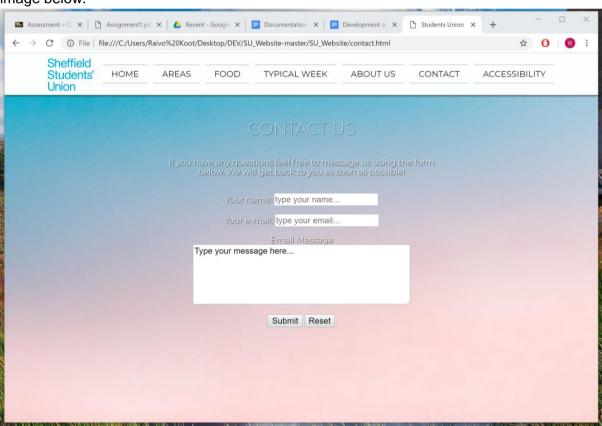
Development and Testing

Because reset.css removes to much of the starting borders we felt, we started off development with a normalised style sheet so that the resulting website would look the same on all browsers. We used a divide and conquer strategy on making the website. Raivo started off with designing the top menu system and hamburger menu, while Will designed the tile system to hold content within the website itself. The top menu system was simple to create as that was in the <header> element, but as soon as the hamburger menu was added, it disturbed everything. Not only was the hamburger menu hard to create, but hard to implement.

The tile system posed several issues. Flexboxes and flex were very tricky to set up properly as the alignment was crucial. And for different sized viewports, font size and display layout would have to change. This was achieved with different @media queries which would alter flex columns to rows and change font sizes. Furthermore, we ran into problems with the content of our tiles overrunning the height of our tiles at the bottom.

As for colors, we started out with a simple white and blue color scheme as shown in the image below.



Soon enough though, we realised this color scheme was causing difficulties to read text and it lacked personality as nothing ever stuck out very strong to keep clear divides. As a result, we changed our scheme accordingly to something more punchy.

Media were limited to two different sizes to make development simpler. The background image has a width of 1920px in the file size as the largest common screen size is 1920x1080, so going larger than this would mean unnecessary data. The images for tiles all

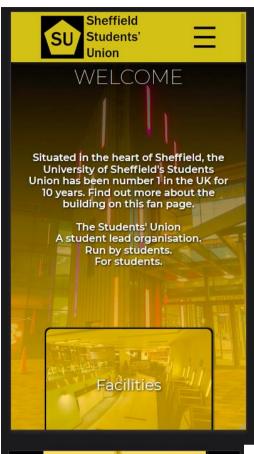
have a width of 800px across, as this adequate for all screen sizes while not being ridiculously big. If i had kept all the photos the size Will's camera records at (5184 by 3456px), the loading times would've been painfully slow. I also set the export quality to a medium value to decrease file size by increasing compression rate.

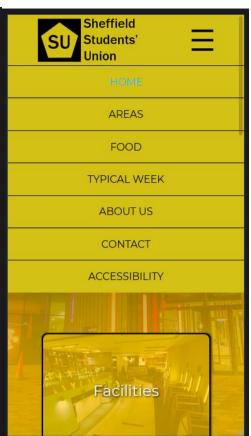
On a side note, the css file areas.css serves as a base css file for most of our pages.

Testing

We designed the website around 3 different widths of view ports: Large desktop, thin desktop and mobile. We chose to design the websites for a 1080p 24 inch screen on full screen (large desktop), a 1080p screen on half width (narrow desktop) and all types of mobile size presets in mozilla firefox's development tools (mobile). At some rather deep into development we ran into problems with screen sizes: One of the screens we were developing on is a 14 inch 3000x2000 screen. Everything worked well on it until we opened the site on a normal 16:9 monitor again where content was suddenly overflowing and not lining up correctly. Fortunately, We corrected our problems soon enough.

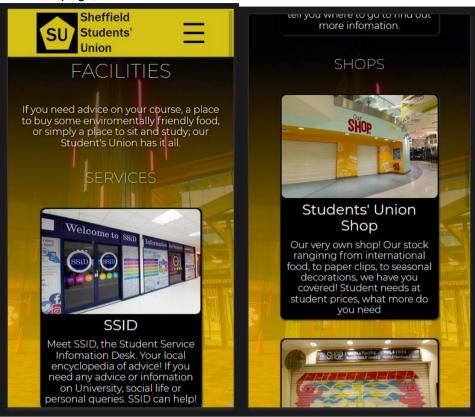
The following are screenshots of the index page on all viewport sizes. Index splash page, mobile:



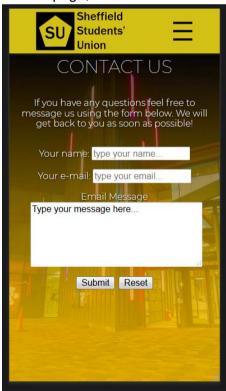




Facilities page, mobile:

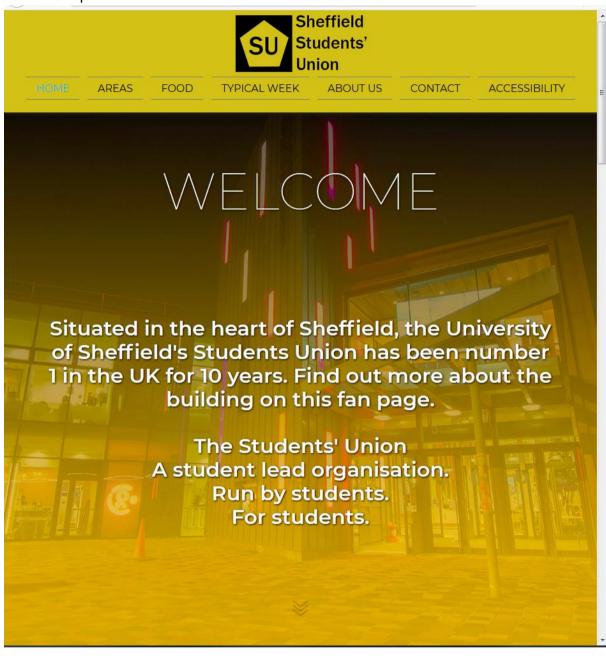


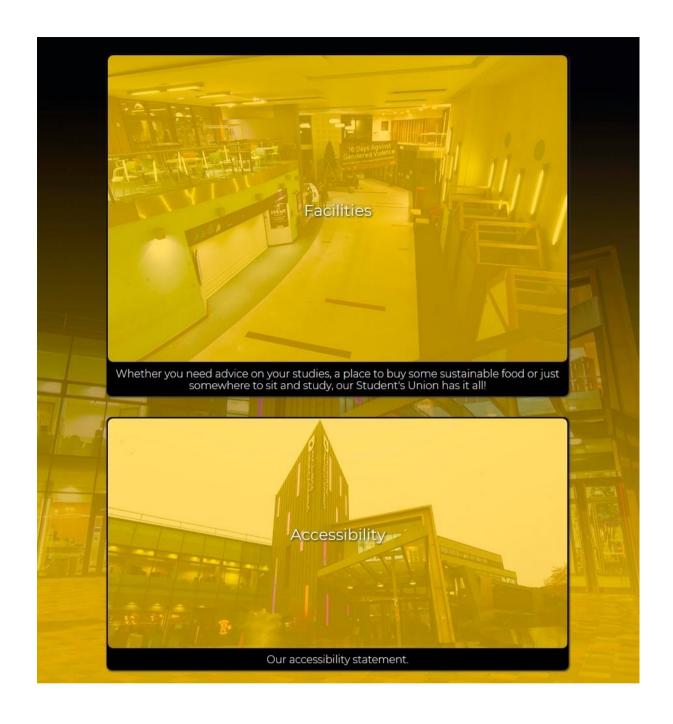
Contact page, mobile:

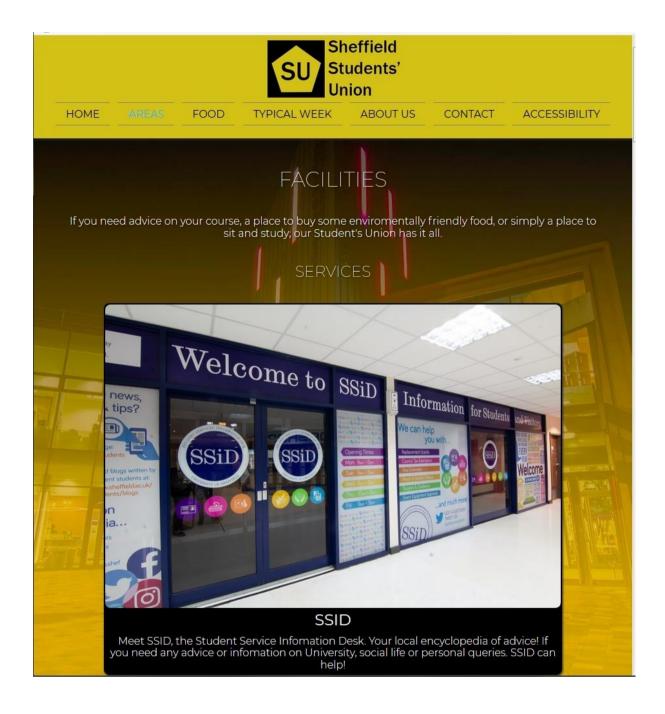


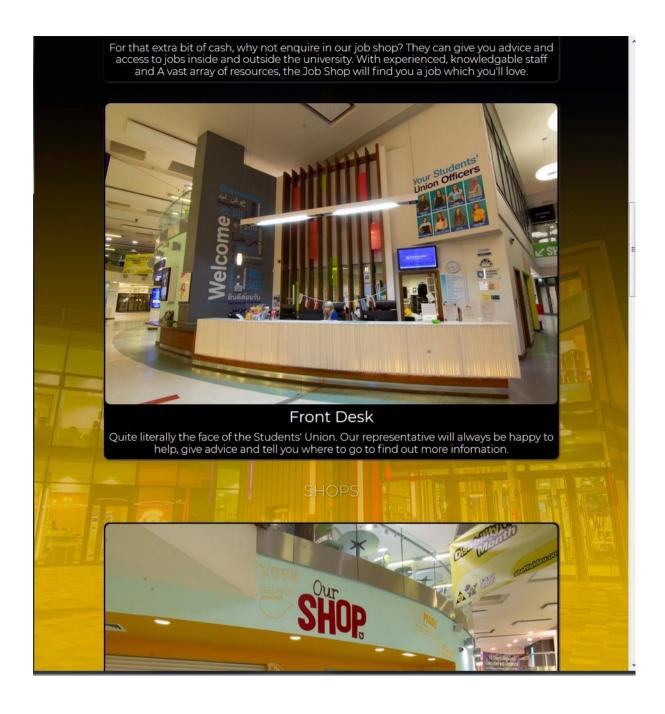
As we used mobile first design, the website on mobile all works well.

