Daniel Wildman  
**Design/Development Process Evaluation**

Coursework I, Task III

horizontal line

# Screen Shot 2015-12-14 at 11.28.19.png

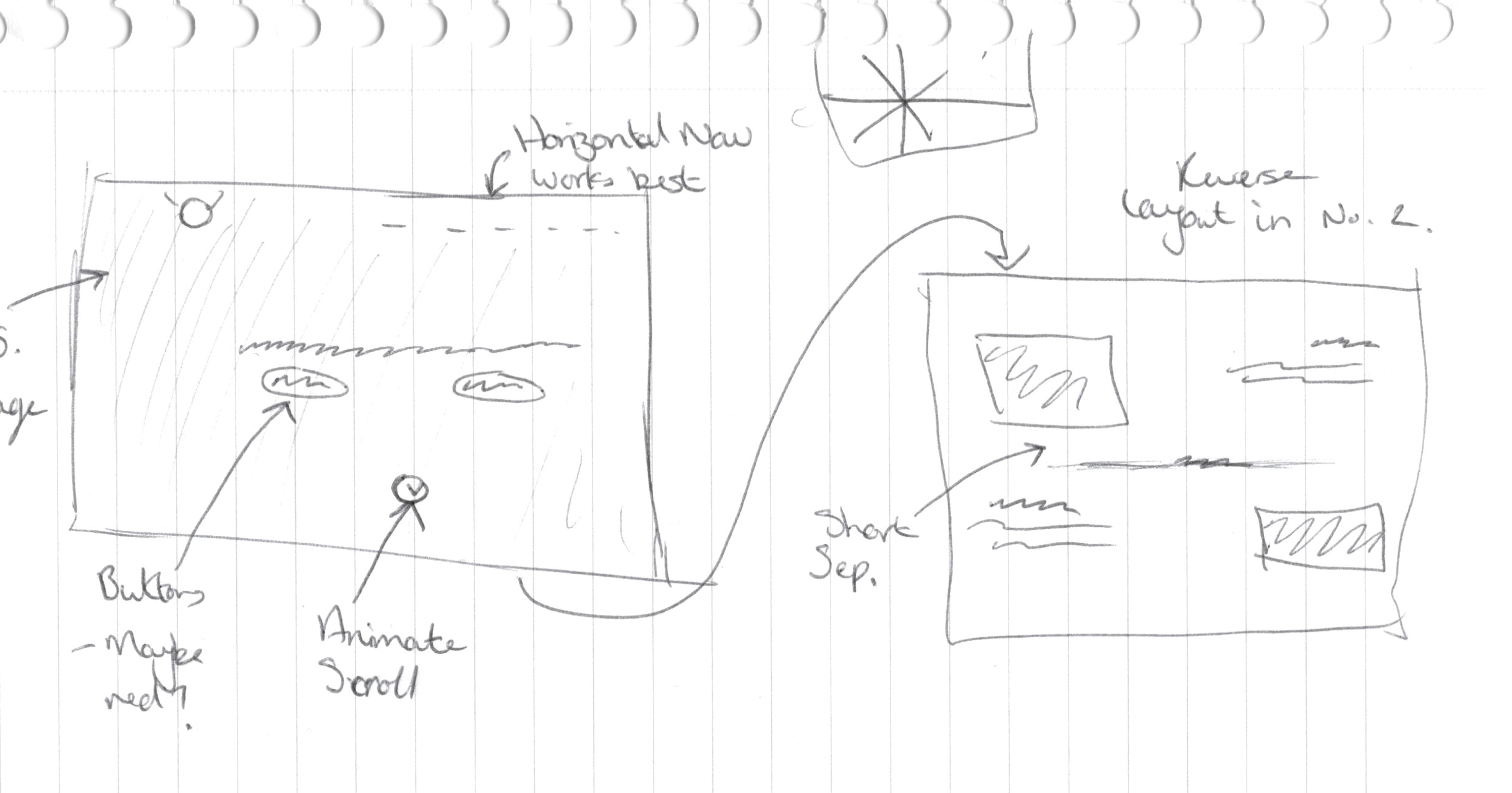
# Overview

The product I have created is a single page website, designed as a replacement for the current DemonFM website, that incorporates elements such a parallax, CSS Animations, jQuery and JavaScript, third-party code, and responsive design with assistance from the Bootstrap framework. My aim was to create a clean and modern interface that would be useable across platforms, however maintaining an air of professionalism across all views.

