



PERSONAL WEBSITE REPORT

Jonathan Farrell
C08661855



APRIL 27, 2017
CMPU4063-A

Contents

1. Introduction:	2
2. A tour of my website:.....	2
2.1. Index/homepage:.....	2
2.2. Employment History and Education:	3
2.3. Certificates:	3
2.4. Hobbies:	3
2.5. Contact:.....	3
3. Website Features	4
4. Performance	6
5. Responsiveness	8
5.1. Methods to ensure responsiveness	8
5.2. Testing Responsiveness	9
6. Validation	9
6.1. HTML Validation.....	9
6.2. CSS Validation	10
7. Other Testing Methods	10
8. Final Thoughts on My Webpage	10

1. Introduction:

The website I have developed for this assignment is a personal website. The purpose of the website is to act as an online C.V. for potential employers. I designed the website in the previous phase of this project using wireframing and paper prototypes. In this document, I will give a tour of the website, discuss the testing and evaluation methods used and point out places where the final draft of the website deviates from my phase 1 design.

2. A tour of my website:

2.1. Index/homepage:

The homepage of my website is split into two columns. The sizes of these columns in relation to one another makes use of the golden ratio. The left column contains the navigation bar at the top, and my own bio or cover page underneath this. I have not filled in genuine information about myself here, rather inserted some text in Latin, simply acting as a place holder to show how text will look here. The right column shows the time, date and time zone at the top. Initially I had the 'Career Ladder' header (seen below) in this div, however later in the design process I was trying to work out some javascript I could put in this website that wouldn't seem out of place. I decided on this clock and found this to be the most fitting place to put it. Underneath this is the Career Ladder.

The Career Ladder is an image I created myself using GIMP (an open source alternative to photoshop). Having created the image, I then mapped the coordinates of the text, to create links out of each line. I set the hover selector for these links, in CSS, to show a border around the text. I found this more aesthetically pleasing than a colour change. The decision to map the links to, and therefore have the borders cover, 100% of the div was both aesthetic and practical, using the entire width of the div ensures responsiveness. You can change the browser size to make the ladder any width and these links will remain in the same place, mapped over the entire text area.

The Career Ladder links are anchored to specific IDs within the Employment History and Education pages. This means that clicking on, for instance, St. Gerards School, doesn't just bring the user to the Education page, but to the specific part of the page that they are looking for.

The page, like all the pages on this website, is fully responsive. The skeleton framework was used and a twelve-column grid. All the images are flexible, achieved by setting a 'max-width' to a percentage value in CSS and a 'height' value of 'auto'. Percentage values are used for image sizes, rather than pixel values, this too is to contribute to responsiveness. The picture of myself has a circular border, this is an example of CSS3.

There is a major difference in this page when compared to my prototype for it. In the prototype the text and image of myself were positioned side by side. In the final webpage, they are positioned one on top of the other. I made this change because when positioned side by side, there simply wasn't enough room for a satisfactory amount of text. The prototype design looked good on paper, but the webpage works much better the way it is structured in the final draft.

2.2. Employment History and Education:

The second and third tabs from the left lead the user to the Employment History and Education pages respectively. These pages have the same structure so I will discuss them together rather than one at a time. Both pages can also be accessed via the links on the Homepage's Career Ladder. Like the homepage these pages make use of a twelve-column grid. Unlike the Homepage, these two, and all subsequent pages, have only one column and a navigation bar that stretches across all twelve grid columns.

The navigation bar is responsive, in that when the browser window is reduced in size, the navigation buttons will start to stack on top of one another, eventually forming a block rather than a line. Had I have had more time to work on this website, I would have liked to implement a sandwich/hamburger button that would have appeared once the browser window got below a certain width. Unfortunately, this wasn't to be but the currency solution is at least better than forcing the user to scroll horizontally when using a small screen.

Beneath the navigation bar is the header for the page and beneath this the content starts. I chose to use an eleven-column grid for the main content of the page, because I prefer the way it breaks up the page aesthetically. Each section contains an image and these images are responsive. This is achieved by using percentage values for sizing, setting 'max-width' values and setting the 'height' value to 'auto'. When the browser window is reduced in size, the images shrink, and eventually the page structure will change to stack elements vertically.