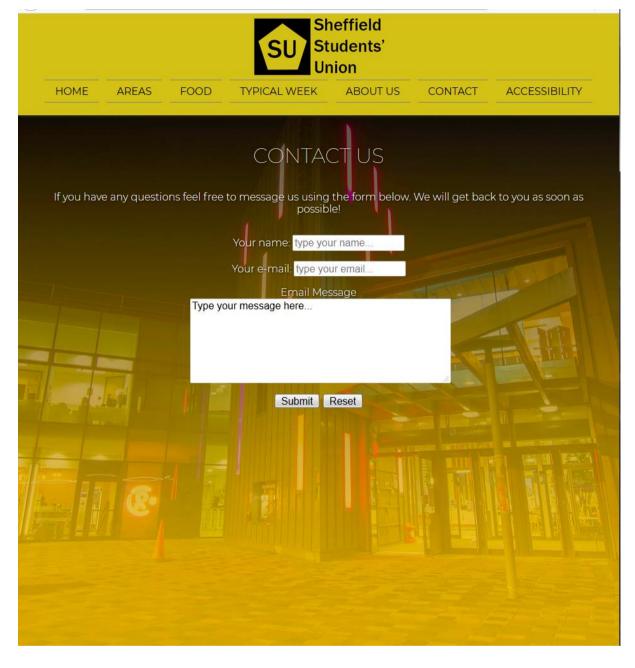
Thin desktop





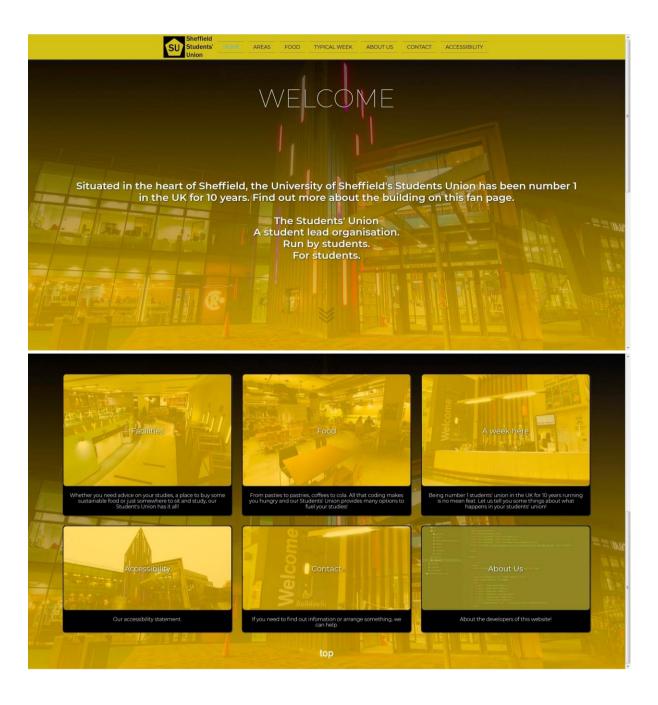


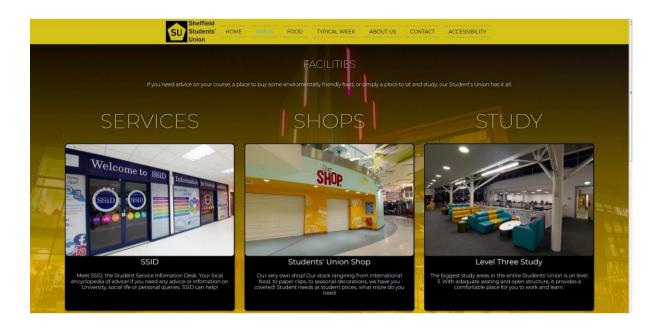




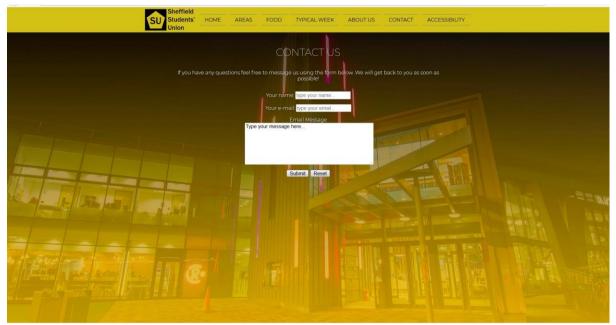
The tiles appear quite large on half sized screen for our thin desktop size. But when you had two rows side by side it felt cramped.

Large desktop:







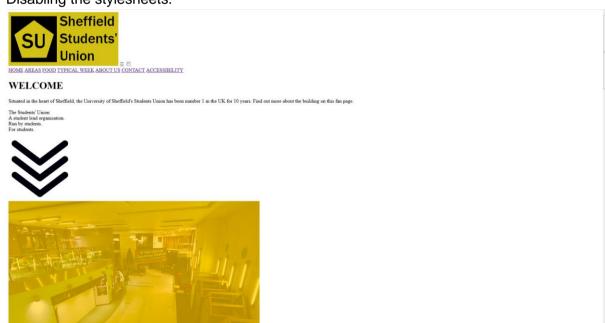


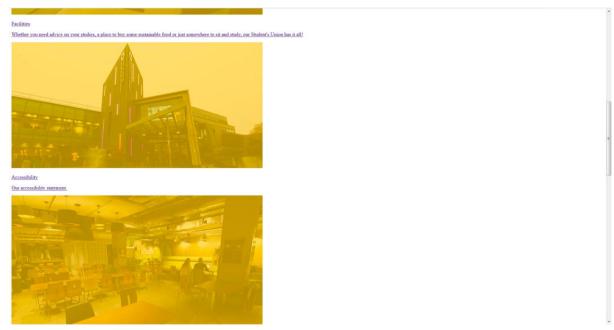
The layout was quite easy to translate from mobile to thin desktop and then to large desktop as our economic use of containers and flexboxes allowed us to simply change the flex-direction of our containers to create the desktop version. When we were planning our design we agreed on making our layout with flexbox so it would be easy to rearrange a page into different sizes.

These screenshots are all from Firefox, but the website looks the same on Chrome and Edge as we used a normalize style sheet.

One issue testing showed is that our website has no footer, but this is an easy fix.

Disabling the stylesheets:





While it does not look that pleasing, it still maintains its logical structure and reads from top to bottom.