## Design Process

To generate ideas for this site, I broke the design into two distinct parts - Low-Fidelity Wireframing, and High-Fidelity Wireframing. This means that I could start with something very simple and didn’t have the pressure of staring at a blank document without any idea of what I was going to create.

### Low-Fidelity Wireframe

This first stage of the design process was intended so that I could quickly and easily come up with many variations of designs, and choose from my favourite ones. Using a pen and paper (no digital design at all), I sketched out very quick mockups of where key components of the UI would be, and chose the design that I thought would look best to take through to a high-fidelity wireframe.

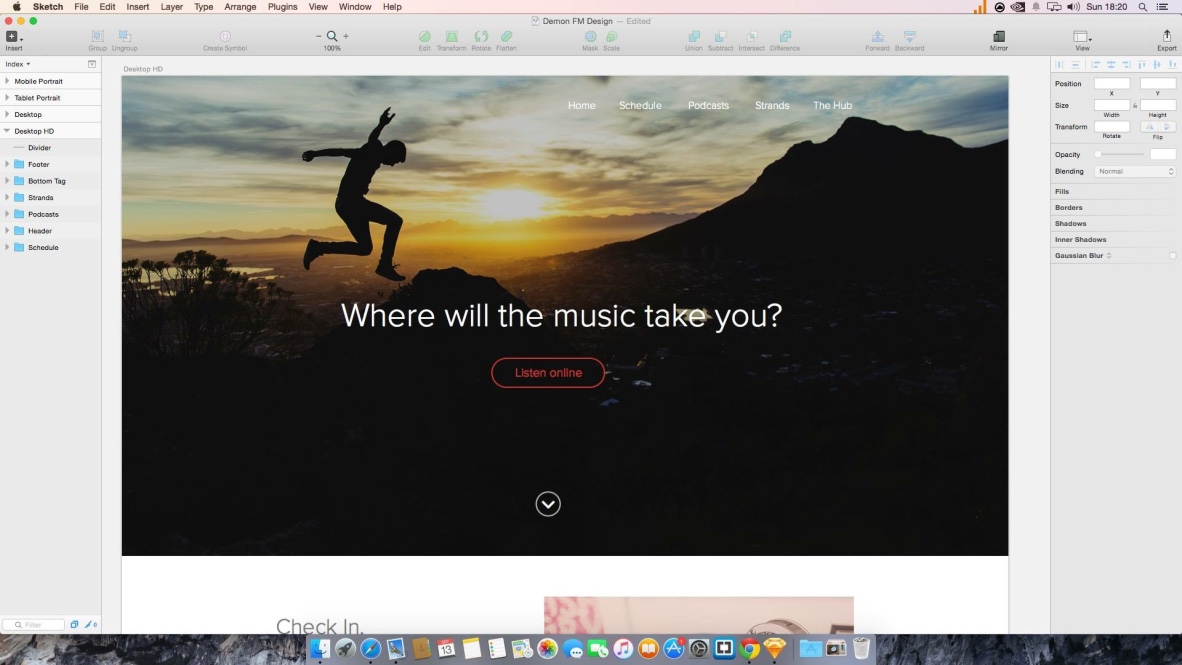
I believe that this section was completed very well, as I quickly had 7 different website designs in front of me, which was a wide variety to choose from and I believe that it enabled me to assess how different elements would fit my aim. Examples of such would be that I sketched out designs with both vertical and horizontal navigation bars, and I decided that a horizontal navigation bar fit the style of the page much better.

However, I only created a wireframe for the desktop view. This is definitely something that I would change if I had the chance to do it again, because it made developing the website considerably harder when I had no designs to work off, so I had to create a page in the development stage by guesswork.

### High-Fidelity Wireframethumb_IMG_3465_1024.jpg

This is the second stage of the design process, and involved recreating my favourite design from the Low-Fidelity section in a design app of choice. Given the choice between Adobe Photoshop and Sketch, I elected to use Sketch as it runs much faster than Photoshop and is designed solely for UI design, so I was able to recreate my design much faster. My end-goal for this stage of design was to have a single completed design that would reflect exactly how the website would look. In other words, if only the design were to be visible, it would need to be indistinguishable from the final produced website.

I believe that this process went very well. Given that I had a sketch that had been scanned into my computer to create the wireframe from, I already knew what had to go where, so I could focus my efforts on the aesthetic side of the elements, as opposed to being concerned about the positioning and structure as well. This allowed for the high-fidelity design process to be completed within a matter of hours.

