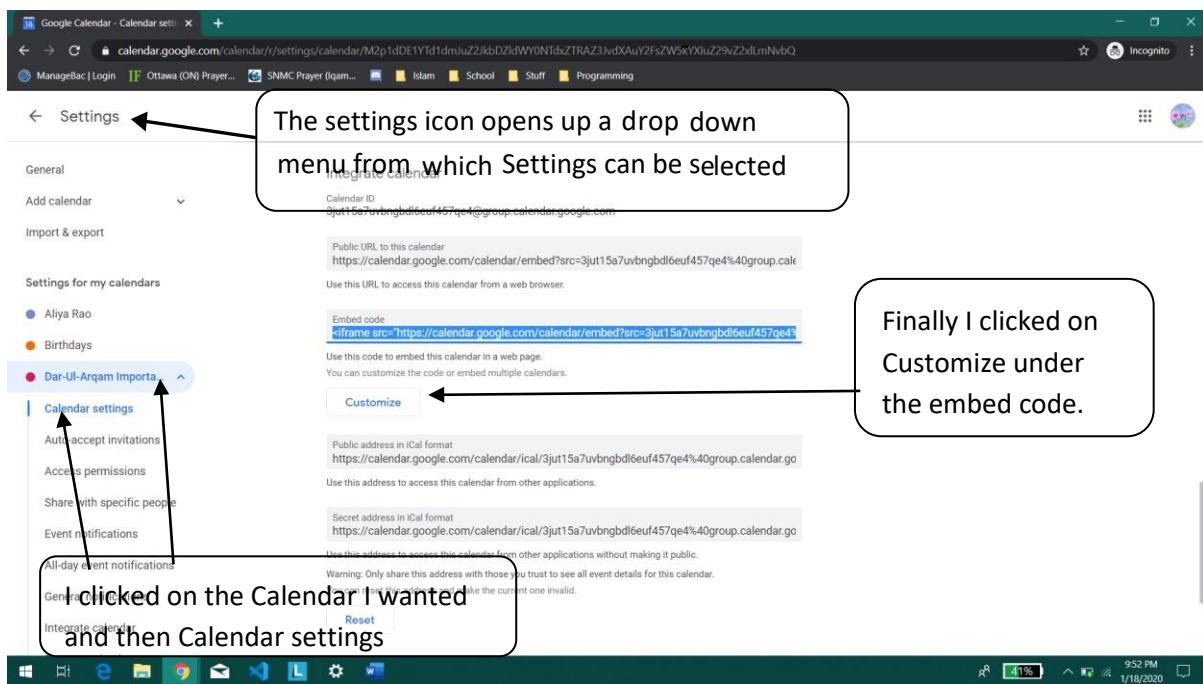
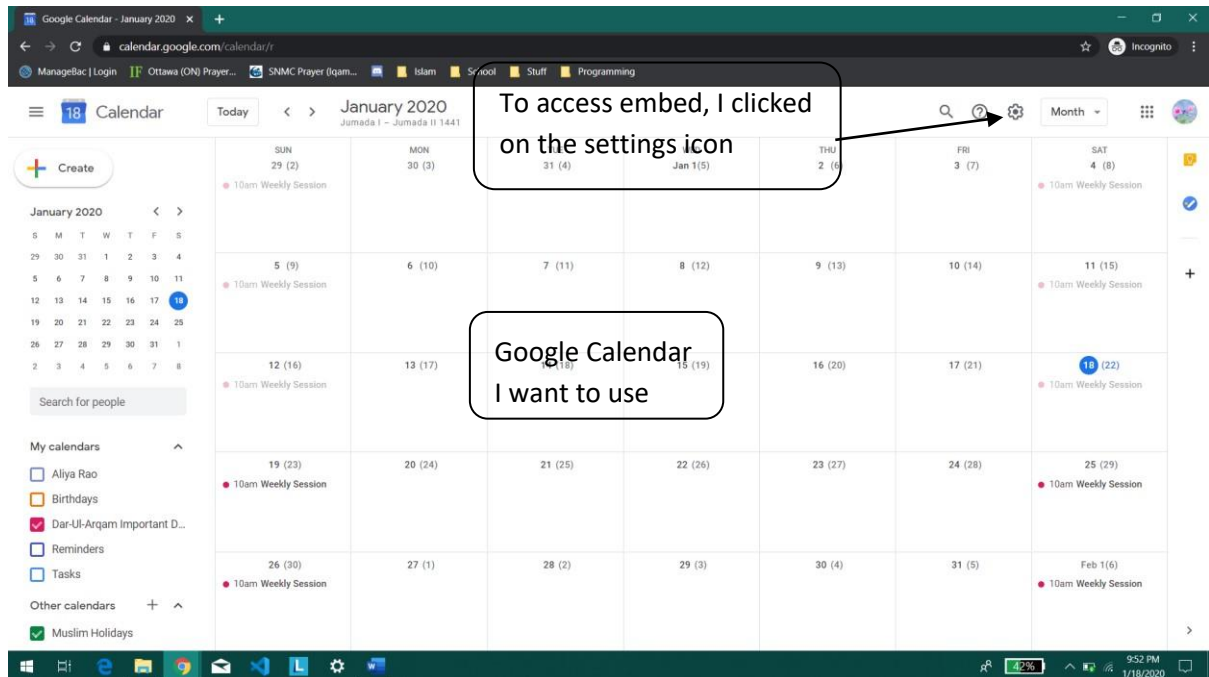
Here is the CSS-styled contact form.