In my phase 1 paper prototype for the Employment History page I had the images displayed above the text, rather than the text wrapping around them. I had a mental model in mind for this, one of a paper CV. I thought that if I arranged everything vertically, I could make the section look like a standard paper CV, which would appeal to hiring departments. In the end this didn't work out as I expected as it required the images to be so large that they filled the entire screen. This meant that the user had to scroll down to read any of the text and this I felt was bad design. I am much happier with the current design, despite the fact that it abandons the mental model I had in mind.

2.3. Certificates:

The certificates page I unfortunately did not have time to properly utilize. I did however, in the end, find some extra time and implemented a flexible iframe here. The frame links to the President's Award website, to keep all elements of the website relevant. The iframe was made flexible similarly to the images in previous pages.

2.4. Hobbies:

This page, due to time constraints, has been left empty for now, though it has been styled in the same way as the rest of the website.

2.5. Contact:

The contact page abandons the bordered, box model utilized in previous pages, in favour of a more open, spacious design. The contact details are contained within a table to organise them neatly. I chose red for the default colour for the LinkedIn hyperlink, because it is consistent with the colour scheme of the website. The text field has an input type of 'email', this means that the field is self-validating and will warn the user against entering email addresses that lack an '@' symbol or placing a dot in the wrong place. Once a valid email address has been entered an alert box will pop up thanking the user for their email address and stating that I will send them an up to date CV asap. I used javascript to do this, the function is defined in an external `myscripts.js` file.

3. Website Features

The purpose of this section is to address some questions specifically asked in the assignment handout.

1. Number of lines of my own CSS code:

My own CSS is contained in the folder 'mystyles.css', I have a compressed version which is the one being used by the webpage, as well as an uncompressed version I have included for your reference. The number of lines in the uncompressed file is 242.

2. Number of lines of adjusted CSS code:

I used the skeleton framework for this website. Once the webpage was completed I went through the 'skeleton.css' file and removed 182 lines of unused CSS code. Beyond removing these lines, I made very few changes to this file, opting instead to use my own 'mystyles.css' file for any custom styles I wanted. An exception to this can be seen on line 67 where I reduced the margin for 'h5'.

3. CSS framework:

(i) What is it?

I used the skeleton framework.

(ii) How can it be accessed?

It can be downloaded for free at <http://getskeleton.com/>, in my files it can be found in the 'css' folder. Both the skeleton.css and normalize.css files are part of this framework. It also provides a sample index file which can be used for html. My website uses a compressed version of these CSS files, I have included a non-compressed version for your reference. A great deal of unused CSS code has been removed to improve performance.

(iii) Why did you use it?

I used the skeleton framework because it is an effective way to ensure responsiveness. It is light, easy to use, and included some nice additional styling features such as table and button designs. The crucial element for me however, is the grid, which allows for fluid design.

4. Use of HTML5 tags:

Some examples of HTML5 tags that I used for this website are <header>, <main> and <nav>. These elements are very useful for defining the pages' structures.

5. Use of CSS3:

Some examples of CSS3 properties used on this website are box-shadow and border-radius. Box shadow can be seen on every page, giving the shadow around the different sections. Border-radius can be seen on the home page, it gives the photograph of myself its round border.

6. Use of iframes:

The Certificates page contains an iframe which links to the President's Award website. The iframe is flexible, meaning it responds to the size of the browser window being used to view it. This was achieved by setting the width to a percentage value and setting a maximum width value of 100% or less, in this case 80%. Unfortunately, this element massively decreases the websites performance. This is a source of frustration for me because throughout the development process I had made sure to design the website in such a way that it would run as quickly and smoothly as possible. It was a difficult decision to make, whether or not to prioritize performance and sacrifice the marks I will hopefully receive for implementing a flexible iframe.

7. Use of forms:

The Contacts page contains a form which is composed of an email input, a submit button and a reset button. This form is simple but effective and makes use of a custom made javascript function which brings up an alert box to inform the user that their email address has been received. Since the website is not hosted, and the form is not actually functional, meaning there is no real mailing list that the website sends inputted email addresses to, I initially had a problem. Whenever the 'submit' button was clicked, a new page would be opened, which would contain an error message, as if the user clicked on a dead link. I managed to fix this problem by setting the 'action' to '#' within the form tag. Now the webpage behaves as if it is supplemented with server side technology, sending the provided email address to a .php or .asp mailing list.

Another noteworthy hiccup in the development of the form was, when I developed the javascript function, I initially called the function within the <input> tag for the submit button with the 'onclick' property. This worked in the same way as it works now however, it overrode the <input type = 'email'>'s build in validation functionality, so the form no longer self -validated, warning the user against inputting erroneous email addresses. This was solved by moving the function call into the <form> tag and calling it with 'onsubmit' rather than 'onclick'. Now the email input self -validates, and presents the alert box when the submit button is clicked.