However, once again, I didn’t create a design for the mobile pages, which causes issues for me during development of the site. In addition to this, I struggled to understand some elements of the design and had to keep referring to my sketchbook in order to make sense of it. If this became a greater issue, I would have considered utilising Mid-Fidelity Wireframes as well, and introducing this as a third stage. This would simply involve taking the Low-Fidelity wireframe and reproducing it (mainly the rough structure) in Adobe Illustrator. As these would be have been vector images, I would have been able to exploit the ability to zoom and look in extreme detail whilst maintaining clarity.

## Development Process

### So by the start of this point, I had a document on my computer that contained a Sketch replica of the product that I wanted to create. However, although design-wise it was perfect, it had no functionality. The aim of this stage was to turn that design into a responsive website, through the use of HTML, CSS and JavaScript.

### WorkflowScreen Shot 2015-12-14 at 12.56.19.png

### In order to be efficient, I knew that I had to have some sort of workflow set up - A way of working that would make my life as a developer easier. Therefore, the first thing I did was get the workflow sorted. The first part of my workflow was an app called CodeKit. This handled production processes such as compiling SASS/SCSS files, minifying JS/CSS files, and created a local server that allowed testing from multiple devices without having to use a hosting service. Another key part of my workflow that integrated quite nicely with CodeKit was to use SASS instead of standard CSS. SASS is a CSS preprocessor, and allows the use of coding techniques such as variables, mixins and nesting. Nesting was particularly useful for me, as I come from a background in C-based programming languages and therefore nesting is a technique that I am rather used to.

### Another key part of the workflow was the folder structure. Naturally, when a website has a lot of dependencies, if the dependencies are not logically structured then it can get a bit messy. In order to keep everything clean and simple, I created a logical folder structure to house all the dependencies. This means that CSS files were kept separate from JS files, or that HTML files were kept away from Bower-based dependencies. The root directory was kept to only the index file, and the CodeKit config file (the latter would be ignored in a git commit or an FTP upload).Screen Shot 2015-12-14 at 12.57.49.png

