

COM1008 ASSIGNMENT 1

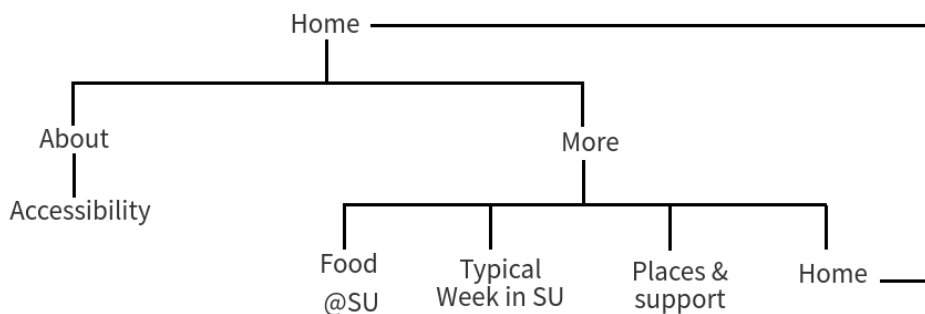
Part 2

Our website design did not change a lot during the course of development. We tried to keep it simple and consistent. The one thing that we surely changed was footer design. Instead of having there only the sitemap of the website we put more information in it like attributions as we did not find unlicensed icons. Also, we put social media links following every brands' guidelines as it is a common practice to do such things nowadays, and we included a sitemap of the website as planned before.

In a sense, we created a simple HTML template that we used across the website. Every page has the same coloring scheme and font to keep it consistent as well as they have the same header, navigation options, and footer. The only difference across pages is the content. However, we tried to make the design clean and minimalistic. We used CSS Normalize as an alternative to CSS Reset because normalizer preserves useful options instead of resetting everything as reset.css does. Also, we tried reset.css at first, but it made developer tools really filled up with an inheritance that distracted us so we chose to normalize as it does not do that (Nicolas Gallagher 2012)

We structured files that it would be easy to reach them and that there would not be any clutter. Thus, we put everything in a separate folder (i.e. we put CSS files into the CSS folder, images into the image folder, JavaScript files in js folder). Only HTML files were left in the starting folder. Such ordering made navigating between files easier as it is a nicely readable structure.

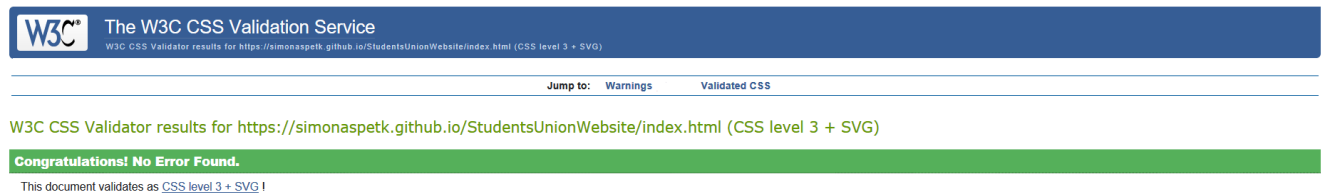
We slightly changed our sitemap, in order for it to accommodate more pages. The sitemap now has a depth of 3. The user can get access to any page within two clicks



