***CEP Project Documentation***

***Mission statement***

*To create a website which can help to educate people on the causes, impacts and solutions to ageism, mainly occurring at workplaces, with the hope that it would change the general population’s perspectives towards aged workers in a positive way.*

**General**

1. Implemented eg. Ageism.html#section1 in the nav-bar dropdown menu to directly jump to a particular section in Ageism.html.
2. Used vw, vh, em or % wherever possible, to ensure scalability and responsiveness when the viewport changes.
3. Used box-sizing: border-box; to ensure that all elements are sized in a more intuitive way.
4. Used “<meta name=’viewport’ content=’width=device-width, initial-scale=1.0’> to give the browser instructions on how to control the page's dimensions and scaling.
5. Provided alt names to all images in case some of them are not loaded.
6. I tried to adhere to the responsive web design information sheet on w3schools (<https://www.w3schools.com/html/html_responsive.asp>) as much as possible to ensure that my website was scalable and responsive.
7. Used margin: 0 auto;, text-align: center;, position: absolute;, float: right/left;, position and transform and flexbox justify-content and align-items to horizontally and vertically align elements accordingly.
8. Implemented google fonts where suitable to enhance the elegance of the webpage (Reference: <https://www.w3schools.com/howto/howto_google_fonts.asp>)

**Navigation Bar (Home.html lines 19-44 and Home.css lines 5-94)**

1. Used flexbox to arrange and order the elements in the nav-bar (*Home.css, line 6-18*)
2. Added a dropdown menu to Ageism and Gallery to jump to different sections in the ageism page and different galleries (*Home.html line 25-33 and Home.css line 31-63*)
3. Used class=’active’ to specify and indicate the current page on the nav-bar by adding a grey border-bottom to the element (*Home.css, line 73-75*)
4. Used a media query to arrange the logo, links and contact us button in a column if the device width is less than 800px (*Home.css, line 77-85*)
5. Used a google font called Amethysta to style the nav-bar elements (*Home.css, line 28)*
6. Implemented a border-radius of 5px around the contact us button to make it look more stylish and elegant (*Home.css line 70*)
7. For the navigation bar in the contact us page, since none of the elements had a margin bottom, I added in a transition effect for the bottom border to slowly appear when the link is hovered on. (*Contact\_us.css line 55)*

Comments: Originally, I was using the display: inline; float: left; method to style my nav-bar however, I soon realized that doing so was quite impractical especially when implementing the dropdown menu and the media queries. With Mrs Neo’s invaluable guidance and help, I used the example that she kindly provided to style my navbar accordingly. I also faced an issue with the dropdown menu not working when the padding beneath the elements was removed, however, this issue was quickly resolved by adding back the padding and adding an empty href=’#’ to the list element to treat it as a hyperlink. Also, when designing the navigation bar, I intended for the current page to have a grey bottom margin, and to do this, I used the class=’active’ method. However, for some reason which I was unable to figure out, the CSS code for .active was not being executed. Despite looking through my code over and over again, I was unable to find the root of the problem so to solve the issue, I added a **!important** behind the code under .active (*Home.css line 74*) to ensure that it does not get overwritten by another piece of code. Doing this resolved the problem for me.

References: <https://www.w3schools.com/css/css_navbar.asp> and <https://www.w3schools.com/css/tryit.asp?filename=trycss_dropdown_navbar>

and <https://www.w3schools.com/css/css3_flexbox.asp>

**Footer (Home.html lines 114-137 and Home.css lines 255-325)**

1. Used flexbox to arrange the elements in the footer using justify-content: space-around; flex-direction: row; (*Home.css line 286-290)*
2. Used a google font named “Proza Libre” to style the footer element headers (Location, contact and follow) (*Home.css line 293)*
3. Implemented a flexbox in a flexbox to arrange the links under “Follow” as a column. *(Home.css line 315-318)*
4. Implemented pseudo elements such as :hover, :link and :visited to manipulate the colours of the “Follow” links based on their current state. (*Home.css line 320-338)*
5. Implemented a media query to change the flex-direction from row to column and to center all text in the footer if the viewport width falls below 800px (*Home.css line 308-313)*
6. Implemented flex-shrink: 0; to ensure that the footer size does not get shrinked due to the content above (*Home.css line 274*)
7. Implemented “secret” href links on “Contact” and “Location”. Clicking on “Contact” will bring you to the contact us page and clicking on “Location” will bring you to a google maps page showing the address. Both links have been styled to look like normal text, making them “secret”. (*Home.html line 119 and 125, and Home.css line 298-302*)