8. Use of javascript:

Javascript has been used twice in this website. The date and time displayed in the top right corner of the Index/Homepage uses a date object which is a pre-defined javascript object. The only customization needed here was to style the output to fit with the design of the website using CSS. The second example of javascript is in the contact page. This has been discussed above in the section on forms. The function is one I wrote myself which displays an alert box in response to an email address being entered into the email field. This function is defined in a separate javascript file, myscripts.js. Both of my uses of javascript are simple but they work well and fit with the design and theme of the website.

4. Performance

Due to time constraints, I was unable to host my website. This meant I had limited options when it came to testing performance. Most websites that offer this service, including www.webtagetest.org and <http://gtmetrix.com> do not allow for file uploads therefore I was unable to use them. I was however able to use Chrome Developer Tools and the Firefox addon, Extended Statusbar.

Chrome Developer Tools:

On first check my webpages loaded in the following times:

Index/Homepage: 57ms

Employment: 52ms

Education: 56ms

Certificates: 1.90s

Hobbies: 47ms

Contact: 55ms

Subsequent refreshes loaded more quickly in a magnitude of roughly 10-20ms.

Firefox Extended Statusbar:

Index/Homepage: 0.125

Employment: 0.138

Education: 0.177

Certificates: 3.889

Hobbies: 0.093

Contact: 0.107

The unit of time being used here is not specified. With Firefox, the refresh times are dramatically quicker than the initial load times. The Homepage load time reduced to as little as 0.07, the certificates page more than halved its load time to 1.5.

If the unit of measurement used by Firefox's Extended Statusbar is seconds the web pages load significantly slower in Firefox, 57ms with Chrome and 1100ms with Firefox. Throughout the development process, I had been conscious to use as little code as was necessary. I had also already removed most of the unused CSS from the skeleton files prior to this. Another step I had previously taken was to remove the 'Font' line from the <head> section of my Index/Homepage page, which was the Index page provided by skeleton. I did this very early in the development process. I had been experiencing extremely slow speeds from the beginning when using the skeleton framework. I did not test these speeds at the time but they were extremely slow, 30 seconds to a minute to load an empty page. I googled this problem and found the solution, to remove the 'font' lines, in a forum. Removing these lines immediately improved performance.

Lines in question:

```
"<!-- FONT
```

```
----- -->
```

```
<link href="//fonts.googleapis.com/css?family=Raleway:400,300,600" rel="stylesheet" type="text/css">
```

Something that stands out in these results is the load times for the Certificates page. These times, through both browsers, are clearly unacceptably slow. Initially I had intended to fill this page, as well as the Hobbies page, with content similar to the Employment and Education pages. Due to time constraints, I had to abandon this and for a while the Certificates page looked identical to the current Hobbies page, ie. Empty. However, after consulting the assignment handout I realised that I had missed some potential opportunities for marks by not adding any video, audio or iframes to the website. Struggling to come up with something that fitted with the theme of the website, I decided on an iframe containing the President's Award website. This is the reason why my certificates page is so slow. Were I to use this website in the real world, I would of course remove this iframe to improve performance. I have left it in the assignment to demonstrate my ability to utilize and style a responsive iframe.

Unfortunately, since keeping the website light had been something I was conscious to do throughout the development process, I am limited to how many changes I can make in order to improve performance and then report a quicker speed. The speeds already specified are my adjusted speeds. One step I could take was to compress my CSS. I used the website <http://csscompressor.com> to do this. I have included the uncompressed CSS files in my submission for your reference.

Speeds with compressed CSS:

Chrome Developer Tools:

On first check my webpages loaded in the following times:

Index/Homepage: 37ms

Employment: 39ms

Education: 33ms

Certificates: 2.01s

Hobbies: 56ms

Contact: 63ms

Subsequent refreshes speeds improve by 10-30ms.

Chrome developer tools specify how much time it takes to read each page, the uncompressed CSS pages take 4-5ms, the compressed versions take 1-3ms, a decent improvement.

Firefox Extended Statusbar:

Index/Homepage: 0.126

Employment: 0.184

Education: 0.158

Certificates: 4.751

Hobbies: 0.125

Contact: 0.095

Subsequent refreshes reduce times by about 0.010-0.020.

As evident from these results, compressing the CSS resulted in a small increase in performance, particularly when using Chrome.