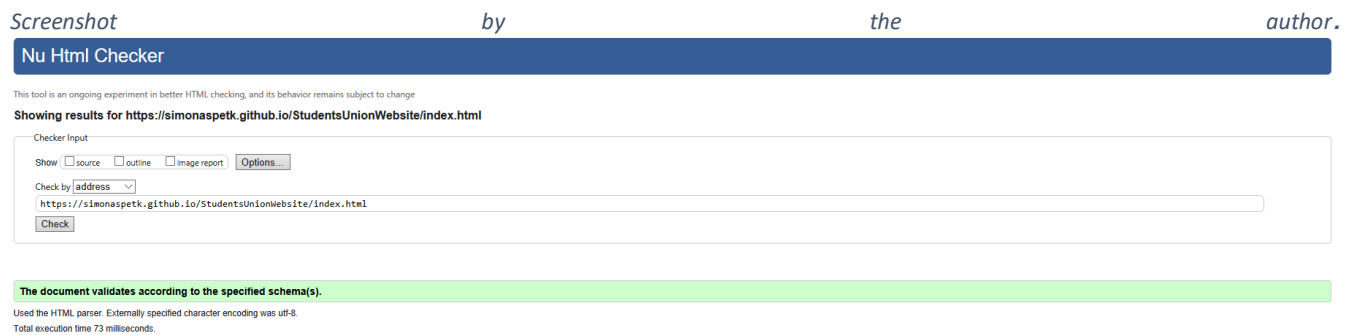
1 Figure. Our website's sitemap in the last stage.

For debugging the website we mostly used Firefox Mozilla's development tools to debug what was wrong and why Google Map's API did not work. Also, we used W3's HTML and CSS

validators to find if anything might cause errors and do we have any other problems with the code. As you can see in figures 2 and 3 we don't have any errors.



2 Figure. Validating our CSS in W3 CSS Validation Service. ©2018 W3.org. Retrieved November 25, 2018. From <https://validator.w3.org/>.



3 Figure. Validating our HTML in W3 HTML Checker. ©2018 W3.org. Retrieved November 25, 2018. From <https://validator.w3.org/>. Screenshot by the author.

We used JavaScript for only one thing which is to represent a map with the SU location and picture on the marker. We used Google Maps JavaScript API which is a free API provided by Google. In case the user uses TOR without JavaScript or he/she has JavaScript turned off for privacy purpose we took a picture of a map and show the picture instead of the live Google map.

It is really important that pages would load quickly on mobile so we considered optimization on images as some of them were loading pretty long, we equalized their dimensions for more consistent design and used free online available tools to compress those images to reduce their size.

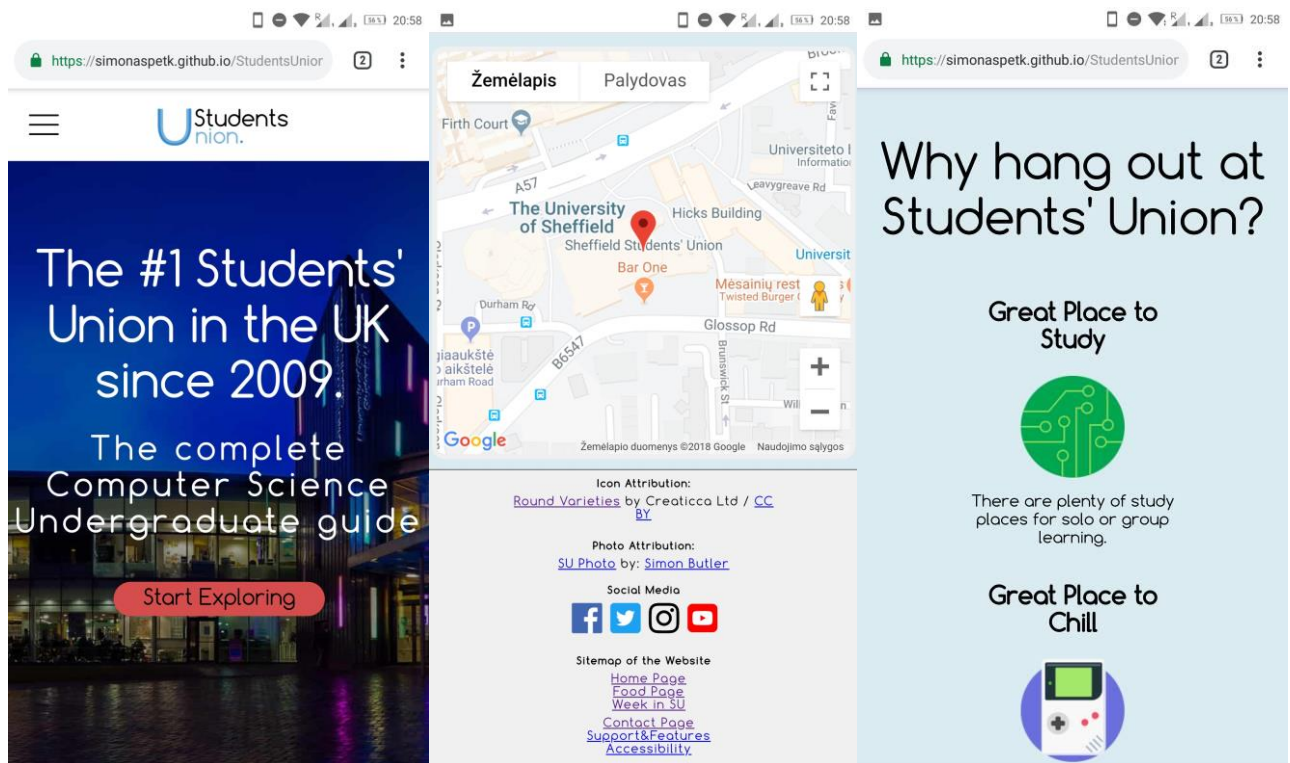
In our Contact form, we have included the required fields. Although the fields might be exploited easily with bots and might spam our emails. Also, if the spambot sends too many emails at once it might cause problems in the server. This should be taken into consideration and security should be increased in the future. Added security can be in the form of some kind of a timer added with JavaScript or a CAPTCHA. Consequently, in some cases, a contact form might be used to perform SQL injections so we would also need to defend against that to keep our data safe if we would use databases.