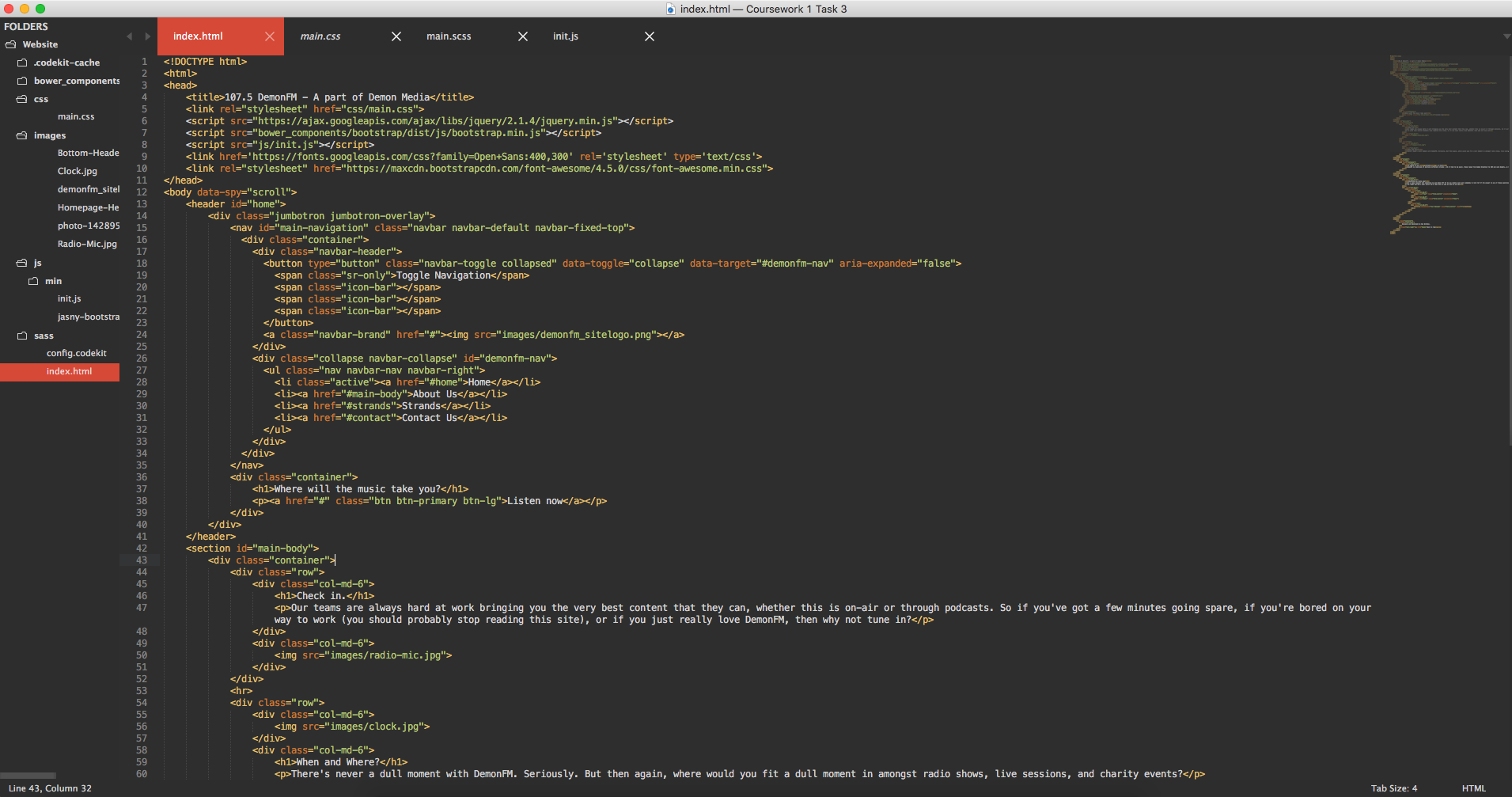
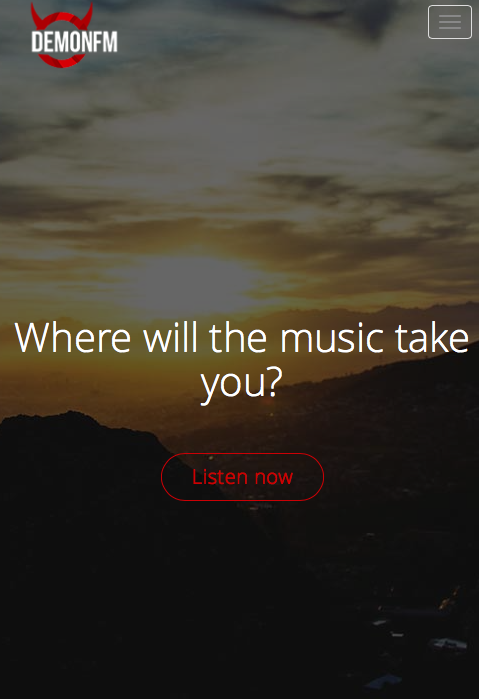
The final key part of the workflow is a system called Bower. This relies on Node.js, and is a dependency management package that allows large and popular dependencies to be pulled and stored correctly. I used Bower to install Normalize.css and the Bootstrap framework.

I was very happy with the way my workflow turned out, and the few minutes that were taken to get it set up were definitely worth the hours worth of development time that I had saved. The folder structure allowed quick and efficient storage of files without getting the directory messy, the use of SASS meant that I could write stylesheets in half the time it would have taken to write normal CSS, and CodeKit provided the backbone for all of these services to work. I believe that it helped me massively with development, and would be of great assistance if I ever decided to bring this site to production.

However there were a few changes that I would make if I was to do this project again. The first would be to remove Normalize.css from the Bower dependencies. The idea behind installing it was to assist with browser cross-compatibility, removing all default browser styles. However, upon inspection, I discovered that Bootstrap contained Normalize.css as part of its core, which meant that I was calling it twice. This was horrible inefficient and could slow page loading on slower connections. Other improvements that could be made are small quality of life improvements, such as using Live Reload or switching CodeKit for something more advanced like Gulp or Grunt, that allows me to configure the settings and modules via a JSON document.

### Main Development

### Once the workflow for my development was ready, I needed to do the difficult job - actually develop the website. However I made this much easier for myself given that the workflow was processing everything I did the second I pressed save, and I had the design and website open in front of me so I spend minimal time and effort clicking between programs.