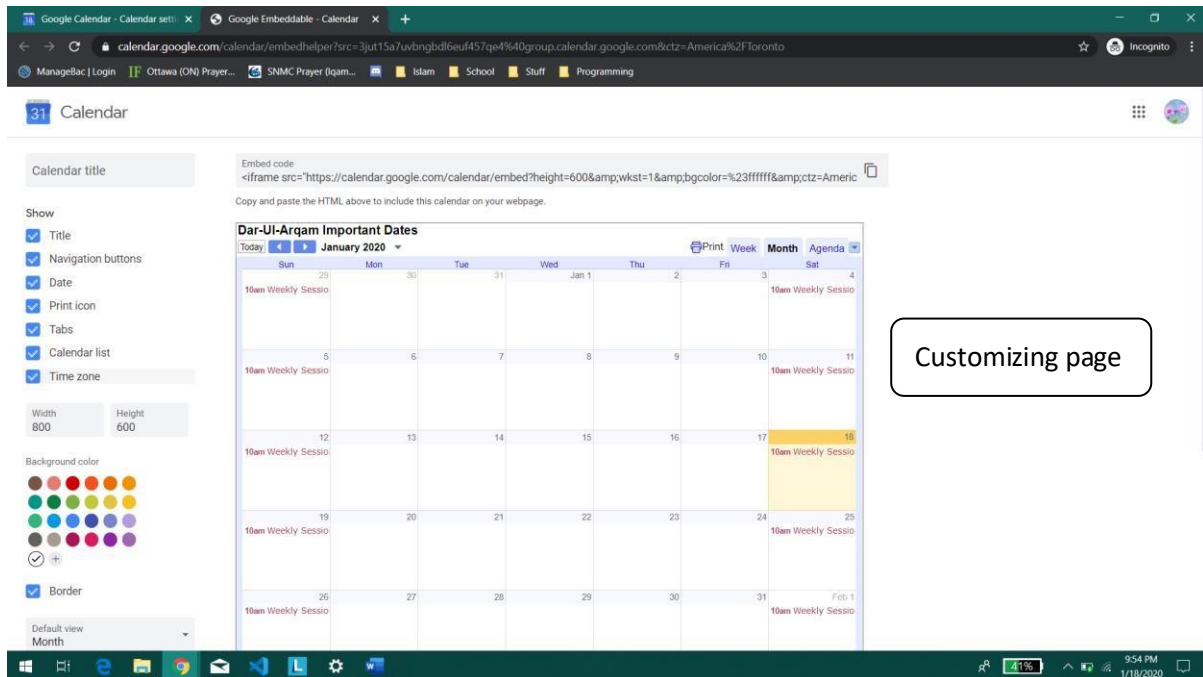


### Proficient integration of the different elements of multimedia

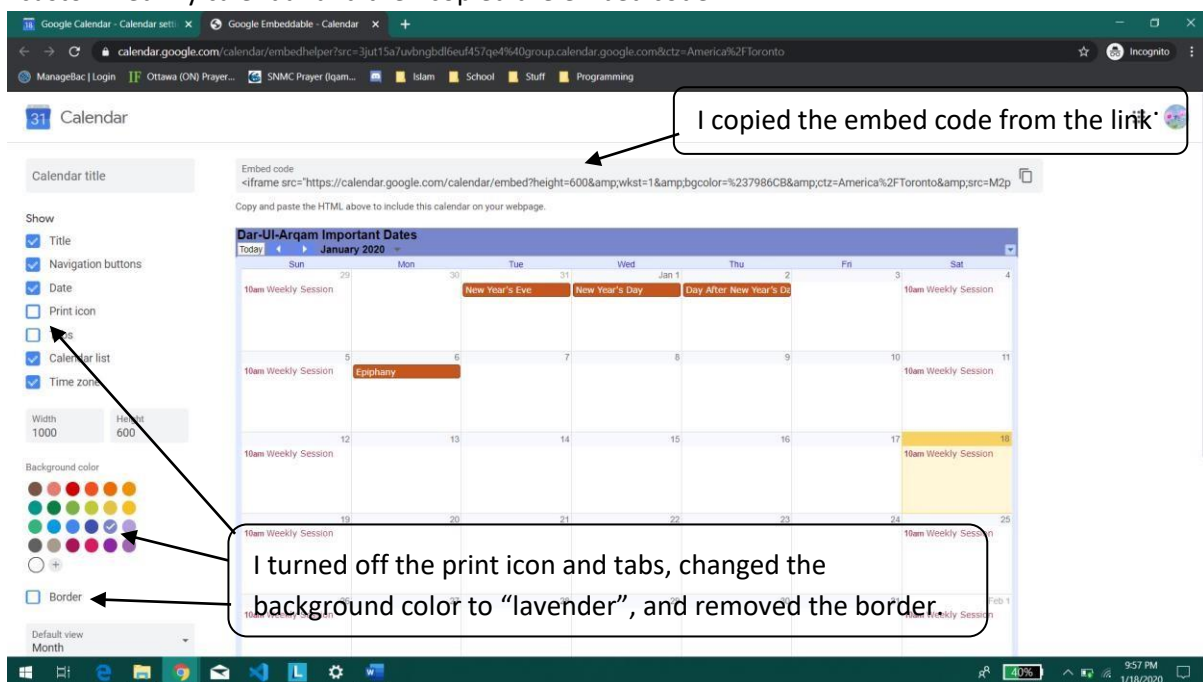
#### Embedding a Google Calendar iframe

In order to add calendar functionality as one of the SPCs my client requested, I embedded a Google Calendar iframe. First, I opened Google Calendar and went to customize the embed.