Another security issue that we have is exposure to Simonas' Google Maps JavaScript API key. However, he has put restrictions in the Google Cloud Developer console to the API key that it could not be exploited by others. Because people that would steal the API would not know what kind of restrictions have been applied and they would fail to use it. The sensible thing to do

would be to put restrictions on the website address, however, Simonas cannot do that because that way it would not work for you just by opening index.html from your computer.

In terms of HTTP and HTTPS, our website would use HTTPS because it is superior to unsafe and old HTTP. HTTPS not only safer due to the encryption of the communication protocol ('HTTPS', 2018) but also it has Search Engine Optimization benefits as most engines downgrade the ranking of websites that are using HTTP ('Hypertext Transfer Protocol', 2018).

The website was tested in various browsers and devices. We uploaded the website on GitHub and hosted it from there on HTTPS protocol. The website was tested on iPhones as well as on the Android mobile devices in browsers such as Chrome, Firefox, Safari, and Brave. Also, the website was tested on desktop browsers such as Firefox, Safari, Chrome, Brave, and TOR. The website was tested on different screen sizes and it let us identify and fix a lot of design bugs that we wouldn't have found otherwise.



4 Figure. Testing the website on Android Phone's Google Chrome browser.

References:

Nicolas Gallagher., (2012) About normalize.css [online]. Nicolas Gallagher. [Viewed 24 November 2018]. Available from: <http://nicolasgallagher.com/about-normalize-css/>

2 Figure. W3 CSS Validation Service. ©2018 W3.org. [Retrieved November 25, 2018]. Available From <https://validator.w3.org/>. Screenshot by the author.

3 Figure. W3 HTML Checker. ©2018 W3.org. [Retrieved November 25, 2018]. Available from <https://validator.w3.org/>. Screenshot by the author.

'HTTPS' (2018) *Wikipedia*. [Viewed 24 November 2018]. Available from: <https://en.wikipedia.org/wiki/HTTPS>

'Hypertext Transfer Protocol', (2018) *Wikipedia*. [Viewed 24 November 2018]. Available from: https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol