Project Exam 1

Sophie Haugland

Word count

3276

Table of Contents

1 Design

1.1 What went well on the project

1.1.1 Research and target audience

When I was decided on the theme of my blog site. I did some research to see what other blogs that were out there focused on similar themes. My blog is aiming to be a bit of light entertainment for other coders. Determining the target audience depends on 2 things the demographics and psychographics [1]. Demographics includes adults between the ages of 20 and 60 years working in the web development field, or hobby coders. Some of the psychographics of the target group are that they are tech savvy, like spending time on the computer, don't like websites that don't work as expected. Use cases include looking for amusement of an evening or during a break at work. It is therefore important that the site is easy to navigate, functions as expected and doesn't take too long to load.

1.1.2 Typographical Elements

I chose font parings and sizes with the help of google fonts pairing suggestions [2], and using advice from "Design For Hackers" [3] I was able to create a good visual hierarchy. As this is a blog based on bugs, I wanted to base the colour scheme loosely on an iridescent scarab beetle. I thought the mix of black, purple, and green could work well together. I used adobe color wheel to extract a gradient from an image of a beetle [4] (Fig.1) to choose colours that worked well together. I then adjusted the colours slightly using the accessibility feature to make sure they would be WCAG complaint. I saved the resulting colour theme to my library making it easy to colour my elements when designing in Xd.

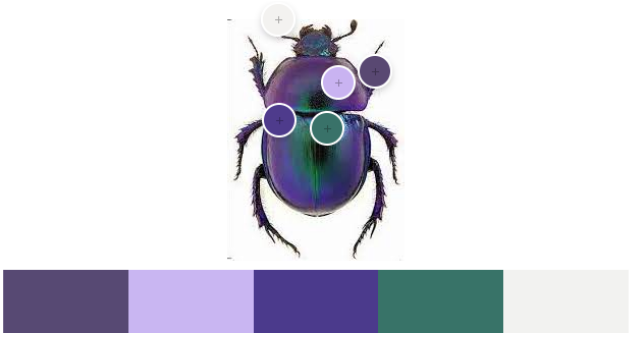

Fig-1

I first designed a few different style tile options, with different colour options and font types. After talking with several people in the target audience I chose on the design elements in this tile based on the feedback I received (Fig 2). I am glad I did this, because my favourite style was the least favourite of the users I asked. This shows the importance of asking users early on.

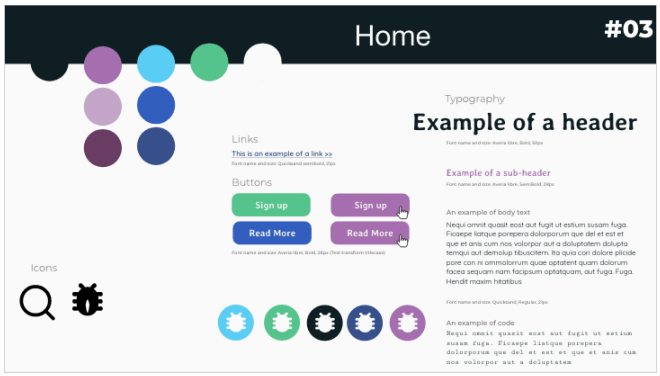

Fig-2

1.1.3 Prototyping

When settled on the design elements I sketched out some wireframes. I noted down all the elements I wanted to include on each page and where necessary sketched out how it would appear on smaller screen sizes (Fig 3).

