<file:///W:/IT5036/Practical%20Tasks/ResponsiveWeb/Index.html>

I used various techniques to achieve responsiveness for my webpage as listed below.

<meta name="viewport" content="width=device-width, initial-scale=1"> This sets the scale of the page. SO it can resize by scale.

Making sure imported features have fast responsiveness:

@import url('https://fonts.googleapis.com/css?family=Roboto+Slab:400,700');

I used ems as they are a proportional measurement which allows for individual devices to scale the font compared to the base font, in this case it is set to 16 pixels.

Using 'position: relative' means that any elements placed inside the parent element will be placed in relation to the specified element itself. Which is useful during resizing as to where the elements move to and how they move.

#page ID limits the width that the page items can extend to. Brings the page elements in line with position: relative in this rule.

Using the 'display: block' rule for objects that would otherwise be inline so that they don’t span the width of the page.

Setting 'background-size' to 'contain' means that the image will fill and scale to the 'width' and 'height' properties specified.

By using a parent variable as in /\*section - all\*/ this ensures that all the content will not touch the left side of the screen when resizing.

In the main aside section the float property is used and set at 33% (in the width property) using float means the individual asides will line up next to each other. The use of percentages means the window can scale and the section will always only take up 1/3 of the page. It creates a 3 column row (can use 25% for four columns and so on).

Using svg graphics are good for responsive pages as the points within the image scale as the browser becomes smaller or larger.

Re-arranged the main section for small screens so they stack vertically. Achieved by setting float to none. This means they will now stack in order according to the html.

I used the developer tools available in the browser to identify screen size and element id so I could then find the code which needed to be modified.

I made sure the photos file size was not to large, large file size photos can slow down responsiveness on mobile devices.

@media queries for different size screens. I Re-wrote the CSS rules and styles so the page would display according to those rules when the screen is the defined max-width (or smaller) for different screen sizes using the developer tools in google chrome to view the page in different devices. This way I could figure out threshold sizes for making the media queries. In some cases the media queries are set to resize when different elements on the screen begin to look messy rather than just resizing to view on different devices. This is because some devices allow for multiple web browsers to be open at one time and next to each other and because there is no ‘uniform’ correct screen-sizes to aim for except for what products are popular at the time. New rules can be written for different screen sizes this way using ‘developer tools’ in the browser to identify screen size and element identification so new rules can be written to accommodate those elements. The media queries screen sizes I used can be seen in the CSS code provided for this assignment.