I wrote the HTML files using Sublime Text 3, with a plugin called Emmett. This allowed for very fast writing of HTML, as simple markup such as “ul>ii\*4” would immediately render out to four li tags wrapped in a ul tag. The SCSS files immediately compiled into CSS thanks to CodeKit, so I didn’t have to worry about manually starting a compiler. This meant that I could make very fast small adjustments on the page, such as changing the padding on an element, and the second I hit the save command I could watch the change occur. This was incredibly handy for fine-tuning.

To make the responsive design aspect a little more professional, I used the Bootstrap framework. I overwrote a lot of the default Bootstrap styles with my SCSS files, however the breakpoints that come with it, and the way the grid system is designed to flow between layouts, was invaluable whilst making the website work. There was still additional work that had to be done though, such as having to create manual media queries that would adjust the font size of the header based on the width of the viewport.

The navigation bar is a great example of how I used CSS animations and JavaScript in my project. Using jQuery’s “.Scroll()” and “.scrollTop()” functions, the browser will detect when the page has been scrolled and will store the new scroll position in a variable. This variable is then compared with a 200px breakpoint and if the position is over the breakpoint, then the black navigation bar class will be applied. However, this has a transition property in the CSS files, which means that it will animate in when the class is applied. A similar piece of code will run to detect when the scroll position is less than 200px, and will subsequently remove the class applying the black background.

I’m generally quite happy with how this process went, however there are still things I would like to change. As mentioned earlier, I didn’t create a mobile design. This made it very difficult for me to create the mobile version of the website, and changed the way I wanted to work. The approach I would like to take next time would be a mobile-first approach, in which I create the mobile version of the site before anything else, and then work up from there to create tablet, laptop and desktop versions of the site. This wasn’t possible to do this time with the time constraints I had, given that I didn’t have a design to work off. Furthermore, I completed the creation of the website in a single pass. I made sure that I had completed all styles and interactivity of a section before I moved onto the next one. Next time I would prefer to do this in multiple passes - Create the HTML structure of the site, then create the styles for the site, then add any interactivity. This would allow me to concentrate on each separate area a little more, as they are vastly different my process this time somewhat hindered that.

On the contrary, there were many things that I was proud of during the development of the site. The use of the Emmett plugin saved an incredible amount of time and made creating the markup for the HTML files so much easier. In addition to this, the use of Bootstrap not only made development considerably faster and more simple, but allowed for a more professional approach to responsive design and cross-browser compatibility. Furthermore, the jQuery file was clean and efficient. Different functions are clearly separated, and code is easy to distinguish. SCSS files are laid out efficiently and simply, and it’s clear just from reading the file that I much prefer writing in SASS than to standard CSS.

## Overall Evaluation

Overall, believe that I have achieved my goal in making a single-page website that is clean, modern and sleek, whilst maintaining an air of professionalism. The use of Google Fonts give it a very clean look, whilst providing continuity throughout the page. I very much like the use of fixed backgrounds as well, it fits nicely with the theme of the website and again helps to assist with the modern look and feel of the website. The horizontal separator in the middle of the two sections of the main body could do with additional margin on the top and bottom, as could the form at the bottom of the page.

In addition to this, the ScrollSpy on the navigation bar doesn’t work quite as intended due to the compact layout of the page. Other small improvements include that the “Listen Now” button at the top of the page could animate, either to add an arrow or to change the element style.

### User Feedback

### Once the web page had been completed, I got a few opinions to gauge what people though on different devices, and what they think could have been improved.

A suggested improvement was to fix a margin error on the right hand side of the screen. There is an element that is pushing into the right hand side of the page and causing a scroll bar to appear. I used “Inspect Element” in Google Chrome to see what was causing the fault, however the body appears to be in the correct position and not extend. If the site was to ever go to production, this would absolutely need to be solved.

Another suggested improvement was in agreement with my earlier comment about the horizontal separator needing to have more of a top and bottom margin. The general opinion was the the sections appeared to be too close together and clashed a bit. I was given the opinion that even if they were just a bit more spaced out, it would make the site look a lot nicer.

In addition to this, I was given the opinion that the form does not look to great on smaller phones. The padding on the elements themselves were not applying properly, however this issue was fixed once I had finished collecting feedback.

In contrast to this, I had numerous people tell me that they loved the colour scheme and design of the website, and when asked if they could describe it in a single word came out with responses such as “Modern”, “Slick” and “Neat”.

In conclusion, both the processes of creating the side and the product itself are not without their issues, however I believe that I have successfully achieved my goal of creating a single-page website that is clean, modern and professional that work well on a multitude of devices and browsers.