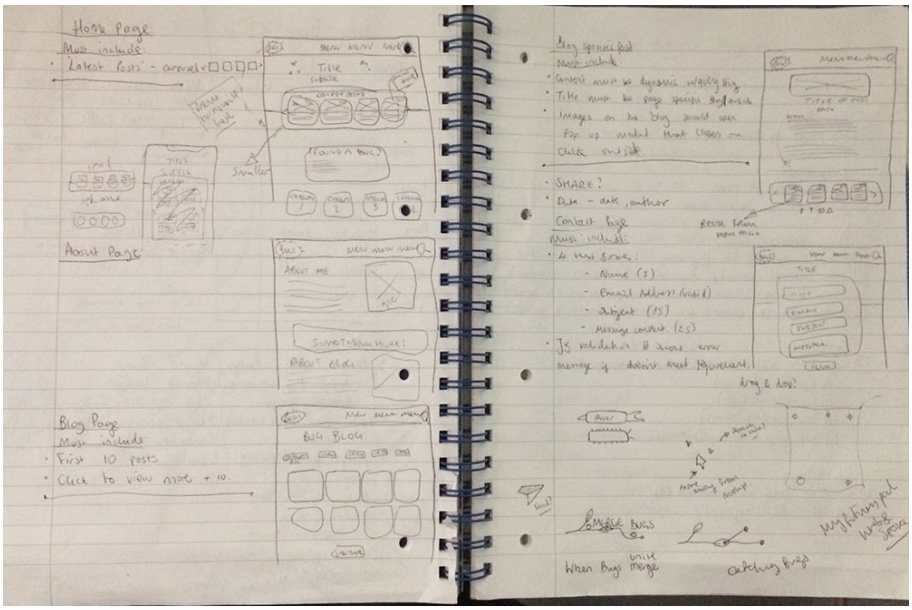

Fig-3

I then went on to make a high-fidelity prototype on adobe Xd that enabled me to test it out with a few users in the target audience. Here I created each page in 3 different sizes to help make the site fully responsive later during styling (Fig. 4, 5).

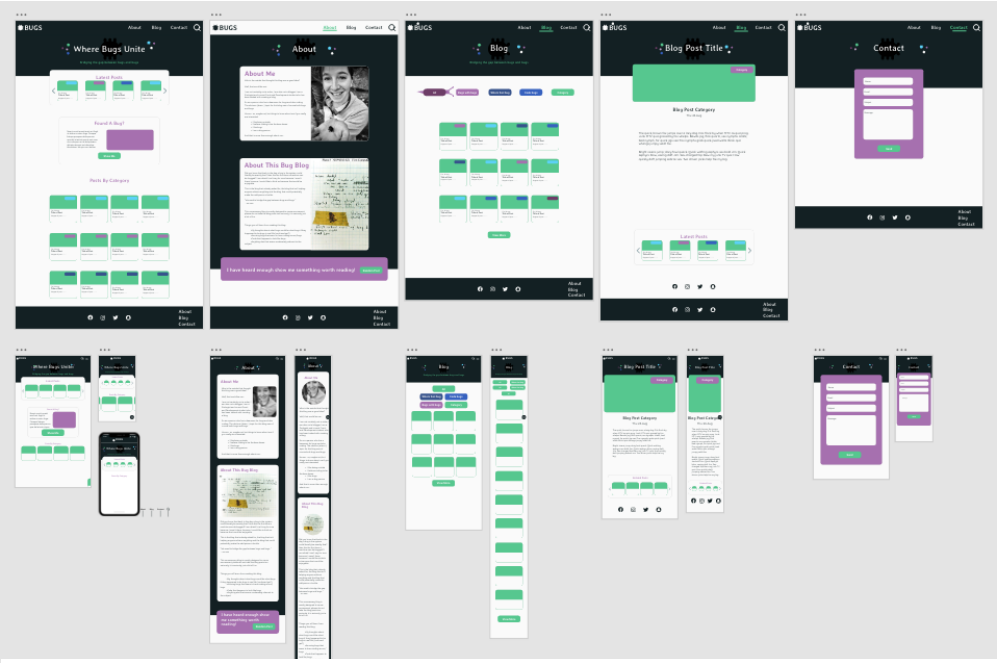

Fig-4

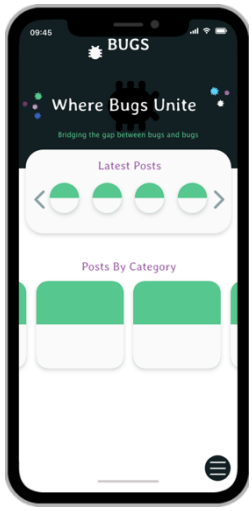

Fig-5

I used components in Xd to speed up the design process. I made a note throughout which groups I made into components in Xd, as these would likely be repeated code later. By doing this I was able to better plan the coding time and make reusable functions and components.

For some of the elements I used the golden ratio to proportion images to text as this is naturally pleasing to look at [3].

For the input fields if there was an error, I used the colour red to outline the box and red text on a white background to alert the user (Fig.6). I chose red because it is a common convention and associated with errors.

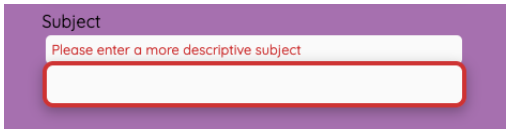

Fig-6

1.2 What was difficult/didn't go well on the project

Fig-7

As I mentioned earlier my favourite contender for heading font (Fig.7) was deemed unreadable by over 50% of the users I asked for opinions on the style tiles. I was reluctant to change it as I thought it looked good but had to remember that my opinion is not what is needed here.

I had originally designed a bug to go behind the heading, supposed to almost look like a shadow behind the text. However, when I user tested the design people remarked that it must be a mistake that it is so difficult to see. It was difficult to know whether they just thought they were giving me useful feedback or whether it was a genuine confusion for users. It was never really designed to be very visible just a nice little feature for those that noticed it. To avoid confusion, I removed the bug behind the headers in the hero section. Leaving the smaller more obviously coloured bugs instead.

I noticed from Hotjar recordings that the image modal caused some confusion; on desktop I changed the cursor to be a pointer to make it clear to the user where they could click. However, I failed to realise that this would not help on mobile and tablets. I soon found through user testing that users hesitated before working out how to close the modal on touch screen. I Therefore researched to see how other sites do this. I had a look at how Facebook [5] solved this problem and first it seemed very similar in functionality, I could click outside and it closed. I then noticed that Facebook used something that I had not. A common convention. A close button. I added this to my site remembering that accepted convention is an important thing for users [6]. I noticed on the heatmaps and user recordings from Hotjar that this reduced thinking time and was especially used by touchscreen users.

The menu button in my design was black, this worked well in theory but when I user tested the prototype it out it disappeared into all the other content so when I asked the user to navigate to a different page, they struggled to find the button. I changed the colour of the button to green which stands out from the rest of the page content. For phone users I noticed with larger phones that it was a struggle to reach a hamburger menu at the top of the screen. It is more common now to see the menu at the bottom of the phone on mobile, so I followed this. It was found 49% of people use just their thumb when navigating on a mobile device, and

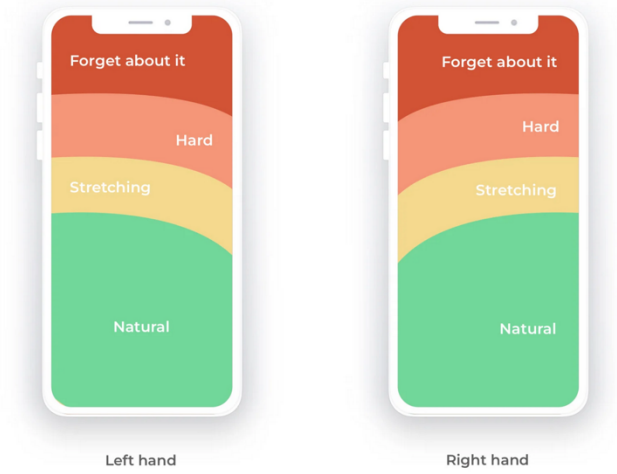

Fig-8 [6]

commonly top navigation bars are unreachable for these users see Fig.8 [7]. By putting the menu at the bottom for mobile users there will be minimal interruption to their flow and user experience will be better.

1.3 What would you do differently next time

I would get more users to try out the prototype before going to code it. This could have saved some time as I would have found out some of the design and user experience issues before coding them making fixing them easier.

2 Technical

2.1 What went well on the project

I read extensively the documentation for the WordPress REST API [8] which allowed me to make use of many of the different endpoints to create different functions for filtering, sorting, searching, posting comments, and sending messages.

To make the fetch requests for blog posts as efficient as possible I only ever fetched 6 posts at a time. When originally thinking about how to show a few posts and then have a view more button I thought I would need to fetch all the posts and then hide some. I realised quickly that this would not be a very efficient way of doing it, especially when the WordPress API has ways of fetching posts with pagination [9]. I therefore decided to limit the number of blog posts fetched to 6 for the main blog page, I then subsequently added number of posts already fetched to the number of posts to offset in the URL endpoint. This allowed for quick loading for the users.

Reading the documentation for the API allowed me to easily follow how to post comments from a site. As I will not be getting users to login to comment on my site, I was required to add a function to the functions.php file in my WordPress backend [10] [11]. I accessed the values from the form using JavaScript and created a headers object to send the information and save the comments to the backend.