References: <https://css-tricks.com/couple-takes-sticky-footer/>

**Home Page (Home.html, Home.css)**

1. Implemented a Navigation Bar to aid with navigation within the website (*Home.html line 19-44 and Home.css line 5-94)*
2. Used text-align, margin-top, margin-bottom, font-size and letter spacing to style the main header/company name and moto. I also implemented a google font named Bebas Neue for the header. I also added a cool shadow effect to the Main header/the company name to give it a nice ‘company signature’ (*Home.html line 48-52 and Home.css line 96-131)*
3. Implemented the use of blockquote to display 3 different quotes as elegantly as possible. To do this, I aligned each quote to the left and to the right alternatively. I also added a left or a right grey margin accordingly. I also implemented a media query to manipulate the padding if the viewport width falls below 300px, though it is unlikely. (*Home.html line 54-66 and Home.css line 133-181)* (Reference: <https://codepen.io/tomhodgins/pen/gPaPoq>)
4. Implemented a mixed column layout to display 3 elements side-by-side under the Our Services section, using variable width percentages/width=33.333% in this case. I also implemented a media query to stack the elements on top of one another using width:100%; if the viewport width falls below 800px. (*Home.html lines 72-76 and 203-208, and Home.css lines 183-223)*
5. Implemented an expanding grid to provide greater detail about our services when clicked on. I also added a close button to close the grid when the reader is done reading. (*Home.html line 78-107, line 139-150 (javascript) and Home.css line 225-245*) (Reference: <https://www.w3schools.com/howto/howto_js_expanding_grid.asp>)
6. Implemented a href link in the form of a button, which allows the user to find out more about the current hot topic, which in this case is ageism. I also added a shadow effect to the button and set the width to increase by 1% when hovered on. I set the transition-duration to 0.5s to make the animation more smooth and realistic to make it seem like the button is popping out of the screen. (*Home.html line 112 and Home.css line 247-266)*

**Ageism Page (Ageism.html and Ageism.css)**

1. Implemented id’s for sections in the webpage which I linked to the dropdown menu in the Nav-bar. This allows the user to jump to a particular section on the page as they wish. I also used html span, where appropriate, to place the tag more accurately in the sea of text. (*Ageism.html line 67, 79, 90 and 102)*
2. Implemented a background image for the entire page and used background-repeat and background-attachment to style the image accordingly. (*Ageism.css line 96-98)*
3. Implemented a full-page image and centered the heading over the image using the position and transform method. I also used a google font called Oswald for the header, to help it stand out more in the picture. I also added a shadow to the text to enhance the visual element of the webpage. (*Ageism.html line 54-61 and Ageism.css line 101-122)*
4. Added an animation to the subsequent picture (“Evidence of ageism”) to make it fade in over a duration of 5s. I also gave the picture a shadow using the box-shadow method. (*Ageism.html line 63-69 and Ageism.css line 124-154)*
5. Used classes *text-right* and *text-left* to alternate the side of the text in the 4 sections using float: right/left;. I also implemented clearfix to clear floats after each section. I also used text-align: justify; to make the content look more legit and professional (*Ageism.html line 71-116 and Ageism.css line 156-213)*
6. Implemented a scroll to top button which only becomes visible after the page has been scrolled down 500px. When clicked, it brings the user back to the top of the page. When hovered on, the background colour changes from red to grey. (*Ageism.html line 121 and line 150-167(JavaScript), and Ageism.css line 215-234)* (Reference: <https://www.w3schools.com/howto/howto_js_scroll_to_top.asp>)
7. Implemented JQuery and JavaScript code to change the background-color of the nav-bar from transparent/no colour to black when scrolled 300px from the top. I also added a transition-duration of 0.5s to make the animation feel smoother and more natural. (*Ageism.html line 169-176 and Ageism.css line 19-22)* (Reference: <https://www.w3schools.com/howto/howto_js_shrink_header_scroll.asp> and <https://www.w3schools.com/howto/howto_js_navbar_shrink_scroll.asp>)