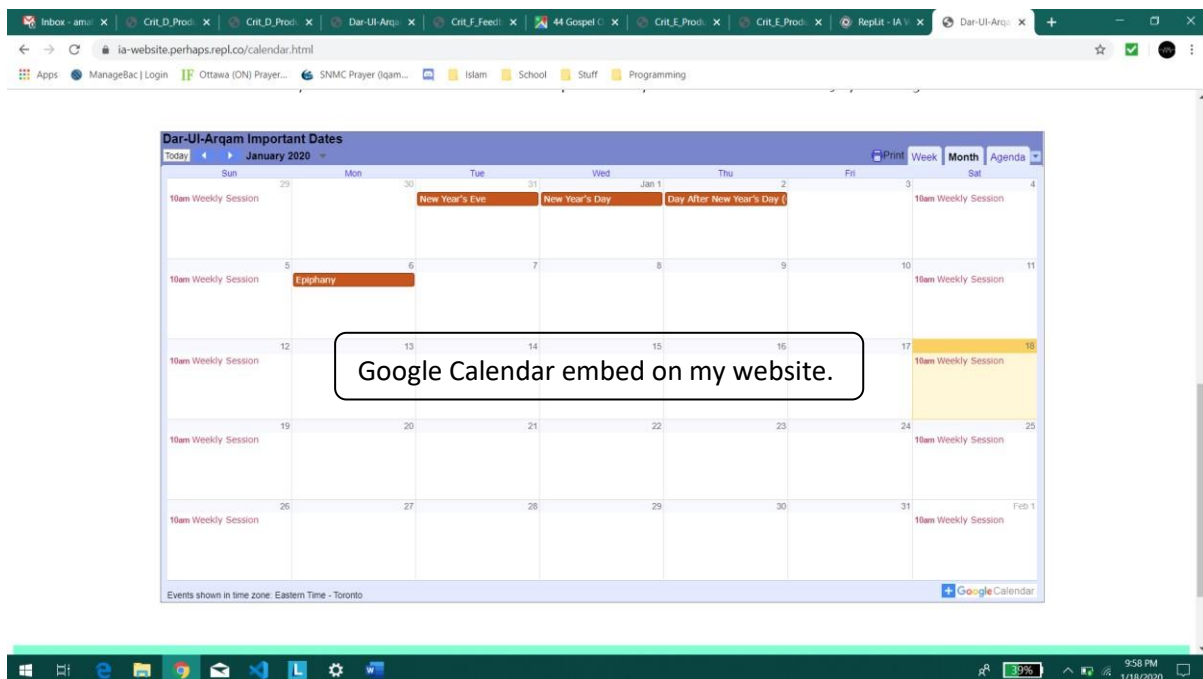
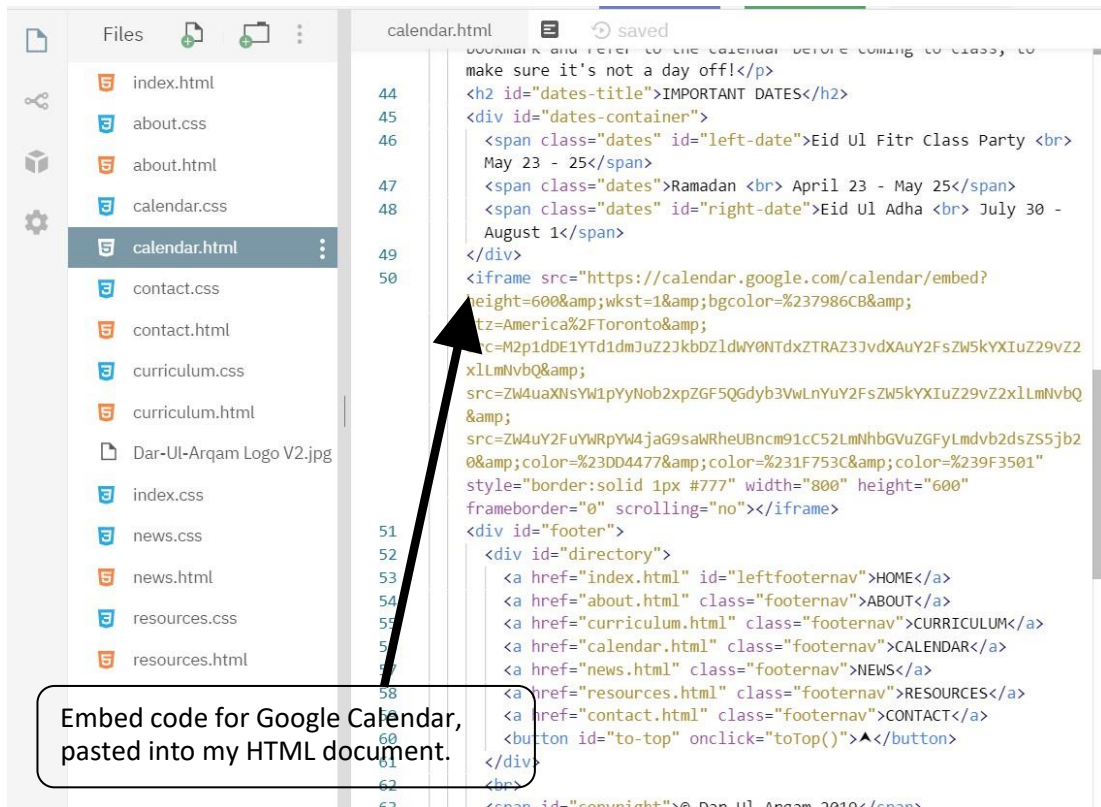




I customized my calendar and then copied the embed code.