I made a small game for people to discover in the hero section of every page. The bugs on the sides of the title can be squashed. On touch screens this is more of an easter egg as you would only notice if you clicked on one of the bugs. However, on desk top I made the bugs move on hover and become squashed on click. I also created a custom cursor that looked like a swatter. In Fig.9 you can see the bugs before and after swatting. I thought it was appropriate to add something entertaining such as these bugs and the snails on the buttons as this site is primarily for entertainment purposes. When watching the Hotjar recordings 100% of desktop users on discovering the snail buttons proceeded to hover on and of them multiple times.

Fig-9

To format the timestamps from the WordPress API I used the Date() constructor [21]. This allowed me to display the time and dates in a way that were readable to the user. This was a very easy way to format the dates.

On the blogs page I changed the text of the view more button according to how many blog posts were left to load. I also disabled the button when there were no more posts. This improved the user experience as they were able to see clearly whether there were more posts or not.

2.2 What was difficult/didn't go well on the project

When I got users to test my site, I found that some of the elements were weirdly placed on safari browsers. I then had to go back and find out that min-width was treated differently in a safari browser. This was a relatively easy fix but would not have been necessary if I had tested my site across all browsers before testing. I did this from this point onwards.

One user during a user test complained that the posts took too long to load on the carrousel on the home page. I thought about ways I could speed this up. I realised I was fetching the largest size of image for the small previews. I changed these to a smaller image size which seemed to speed up the loading considerably.

I had to find a WordPress plugin to allow me to post messages to the server, for this I used contact-form-7, supplying endpoints and functionality to post messages [12] and Flamingo [13] a plugin by the same developer that is used to store the messages. After reading up on the documentation and methods for collecting data from form. Using the FormData() constructor I was able to create an object with key/value pairs using the name and value properties from the inputs in the form, this was then used to in the body of the post request [14] [15]. With a lot of trial and error I was able to send and receive emails from the blog site to my email address.

I used the same code for form validation for the contact form and the comments form. This was a difficult task because I wanted to check the user input as they typed but I didn't want them to receive an error message as soon as they started to type as this is not a pleasant user experience. I therefore used a few different event listeners including key up and focus out. This allowed me to tell the user as soon as they had given a valid input and only told them about an error if they moved out of that field to type somewhere else. This made the flow easier for users as they can correct errors as they go through the form. In the user tests this was confirmed as all users were able to easily fill the form and correct any errors before reaching the end. I also added a feature that disabled the send button until the form was properly filled out. This was another helpful hint to the user that they needed to fill out the whole form.

I found it difficult to change the colour of the svg icons on hover after searching I found that I could use filters in CSS [16]. This worked effectively for on hover properties.

I found that when the user was scrolling the carrousel to the end they often continued to try to scroll, there was no indication that there was no more posts left to view (Fig.10). This is bad for the user flow. I therefore made a function on "scroll" that deactivated the button when the content was scrolled all the way to either end of the container.

Fig-10

When adding event listeners to the images on the blog post pages, I wanted to select the images rendered in the content of the blog but also the featured image at the top of the page. I therefore used querySelectorAll() to select all the images and then used the spread operator to append the featured image to this array before adding the event listeners.

For the search function on the page, I added auto focus to the text input, this makes it super quick for users to start typing. I didn't add a search button, but users can send the input by pressing enter. I found during user testing that users tried to leave the search area by pressing the escape key. I had not taken this into account in the beginning, so I had to add this. I created a new statement that would check if the user pressed the escape key, and this closed the search popup.

2.3 What would you do differently next time

I would have checked the site on all different types of common browser before user testing to sort out any bugs, styling or otherwise.

I added social media icons to the footer, and I would ideally have added function to these to allow a user to share the blog post straight to their social media site. This was however beyond what I was able to achieve this time. Next time I would spend some time researching how to implement this.

3 WCAG guidelines, content management and SEO

3.1 What went well on the project

3.1.1 WCAG Guidelines

I checked all my colours used on my site, the contrast ratios, and the font sizes to ensure they meet the standard guidelines for web accessibility. I made sure that all images had descriptive alt text and that the order of the content was in an appropriate order, especially important for those using assistive technology. I used the human interface guideline [17] to size buttons and icons ensuring it would be easy even for people with motor problems.

