Development and testing

COM1008 website assignment

# Changes

## Menu system

### Mobile

#### Figure 1

#### Figure 2

Figures 1 and 2 show my initial mobile designs for the menu system when on a mobile device. This design was inspired from the menu overlay in <http://responsivenavigation.net/index.html> as shown in the design document.

#### Figure 3

Figure 3 shows the final implementation of the menu system for a mobile device. I decided against the design of a menu overlay as the menu button was just as awkward to tap as the links if it was a standard navigation bar. So, in order to keep in line with my accessibility statement and make the different page options big enough I used a similar design to the grid based method from <http://responsivenavigation.net/index.html>.

### Desktop design

#### Figure 4

#### Figure 5

## Degree page

### Desktop design

#### Figure 6

Figure 6 is my design for the degree page on a desktop screen size, but the two-column organisation was implemented for the tablet screen size instead as there was more than enough room for the text.

#### Figure 7

Figure 7 shows the desktop implementation for the degree page. I found that 3 columns used the space more efficiently than the two-column system for the tablet.

# Organisation

#### Figure 8

canvas.html

js

css

images

GeorgeSmith\_com1008

quiz.html

degree.html

index.html

contacts.html

accessibility.html

canvasVisuals.js

questionOne.js

questionTwo.js

questionThree.js

stylesheet.css

mobileStyle.css

tabletStyle.css

desktopStyle.css

jsStyle.css

Archie.jpeg

COM1001img.jpeg

COM1002img.jpeg

COM1003img.jpeg

COM1005img.jpeg

COM1006img.jpeg

COM1008img.jpeg

courseLanguages.jpeg

logo.jpeg

mapUK.jpeg

mountainsPortrait.jpeg

sheffieldBuildings.jpeg

Figure 8 shows the organisation of the files that work together to produce the site. All the different file formats are separated with html being the only one not in a separate folder as it is the html pages that use the css, js, and jpeg files. This is also in accordance with the assignment submission rules. I have used the W3C web page debugging tool which I will discuss in the debugging section of this document. As planned in the design document I allowed access to any page from any page as the navigation bar has all the links in it and is the same on each page. It is a very simple idea but allows easy access in accordance with my accessibility statement. I organised my JavaScript with one js file for each question on the quiz page and then one main js file for the canvas. In each file the code is split into functions to be activated by buttons like drawBarChart() or for efficiency like drawAxis().

# Optimisation

To improve image load times each image is stored in jpeg format which is a common example of lossy compression that drastically reduces file size as opposed to png format. The designs of each page are very simple and plain to suit the design ethos and ease of access. As a happy coincidence, this makes each page load relatively fast.

# Security

The main area of scrutiny for security risk is the email form submission on the contacts page. To help with this, I used POST method which is better than GET method as the parameters will not be passed through the URL of the site.

In addition, a common way to recognise a secure site is a small padlock next to a URL containing https. This means data being sent to or from the website is encrypted. This level of security is unnecessary for my website’s purpose as no significant private information is being used such as passwords or pin numbers.

# Debugging

I made use of the W3C’s validators for html (<https://validator.w3.org/>) and css (<https://jigsaw.w3.org/css-validator/>) by uploading each of my html and css files. They all returned no errors except one. There was a problem with the use of a p1 tag on the degree.html page which I used for the small text paragraphs for each module. Any errors with the JavaScript would appear in the console of web page and were corrected during implementation.

# Testing

Throughout implementation I was constantly using the chrome emulation tool to test different screen sizes on each page. The following images are screen captures from each page using 3 different screen sizes and therefore the three different css stylesheets: mobile, tablet, and desktop.

The width matches the devices; however, height was changed using the emulator in order to capture the whole page in one screen shot.

## Home page (index.html)

As screen size changes, only the text size and image size warps to fit within their bounds.

### Mobile view

#### Figure 9

iPhone 6/7/8 width

### Tablet view

#### Figure 10

iPad width

### Desktop view

#### Figure 11

Desktop width

## Degree page (degree.html)

As screen size changes, the layout changes from one long column on mobile, two columns on tablet, and 3 columns on desktop. This is to make sure text is legible while maximising space and not being too large.

### Mobile view

#### Figure 12

iPhone 6/7/8 width

### Tablet view

#### Figure 13

iPad width

### Desktop view

#### Figure 14

Desktop width

## Quiz page (quiz.html)

As screen size changes, the layout changes from the usual one column system on mobile and tablet to a one then two columns on desktop. Again, this is to maximise space but also to try and make the images appear with the question and buttons as much as possible.

### Mobile view

#### Figure 15

iPhone 6/7/8 width

### Tablet view

#### Figure 16

iPad width

### Desktop view

#### Figure 17

Desktop width

## Canvas page (canvas.html)

This is similar to the quiz page as the mobile and tablet layouts are the same. Again, only the desktop view has the width to display the canvas alongside the table and buttons to have them all in view at the same time.

### Mobile view

#### Figure 18

iPhone 6/7/8 width

### Tablet view

#### Figure 19

iPad width

### Desktop view

#### Figure 20

Desktop width

## Contacts page (contacts.html)

All screen widths are using the same one column method as to maximise the size of the email content box without increasing the height too much to encroach on the space of the other objects like the map.

### Mobile view

#### Figure 21

iPhone 6/7/8 width

### Tablet view

#### Figure 22

iPad width

### Desktop view

#### Figure 23

Desktop width

## Accessibility page (accessibility.html)

As planned in the design document, the accessibility page is very simple and has no real layout to change due to the small amount of content compared to the other pages.

### Mobile view

#### Figure 24

iPhone 6/7/8 width

### Tablet view

#### Figure 25

iPad width

### Desktop view

#### Figure 26

Desktop width

## Canvas page canvas testing

This testing will show each data visualisation with each colour scheme on the mobile screen size as this is a mobile first design.

I chose these 3 different colour schemes as they represent 3 different parts of the colour spectrum in order to stand out. This should also account for colour blindness accessibility issues.

### Pie chart

#### Figure 27

#### Figure 28

#### Figure 29

### Bar chart

#### Figure 30

#### Figure 31

#### Figure 32

### Line graph

#### Figure 33

#### Figure 34

#### Figure 35