Web Technologies: Client Side Report

James Collerton

March 15, 2015

Initial Warning About Chrome Console Errors

Within my submission I have included a Google Maps API. This API generates errors within the console, there are only four and as are displayed below:

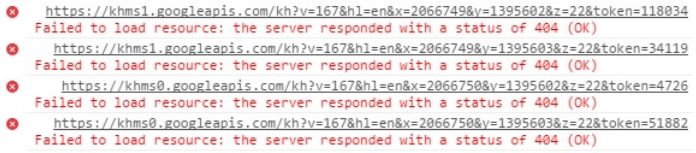

Figure 1: Errors generated from Chrome.

If you comment out that section on every page then they run totally clean. I am fairly sure that this is due to the Maps API not responding well to being hosted on the localhost. It actually took me a lot of time to get the Maps API working correctly and so I went and asked whether or not I should leave it out. The response was to leave it in but to mention it in the report.

```
<iframe title = "embedded_google_map"
src="https://www.google.com/maps/d/embed?mid=zPlYNMRlsqPw.kRRmejvbQabQ&amp;z=15&amp;ll=51.459847, -2.609405"
width="100%" height="480"/>
```

Figure 2: The offending code.

Forms Do Not Work

None of the forms on the site work at the moment! This is as they are all intended to be used with the back end of my project. I have hidden the submit buttons for the time being.

Overview

Before outlining what was done in my project I would like to mention a couple of key factors regarding my submission.

- My aim for the client side part of the project was to create a polished, good looking website with a great user experience.
- My secondary aim was to learn about as many different areas as seemed reasonable. Therefore I have not concentrated my efforts on one particular discipline (e.g. HTML, CSS or JavaScript) but have tried to cover these and then to branch out and learn new things, like different libraries and frameworks. This is because this attitude fits with my aim of creating a polished site, and I also thought that it more realistically reflects the skills you would need in industry.

  Therefore the area that I would like to have considered having examined in depth is design and different frameworks. These have definitely been my two main areas of focus and research.
- ***All*** of the images, icons and graphics on the site are my own and created from scratch, including the photography.
- The site is designed to work on the latest releases of Chrome, Firefox and Safari, but does not operate fully on Internet Explorer. It was mainly designed on Chrome and so all features appear there, with more limited availability on Safari (covered later in the report).
- Most pages should collapse into a mobile site once they reach certain dimensions, and those that don't will become scrollable at a small enough size, demonstrating fluid design.

The reason I mention these here is because I spent a lot of time and effort on areas that perhaps are not directly tangible in the code. For example the images, layout and feel of the website I took great pains to try and make like a 'real site'. Hopefully having these factors in mind as you go through the marking will help.

The rest of this report will break down into the sections *HTML, CSS, JavaScript, GIMP, InkScape* and *Originality and Creativity*. The first five sections will be a very brief overview of what has been put into the website in order to prove I can use the technologies as required by the course. The last section will concentrate in more detail on the work I put into researching different technologies and achieving my ultimate goal of creating a good looking, polished website front end.

HTML

- All pages are fully formatted in Polyglot HTML5 and have been validated using Total Validator with no errors or warnings except those mentioned at the end of this section.
- I have used a variety of different HTML5 commands including classes, IDs, links, forms, lists, divs, spans, iframes, script links, canvases, img, SVGs and objects to name just a few.
- I have dynamically generated HTML5 using outside libraries and my own JavaScript.

- I have used conditional IE comments in order to include the *html5shiv.js* and *respond.js* polyfills in an attempt to format the site for IE. However, despite my best efforts the site does not function well on Internet Explorer.
- I spent a lot of time learning and writing the necessary HTML in order to use the *Bootstrap* CSS and JavaScript framework (I will cover this in the later section on creativity).

Total Validator Results.

In order to make sure that my pages functioned to the necessary standards I validated each of them using Total Validator on the HTML5 Polyglot setting. All pages came out clean except for:

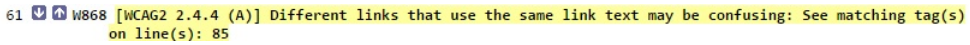

Figure 3: First warning type.

This is due to two links having the same text within them. However as they link to the same page I figured that this was an acceptable warning to allow. The reason that this sometimes occurred is for design purposes.