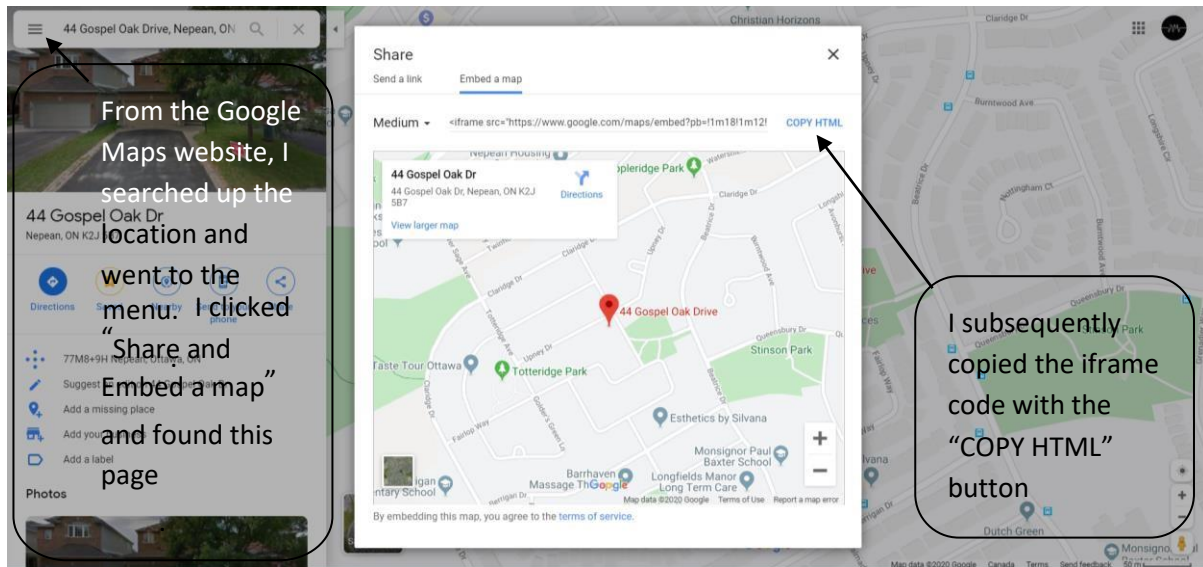
From here I pasted the code into my HTML document, and had a calendar.



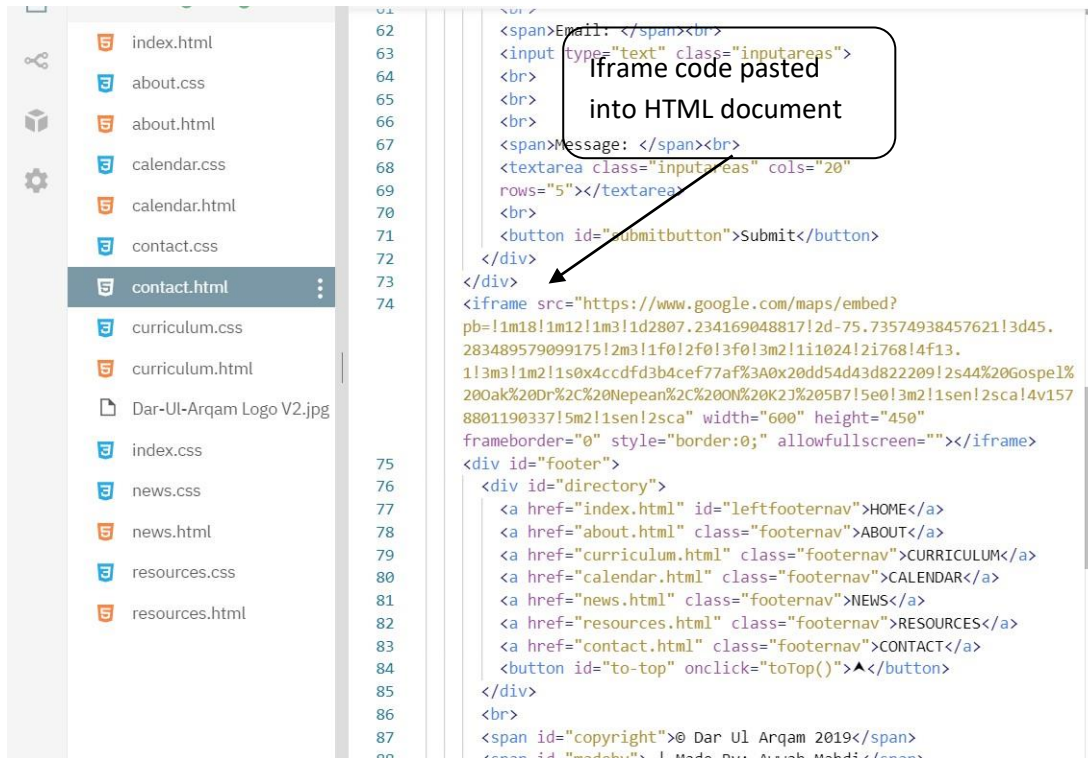
### Embedding a Google Maps iframe

In order to fully display contact details as requested by my client, I opted to include a Google Maps iframe. To do this, I first opened up the Google Maps website to the address, 44 Gospel Oak Drive, and copied the embed from "Share and Embed a map".