Comments: I initially faced some difficulty implementing point number 7, however, with Mrs Neo’s help, I later realized that I had done the silly mistake of not importing JQuery into my html file. After importing it, the code worked just fine. I also faced another problem that sections of text would be cut off by the nav-bar when I assigned the “jump” id to the respective section. As a workaround, I assigned the id to the last paragraph of the previous section to ensure that the desired content does not get cut off. I also used html span to ensure that the tag was placed at the end of the paragraph, which was useful in more accurately jumping to the desired section. This was especially useful in the case of long paragraphs. (eg. *Ageism.html line 61, 72, 82 and 94).* Adding on, I faced some difficulty in adding a margin-bottom to the content divs. Cen Han kindly pointed out that margin or padding would have no effect if the display was set to inline, so I changed my code and it worked fine.

**Gallery Type 1 (Gallery.html and Gallery.css)**

1. Implemented a grid-based approach to the gallery using class row and column (*Gallery.html line 50-90 and Gallery.css line 97-137*) (Reference: <https://www.w3schools.com/css/tryit.asp?filename=trycss3_flexbox_image_gallery>)
2. Implemented a img:hover function to change the opacity of the image from 1 to 0.7 (*Gallery.css line 140-143)*
3. Implemented an image modal structure to the gallery such that an enlarged version of the picture is displayed above the content already on the page, when selected*.* I also added an animation to slowly zoom into the image modal. *(Gallery.html line 93-98 and 125-148(JavaScript), and Gallery.css line 139-214)* (Reference: <https://www.w3schools.com/howto/howto_css_modal_images.asp>)
4. Implemented a media query to change the grid from a 4-column layout to a 2-column layout if the viewport width is between 600px and 800px, and to change the grid from a 2-column layout to a 1-column layout if the viewport width is less than 600px. This was achieved by varying the flex and max-width variables of the images, from 25% to 50% to 100%. I also added an additional padding when the viewport is smaller, to prevent content from being cut off by the navigation bar. *(Gallery.css line 122-137*)
5. Implemented an animation to zoom into the image modal. (*Gallery.css line 181-189)*
6. Implemented a media query to change the width of the .modal-content to 100%/full width if the viewport width falls below 700px. (*Gallery.css line 210-214)*

Comments: While coding this section of the webpage, I was faced with the problem that some of the CSS code applied to all img elements were selectively effective on a few images and not effective on the rest. I was unable to figure out why this was happening, so I reached out to Mrs Neo for help, and she explained that it was because of the negative z-index of the image container. To solve the issue, I assigned the images z-index: 1 and changed the z-index of the navbar to be a larger number/5. This ensured that the image container does not have a negative z-index, whilst ensuring that the nav-bar stays on top of everything else.