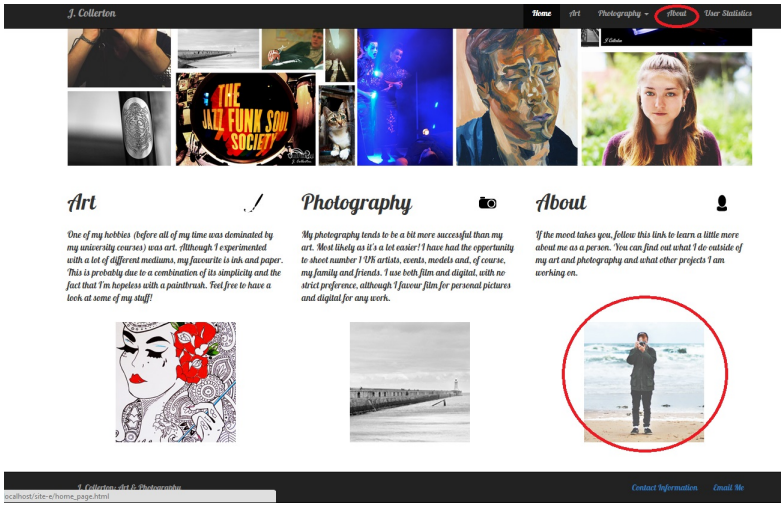

Figure 4: Circled are two buttons with the same purpose but styled differently.

The only error I recieve is described below, but it is self-contradictory. It says that the value must be an integer over 0, like height in the example below, and must be relative. As you can see, the width is relative and still generates an error, therefore it seems impossible to satisfy the validator.

I was unsure of this and so checked it with Ian, he said to leave it as it was.

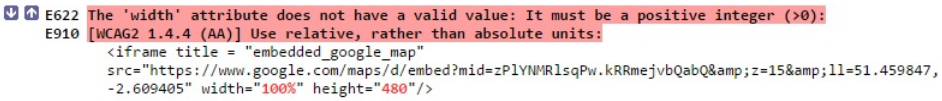

Figure 5: Example of contradicting warnings.

CSS

There are lots of minor formatting tricks I did in order to create the layout seen on the site, the first of which was to use the normalize CSS sheet on every page to ensure compatibility across browsers.

However, most of this you can look over in the submission. There were three quite major CSS effects that I invested a lot of time into and I will note them here.

- **Fading (see *Models* page):** I had seen other sites that used fading to transition between pages. However, a lot of the time this was slow and hampered the user experience. In spite of this I did like the general idea and so looked into emulating it for myself.

  Initially all of my searches came up with JavaScript solutions. As I was attempting this effect early on in the course and we had not started the JavaScript lectures yet I avoided this and looked for CSS solutions. Eventually I found some web tutorials and using them and the documentation for the *webkit-animation* features put together the staged fading effect that you see in the code.

- **Modals (see *Email Me* link in bottom navbar):** Although I have used the Bootstrap modal, I also created my own one from scratch using JavaScript and CSS. The CSS for the modals consists of an overlay and then formatting the inner modal box. The visibility of these are then swapped using JavaScript.

  Between the overlay and the actual modal box a lot of work was put in to make them fit in with the appearance of the rest of the website and to position them correctly and with fluid design.

- **Hovering and Text Appearing (see bottom row of photos on *Main* page):** I came up with this idea and the CSS from scratch. Although the CSS is quite short, getting it to work as I needed it to was conceptually challenging, especially as this was one of the first things I attempted.

  I started by looking at some tutorials for making image galleries, however none of them had the right effect I was looking for. I could easily change the opacity of the photos on hovering, so the problem was to then overlay text on top of the image.

  After a lot of Google searching I came across the idea of putting the image and the text within one div and then using that to control the behaviour of both simultaneously. I then took that idea and spent a lot of time playing with positioning and centering to make sure the text appeared in the center of the design and that it synchronised with the opacity change.

The above combined with the other CSS I have used should demonstrate CSS being used effectively.

JavaScript

I set out on three major JavaScript projects, the *Email Me* modal, the bouncing ball and the opening panorama.

- **Bouncing Ball (see the *About* page):** As I was looking through JavaScript materials I came across the idea of animating on a canvas. The actual code for making a ball appear and bounce up and down is quite simple physics. Adding the click functionality so the ball bounced when clicked took a little while, but the most complicated part was making sure that the ball animated smoothly between being clicked and starting to bounce.

  Another thing that took a lot of effort was to have the 'About' text fade over the ball, then having it dissapear so the ball could bounce properly once it had been clicked. Although this only manifests itself in one line of code, learning about removing CSS classes using JavaScript took some work and a lot of time to implement effectively.