5. Responsiveness

The entire website has been designed to be fully responsive. This means that the website responds to the size of the browser window that is being used to view it, to ensure a satisfactory viewing experience, with little or no horizontal scrolling required.

5.1. Methods to ensure responsiveness

1. Use of skeleton framework:

I made use of the skeleton framework for this website. The framework adds CSS classes that specify the widths of divs and include media queries that modify the size of divs and design elements in response to browser window sizes.

2. Flexible images:

To ensure images would resize properly on a variety of browser window sizes I ensured that they have 'height: auto;' set as well as a 'max-width' value.

Eg:

```
.EmplImage{  
    max-width: 98%;  
    height: auto;}
```

This allows the size of images to be determined by the size of the browser window being used.

3. Use of percentages over pixels:

As much as possible, I have attempted to use percentage values to determine the sizes of divs, images and other design elements. Sizing by percentages is better than sizing by pixel values for responsive design because it ensures that design element will stay the same relative size in a variety of browser window sizes.

5.2. Testing Responsiveness

As mentioned above in the section on performance, my resources for testing are limited because my website is unfortunately not hosted. This is true for responsiveness testing as well as performance testing. One testing tool I could use was Responsive Web Tester for Chrome. This tool opens your website in a variety of browser windows corresponding to the dimensions of many common mobile devices. My website fared very well under these tests. I found that regardless of the browser size, the website responded appropriately. Images resize based on the size of the browser, and when the browser's width becomes very small, such as for mobile phone browsers, elements that would otherwise be organised side by side, swap to a vertical orientation. I also tested responsiveness by trying my website on a variety of computer browsers including Chrome, Firefox, Edge and Internet Explorer. The website remains consistent across all browsers.

6. Validation

To validate my HTML and CSS code I passed them both through an online validator. The results are discussed below.

6.1. HTML Validation

To validate HTML I used the website <https://validator.w3.org/#validate-by-upload>, which allows for validation by file upload, rather than only by URL. The validator identified three issues with my index/homepage:

1. Error: Saw < when expecting an attribute name. Probable cause: Missing > immediately before.
At line 14, column 1

`...r, bartender"<<<!-- Mobile Spe...`
missing closing tag '>' in meta data

Cause: This error was flagged due to a missing closing tag in the meta-data

2. Error: Bad value images/me - cropped2.jpg for attribute src on element img: Illegal character in path segment: space is not allowed.

From line 49, column 15; to line 49, column 93

`<`
Invalid image name – spaces

Cause: This error was flagged due to the name of the image file having spaces in it.

3. Warning: The document is not mappable to XML 1.0 due to two consecutive hyphens in a comment.

At line 85, column 5

<!-------Ri

Comments have '—' consecutive hyphens in them

Cause: This warning was flagged due to my comments having consecutive hyphens within them. I had been using comments to break up my HTML files into sections and had not realised that consecutive hyphens are forbidden.

Since issues 1 and 3 apply to all the webpages in the website, I updated these details in all of them before running the rest of my pages through the validator. There were no issues found in any of the other pages.

6.2. CSS Validation

To validate my CSS I used the website <http://jigsaw.w3.org/css-validator/>. The validator did not identify any issues with my CSS file.

7. Other Testing Methods

Although a variety of websites and browser add-ons were provided with which to test our websites, I found the majority of these did not work with my website. I presume in most cases this is due to my website not being hosted. Some examples of some of these websites and add-ons that I attempted to use are listed below:

- Browserstack
- Junkfill
- Validity add-on for chrome
- Web Developer Checklist (The checklist works however the items on the checklist for the most part do not)
- Website analysis by pagelocity

One of my major regrets with this assignment is not organising hosting at an earlier time, due to the testing restrictions I faced as a result.

8. Final Thoughts on My Website

My website is simple however, I hope that you will agree with me that is efficient, practical, consistently styled in a way that is aesthetically pleasing, completely responsive, lightweight and fast. Although light on content and features, it does everything it needs to as a personal website and utilizes forms, images, iframes, javascript, HTML5 and CSS3 in a way that fits with its theme. I feel there is nothing in the final product that feels out of place. In a perfect world, or given a long enough timeline, I would have liked to enter genuine data into the website, rather than Latin text acting as a placeholder. I would have liked to finish the Certificates and Hobbies pages. I would also like to have hosted the website and to have implemented an evaluation method. However, I am happy with my final product, and pleased that everything works so well and so responsively.