Next, I pasted the code into my code editor, and my Google map was now embedded.





Google Maps embed on my website

CONTACT

**Manipulation of codes such as HTML, XML, Java, JavaScript or Visual Basic to customize pages or improve functionality**

### Creating a to-top button using HTML and JavaScript

In order to emphasize the cohesive design, I added a consistent to-top button to all pages. To create it, I used HTML and JavaScript. First I used code from [https://www.w3schools.com/howto/howto\\_js\\_scroll\\_to\\_top.asp](https://www.w3schools.com/howto/howto_js_scroll_to_top.asp), but only the JavaScript topFunction().

### Example

```
//Get the button:
mybutton = document.getElementById("myBtn");

// When the user scrolls down 20px from the top of the document, show the button
window.onscroll = function() {scrollFunction()};

function scrollFunction() {
  if (document.body.scrollTop > 20 || document.documentElement.scrollTop > 20) {
    mybutton.style.display = "block";
  } else {
    mybutton.style.display = "none";
  }
}

// When the user clicks on the button, scroll to the top of the document
function topFunction() {
  document.body.scrollTop = 0; // For Safari
  document.documentElement.scrollTop = 0; // For Chrome, Firefox, IE and Opera
}
```

I used the topFunction code to allow my button to function like an <a> anchor tag.

Try it Yourself »

I copied the code and pasted it into my HTML document.

```

10      <script>
11      |   function toTop () {
12      |       document.documentElement.scrollTop = 0;
13      |   }
14      </script>

```

Function name changed

Then I used my pre-established HTML code with my JavaScript code to create a to-top button.

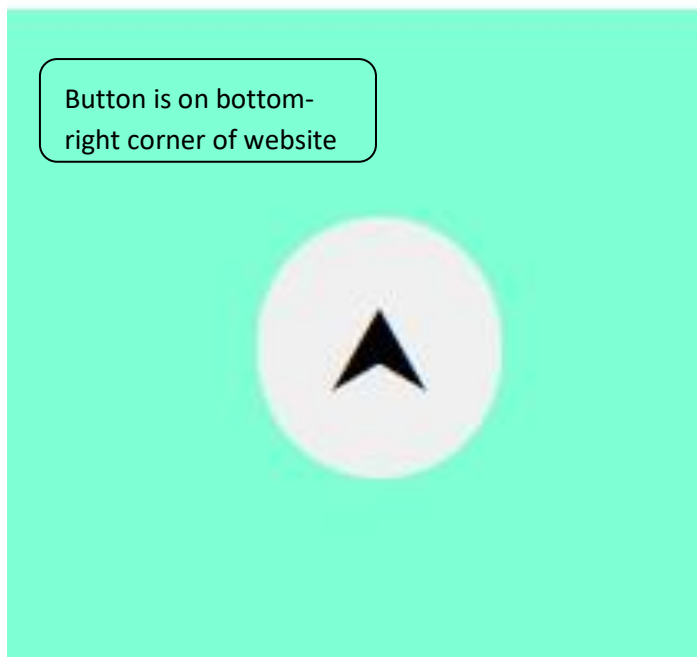
```

51      <div id="footer">
52      |   <div id="directory">
53      |       <a href="index.html" id="leftfooternav">HOME</a>
54      |       <a href="about.html" class="footernav">ABOUT</a>
55      |       <a href="curriculum.html" class="footernav">CURRICULUM</a>
56      |       <a href="calendar.html" class="footernav">CALENDAR</a>
57      |       <a href="news.html" class="footernav">NEWS</a>
58      |       <a href="resources.html" class="footernav">RESOURCES</a>
59      |       <a href="contact.html" class="footernav">CONTACT</a>
60      |       <button id="to-top" onclick="toTop()">▲</button>
61      |   </div>

```

Button with an onclick attribute that corresponds with the function name.

This allowed the button to correctly bring users to the top of the page, while also bringing in a little design.



Word count: 1015