  All of the above research was implemented using a mixture of the W3 documentation on functions, Stack Overflow threads and Google searches of JavaScript.

- **Modals (see *Email Me* link in bottom navbar):** Originally I had used the Bootstrap modal as seen on the *Contact Me* button on the bottom navbar. Rather than rely totally on Bootstrap I decided to do my own version.

  This design was mine and created entirely from scratch. I first use CSS to make an overlay and format an email form whilst keeping them hidden. Then I add JavaScript to toggle them between being hidden or visible when the user clicks certain buttons.

  The implementation of this took quite a lot of fiddling. I had a lot of difficulty applying the listener functions to the webkit-animations. If I used two of them, one for the starting animation and one for the finishing animation, they tended to detect the starting animation as the ending one and close the modal straight away. Eventually I used a timer to get round this issue.

  Another thing was getting the JavaScript to recognise the right animation to detect for. I spent a very long time researching on Google why the browser would not pick up certain animations and then finding out how to fix this. It is not a particularly easy thing to solve, but I persevered and in the end wrote the short function for detecting the correct animation so that the site worked on Chrome, Safari and FireFox.

  Please note: None of the forms have yet been validated using JavaScript. That is because it seemed to make more sense to do that once I had started my back end of the project. In my back end I will set up an automatic email service using the email form, and I will also create a user statistics database to feed into the D3 visualisation. Therefore neither of those forms works at the moment and are there for cosmetic effect.

- **Opening Panorama (see *Index* page):** The main challenge of this was getting the image to clip to the right size properly. I am quite proud of the overall design of this and think it demonstrates an effective use of JavaScript.

  In order to achieve this effect I did a lot of research into different JavaScript ideas and how JavaScript deals with drawing images. One of these ideas was to have an image panning round. I took the idea but no code and wrote it myself in order to achieve the effect witnessed. In order to write the code effectively it took research into the JavaScript `drawImage` function and how clipping and cropping worked, using a mixture of documentation from W3 and Google.

I also used JavaScript in linking in Bootstrap and with the D3 data visualisation library. These will be covered more extensively in the later section on creativity, the above were just to show that I knew how JavaScript worked on its own.

InkScape

My main motivation for using InkScape was to create icons for my page as this seemed like the most appropriate use of the software. I sketched them each by hand and then used the mouse to draw them. I have used a graphics pad in the project, but that will be covered in the GIMP section.

Figure 6: Selection of my InkScape drawn icons.

One technique I used fairly heavily was using the rectangle tool and then altering the radius of the edges to round the corners, building up images bit by bit using these components. Icons are designed to be small and so it made no sense to do anything overly intricate as it would not be seen on a

smaller scale.

I will quickly walk through how I made the photography icon to show my approach to InkScape:

Figure 7: Steps in creating an icon.

We start with a rectangle, then round the corners using the edit paths by node functionality. We copy this above the original rectangle and add circles for the trigger and lens respectively. Other tools that have been used include the calligraphic tools for pen lines and the polygon tool.

I do very much appreciate that I have not neccessarily done anything 'flashy' with InkScape. However I have created a variety of good looking icons, which was much more useful for my site than unnecessarily creating complicated useless drawings.

GIMP/ PhotoShop

I have used GIMP on occasion, and I did open it up and have a go as part of the project. However I have used PhotoShop for a number of years and so I fell back on that as my photo editing tool of choice. All images are still in the required *.png* format.

As every single image on the website is my own, I have put a lot of time into this part of the project. It would not make much sense for me to go through how I edited each and every photo as there are so many, so I have picked one to use as an example, and will then give a quick overview of some others.

Most of the photos have had a similar amount of work done on them in order to look as they are in the site, and I really feel that this is an area I have put a lot of focus into.

Figure 8: Original Image.

Here we have the original image. This was taken on a Nikon D3200 with a Nikon 50mm 1.8 Lens and an SB600 speedlight. My personal preference is to use flash to wash out a lot of colour and then bring it back in post-processing.

Notice the skin tone is uneven, there are dark patches under the eyes and the colour is flat. The structure and focus are both fine, so we look to correct the prior points in PhotoShop.