**Gallery Type 2 (Gallery Type 2.html and Gallery Type 2.css)**

1. Implemented a “float-based” approach where I floated 4 imaged side by side by giving them a width of 25% each. To make the gallery more user friendly, I added CSS code to change the colour of the border to a darker one and to reduce the opacity of a picture from 1 to 0.85 when hovered on. I also added a target=”\_blank” to each of the images such that it would open up in a new tab. Furthermore, I implemented a small caption below the image to describe it and give the reader a brief description of the image. Lastly, I implemented 2 media queries to make the gallery responsive; First, to change the gallery from a 4-column layout to a 2-column layout by changing the width value from 25% to 50% when the viewport width falls below 700px. Second, to change the gallery from a 2-column layout to a 1-column layout by changing the width value from 50% to 100% when the viewport width falls below 500px. I also added an additional padding when the viewport is smaller, to prevent content from being cut off by the navigation bar. I also added a white shadow to the description text to enhance the visual element of the webpage. (*Gallery Type 2.html line 48-86 and Gallery Type 2.css line 97-145*) (Reference: <https://www.w3schools.com/css/tryit.asp?filename=trycss_image_gallery_responsive>)
2. Added in some text below the gallery to encourage the user to resize the browser window and observe the responsiveness of the webpage. I also included a button (*Gallery Type 2.html line 92*) to “open page in new tab”, which the user can use to open the same page in a new tab, which he/she can resize and play around with (continued in comments section). To make the button feel smoother, more life-like and realistic, I added a transition-duration of 0.5s to show a shadow around the button and increase the width of the button by 1% to make it look like its popping out of the screen, when hovered on. I also used display: inline; to position the text beside the button. (*Gallery Type 2.html line 90-93 and Gallery Type 2.css line 159-174*)

Comments: For the button, I initially wanted it to open the page in a new window (not tab), so that the user could resize the window and try out the responsive design of the gallery. To do this, I implemented the Window open(), ResizeTo() and focus() methods in js (Reference: <https://www.w3schools.com/jsref/met_win_open.asp>), however, I soon realized that it was not possible to open a link in a new window as it was, quote on quote, “misused a lot in the past”. (Quote taken from: <https://stackoverflow.com/questions/12939928/make-a-link-open-a-new-window-not-tab>) Regardless, I still kept the JavaScript way of opening a new page, just for a change from the usual href link and target=’\_blank’.

**Contact Us (Contact\_us.html and Contact\_us.css)**

1. Added a cool shadow to the landing page text to make it seem like the words are popping out of the screen. (*Contact\_us.css line 107-111)*
2. Implemented a responsive contact-us form using float to separate the image and the text. I also implemented a border-radius of 5px on the container to make it look nicer. The shade of green on the submit button also changes when hovered on. I also added a box shadow to the image to enhance the visuals of the site. (*Contact\_us.html line 54-328 and Contact\_us.css line 113-167*) (Reference: <https://www.w3schools.com/css/tryit.asp?filename=trycss_form_responsive>)
3. Used nested CSS to assign the same CSS code to 3 different elements. (*Contact\_us.css line 114*)
4. Implemented a media query to make the image and the text stack on top of one another rather than beside one another when the viewport width falls below 600px. The submit button also becomes full width in this case. (*Contact\_us.css line 161-167*)
5. Implemented JavaScript code to display an alert thanking the user for the submission and automatically clearing the form after the user has clicked the submit button *(Contact\_us.html line 356-363*)
6. For the navigation bar, since none of the elements had a margin bottom, I added in a transition effect for the bottom border to slowly appear when the link is hovered on. (*Contact\_us.css line 55)*

**Conclusion**

To test out the usability of my website, I asked my parents to navigate through my website and give me some feedback (they had not tried out the website before and didn’t have access to the code). They pointed out areas where the styling of elements could be better such as the font-size, colours, positioning, borders etc. I found these comments extremely valuable as I would not have been able to figure out these inconsistences as the one coding the website. I had tried out the same technique with my python adventure RPG game last year and saw how helpful it was, so I implemented it once more. Extensive comments can be found in the code files to increase readability of the code. I have put in a lot of time and effort into making this possible, so I really hope you like my website submission! Thank you and enjoy navigating!

Note: The entire website has been coded using Safari and some JavaScript codes may not work in other browsers. (eg. the action.php in the contact us form, does not work with google chrome but works with safari. This is the only one I’ve found so far. If convenient, please open the webpage in safari. Also, if possible, please open the file in visual studio code first before using the “open in safari” button to open the file and there seems to be some kind of slight formatting issue if the file is directly double-clicked on and opened. I sincerely apologize for the inconvenience caused, thanks!)