To bring out the colours more we use a variety of techniques. Firstly, I used a curves plug-in and created a light 'S' shape. This creates a good contrast between darker and lighter areas without ruining the midtones. I then concentrate on the focal areas of the face. In this image I centered on the eyes and the lips. I used the magnetic lasso tool to select the areas and upped the red saturation on the lips to make them seem more full, as well as the brown saturation on the eyes. I also adjusted the contrast on the eyes to make them pop. This process probably takes around twenty minutes to get right.

In order to even skin tone we apply a layer above the background and use colour dodge and colour burn on the light and dark areas of the face to create a more smooth appearance. We also take colours from the more neutral areas and paint them with a low opacity over the red or blotchy areas of the skin. This process takes about forty five minutes in order to be done well.

Figure 9: First Touch Up in Colour.

So after evening the skin and adjusting the colours for about an hour, we have reached this point. The colours look a lot more lively and the skin looks a lot more even. However, I still don't really like the effect as the colours aren't that appealing. When this is the case, and the focus of the picture is more on light than colour, I tend to just swap to black and white. This does mean that a lot of the work done on sorting out the colours before has been negated, but I think that's just the way it goes sometimes.

Figure 10: Final Image (3 - 3.5 Hours).

At this point I realised that I perhaps had not shown off as many of my drawing skills as I could have, so I then decided to create a layer over the top of the image which would hold some vector style patterns.

I used some of my sketches to create tattoo designs for his neck and face and then overlaid them on the image using a Wacom Bamboo graphics pad. I tweaked the contrast and tones so that they matched the image well and it created a desirable overall effect. In total this probably took about another hour and a half or two hours.

I have spent a lot of time gathering up different tutorials on photography and photo editing and employed both these tutorials and my own experience in order to create all of the images on the site. I also sourced all the images myself from my own photography.

I could go through every single image on the site and discuss how long each took to make and how I made them, however I think that might be overstating the point. Instead I think that the most effective method of showing how much time and effort has gone into the images is to take a selection and show the original and finished photos and give a brief overview of the tools used to create the effect.

Figure 11: Tools: Curves, Levels, Gradient Maps, Saturation Control, Brushes, Eraser.

Figure 12: Tools: Curves, Levels, Gradient Maps, Saturation Control, Colour Dodge and Burn, Brushes and Erasers, Gradient Maps and Light Tools.

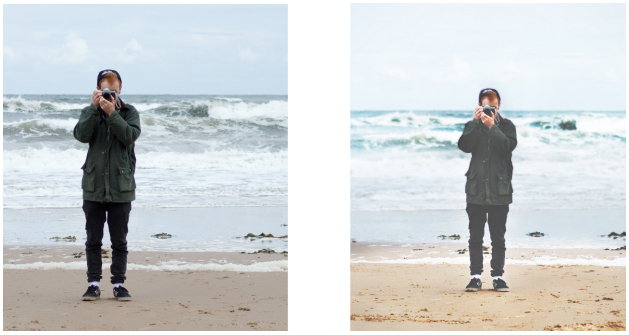

Figure 13: Tools: Curves, Levels, Gradient Maps, Saturation Control, Brushes and Erasers and Gradient Maps.

Originality/ Creativity

Like in the mark scheme, I used the prior 50% to show that I had the capacity to use the different tools required by the course. Now I want to focus on the area where I demonstrated a depth of learning, rather than a breadth. My area of focus was design, frameworks and libraries and how I could use the combination of the three to create a professional looking site. Within this aim I covered:

- Use of the Google Maps API.
- Use of the Bootstrap framework.
- Use of the D3 data visualisation library.
- Giving the site an icon.
- Design work (including Google fonts API) and cross browser compatibility.

Google Maps API

Although the code for this is only one line, getting it to work was very difficult and took a long time! Google has recently updated its maps service and the method it uses for embedding maps in websites. All tutorials are based around the old version of Google Maps and so it is very hard to get information on the newer release.

Getting *a* map to show on the site was actually relatively easy. By going onto Maps, and then following links to the embed option you get to this screen, which gave you the necessary HTML5 to put on the site:

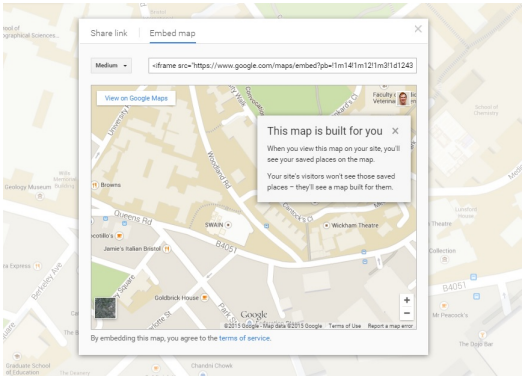

Figure 14: Map embedding screen.

Excellent, that's exactly what I want to put on my site! However on actually embedding the link the result you receive is this:

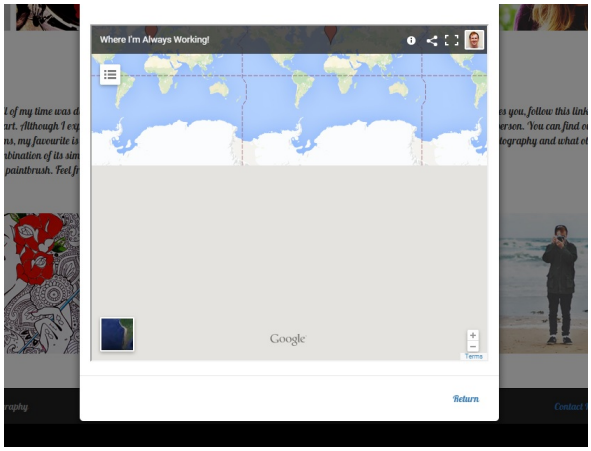

Figure 15: What actually appears on the screen using the Maps link.

So I set about trying to track down how to control the zoom. After a lot of trawling I found that you could attach zoom and location coordinates to the end of the URL in order to specify these parameters. On trying this it still didn't help and I had the same problem.

A few days later I noticed that whenever someone was displaying their map on a website, they had saved it to their own Google Maps account and were displaying it through there. I created my own map on my Google account, saved it, took the embed link, found the longitude and latitude of the place I was trying to display and attached them to the URL.

Finally it worked!

Use of the Bootstrap Framework

Originally I got halfway through making a site without Bootstrap doing all of my own HTML and CSS. However, I found that this was quite slow going and if a framework was available I might as well use it [1].

My tactic was then to use the CSS and HTML I had created in my first site in combination with the Bootstrap framework in order to demonstrate that I could use both HTML and CSS as well as learn how to properly use frameworks. I think that the skills I learnt from taking this approach and being able to put together a polished looking website from the combination of the three were more valuable to me personally than doing a smaller amount just using HTML and CSS.

In order to demonstrate that I know the framework well I have tried to incorporate a lot of the different features that it provides.

- ***Navbar:*** The top navbar is made in Bootstrap. I also incorporated drop downs on the photography section and it collapses to three lines similar to a mobile site on reaching a certain size. In order to achieve this effect I spent time looking over the Bootstrap documentation to familiarise myself with the necessary HTML and also looking through lots of YouTube tutorials.
- ***Responsive grid system:*** Many of the pages are formatted using the Bootstrap responsive grid system. This means that the site has a fully fluid layout as in the requirements. To fully understand and implement the grid system took research into how to use the grids and what parameters to use where, as well as how to nest grids. All of the above I took from the documentation.
- ***Modals:*** Found in the *Contact Me* modal and on the photography page. Modals are a great feature of Bootstrap and helped me add an extra dimension to my own site, as well as offering me inspiration for my own modal. I looked to a variety of sources in researching how best to get these to work including Google, the documentation on the website and YouTube tutorials.
- ***Dropdowns:*** The drop down sections on the *Models* page are adapted from Bootstrap. Originally bootstrap uses dropdowns with text rather than images, but I looked over the documentation and fitted the design to work with images instead.
- ***Popovers:*** Found on the *Art* page. These as well as the next two features, were composed using help from W3 and the Bootstrap documentation.
- ***Tabs:*** Found on the *Art* page.
- ***Carousels:*** Found on the *Photography* page .

Although I do not expect as much credit for these features as if I had created them from scratch myself, I would like to point out that creating the Email Modal in JavaScript and CSS did not take that much more time than doing the exact same thing in Bootstrap.

A lot of time and research went into these Bootstrap solutions and making sure that they worked effectively with the rest of the site.

[1] I have attached the original website as well in my submission as I did put a lot of time into it before I swapped.

Use of the D3 Data Visualisation Library

Something I really wanted to take away from the course was an ability to work well with data. Therefore I decided to learn how to use the D3 JavaScript library to create interactive data visualisations and put them on the web. These can be found on the *User Statistics* page.

My concept for the server end of the project is to collect data from the users of the site and then display it using D3. Hence the form on this page is only cosmetic and does not function yet, as it made more sense to add this in once the collection of data had been finished.

The JavaScript for the current D3 graph is contained in the *user_stats.js* file and was made using an amalgamation of sources including: tutorials from the book *Interactive Data Visualisations for the Web*, Stack Overflow threads, Google searches and references to the D3 documentation. On the surface it works perfectly by clicking the provided text and gives an impression of the final form of the graph once the necessary data has been fed in.

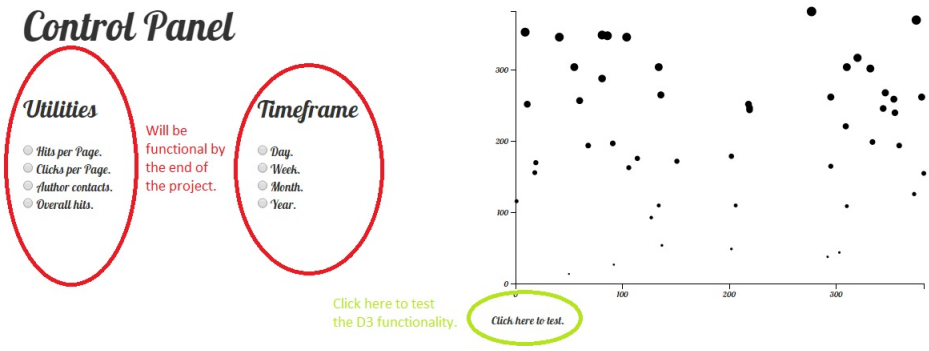

Figure 16: D3 in the testing stage before being implemented fully.

Although I have only demonstrated the one use of D3, I feel I have learnt the library well enough to expand the design I have and to use D3 for a variety of different data types. A lot of time went into learning D3, but it only made sense to include the one example on the site.

Giving the Site an Icon

Something I had noticed most 'proper' websites have is an icon that is displayed whenever you visit a page. I then began to track down how exactly you achieved this effect.

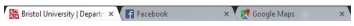

Figure 17: Examples of icons on sites.

After Google searching for how this was achieved I came across embedding the icon in the head of the page. However, it needed to be of a very specific size and type. In order to fulfill these requirements I spent some time investigating plug-ins for PhotoShop and GIMP. This proved unsuccessful. The

easiest method of getting the necessary format was to host the image online and then link the URL to the page.

Figure 18: Finished icon.

This was a remarkably tricky process in order to get the icon to display correctly. Initially I forgot to clear my caché and so it did not appear as I thought it would. I spent a lot of time looking for solutions to a problem that didn't exist because of this!

Design Work (including Google fonts API) and Cross Browser Compatibility.

As I mentioned before, a lot of work went into the design of the website and into making it look and feel like something you might concievably encounter on the net. Elements of this can be recognised in the work done on the images and on making the icon set look cohesive and professional in InkScape. However, I think most of the work done on the design should be evident from browsing the site as I hope it feels complete and polished.

I tried to make the design as interactive as possible in order to enhance the user experience. This can be seen by the use of tabs (*Art* page), dropdowns (*Models* page) and the JavaScript bouncing ball animation (*About* page) that I created from scratch.

I also spent time making the site work on the three browsers specified at the start of the report. The only points worth mentioning surrounding this are concerning Safari.

Safari handles translation slightly differently to Chrome and Firefox and so some elements are not displayed the same. In addition to this, SVGs are not rendered by the same rules and so do not appear on Safari. However, these are both minor cosmetic differences and so I did not think they were important. The only time they makes a pronounced effect is on the User Statistics page where the graph is cut off the screen.

Conclusion

I would just like to conclude by summarising the points of the website of which I am most proud and were researched with the greatest depth.

- **Design:** The site looks and feels complete and polished. It is well populated and looks professional. I feel I have made a lot of effort to make it appear cohesive from page to page and to function well with an interesting user experience. Furthermore there is good attention to detail demonstrated in doing things like changing the site icon and spending time getting the Google Maps API working correctly.
- **Breadth of Learning:** Going back to my original aim of having a polished, good looking website, I have spent a huge amount of time researching and learning about a lot of different areas and in considerable depth. In total I feel I can take away an understanding of all

of HTML, CSS, JavaScript, InkScape, PhotoShop, Google APIs, Bootstrap and D3 Data Visualisation.

- **Effective Use of Tools:** I feel that everything I have created using the above has added value to the site and contributed to its overall feel and appearance.

I genuinely hope you enjoy using my website.