I used rem to size all text on the site, which allows those with magnification technology to enlarge the font without any trouble. I kept all paragraphs short using a width or max-width property to prevent users straining their eyes on large screen sizes. I also made sure text sections were left aligned and not justified as this also makes it easier and more comfortable to read. I refrained from using images behind text in my design as this can make it difficult to read. I added categories to the post preview cards but I colour coded them and WCAG checked them (Fig.11).

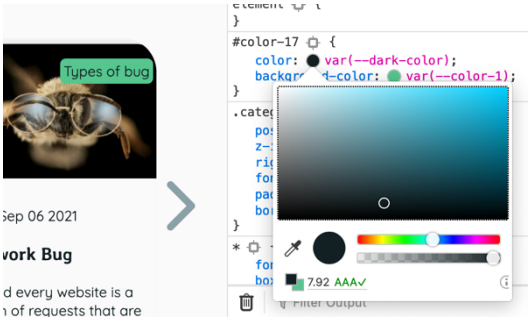

Fig-11

I ensured I kept all the original styles on form inputs as this helps people using assistive technology. I also ensured all inputs, buttons and text areas had clear labels also important for assistive technology.

3.1.2 Content Management

I sourced all my images from open sources that had creative commons licences (See appendices). I used images that had permission to both be used and edited. Many of the images I used in my blog I edited together using photoshop to make them amusing and appropriate for the theme of the site and to the target audience. I ensured that all images were below 200kb so there would not be trouble loading, but also that they are clear enough to view full screen for users who wish to do so. I used photoshop to resize files and crop images.

3.1.3 SEO

I used semantic elements where possible. I made sure that every page had a unique title and meta description, and 1 h1 tag. I ensured that all header tags descended in order. I made sure that all content was added in the HTML and all styling came from the CSS, including transforming text for titles. I used the prettier extension in vscode to ensure the code was kept neat.

I kept the style sheets DRY making sure I did not repeat styles in the media queries. Using CSS variables, I was able to make styling easier, I made variables for fonts, colours, and shadow styles. I also kept from repeating code in the JavaScript files by creating functions and components.

3.2 What was difficult/didn't go well on the project

I was unsure how was the best way to add icons to my site, a way that was useful for people with assistive technology. After doing some research I found that it would be best to use an img tag, this allows for alt text to be added [18].

When adding event listeners to the filter buttons I ended up with 5 very similar blocks with a lot of duplicated code. I went back and took some time to clean this up by using .querySelectorAll() to make an array of all filter buttons and add event listeners to each by iterating through them.

I forgot to add aria-labels to my buttons, so I had to go back and do this at the end. It would have saved time if I had done this as I went along. I used this site [19] for information about properly labelling buttons and other elements.

When checking my site with assistive technology I found that I had missed some labels from <a> tags I had styled to be buttons. I also checked the site using the accessibility plugin wave where I was able to catch and fix a few errors and alerts [20]. For example, in Fig.12 you can see that I had used a p tag for the logo text. This was not useful or semantic, I therefore changed it to be a tag instead.

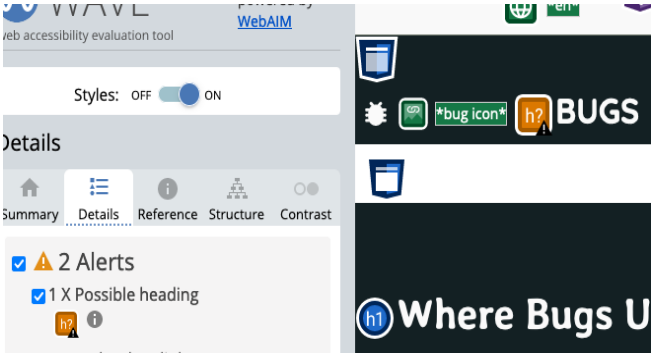

Fig-12

For the modal I added a close button, I added this mainly for people using touch screens to make it clear how to close the image. However, I realised that for people with assistive technology they would not be able to recognise this element and potentially if they opened the modal, they may not have been able to close it. I therefore went back and added a proper button element to solve this problem.

3.3 What would you do differently next time

After checking my site with an SEO checker [22] I found that there was a list of things I needed to do to improve my websites score (Fig.13). Next time I would pay more attention to this whilst coding and whilst creating content. Next time I would consider adding more external links to the page content and, I would try to add more helpful information. There are other suggested points including using 301 redirects. I have little experience with this, so next time I would like to spend some time researching this to ensure my site scores more highly.

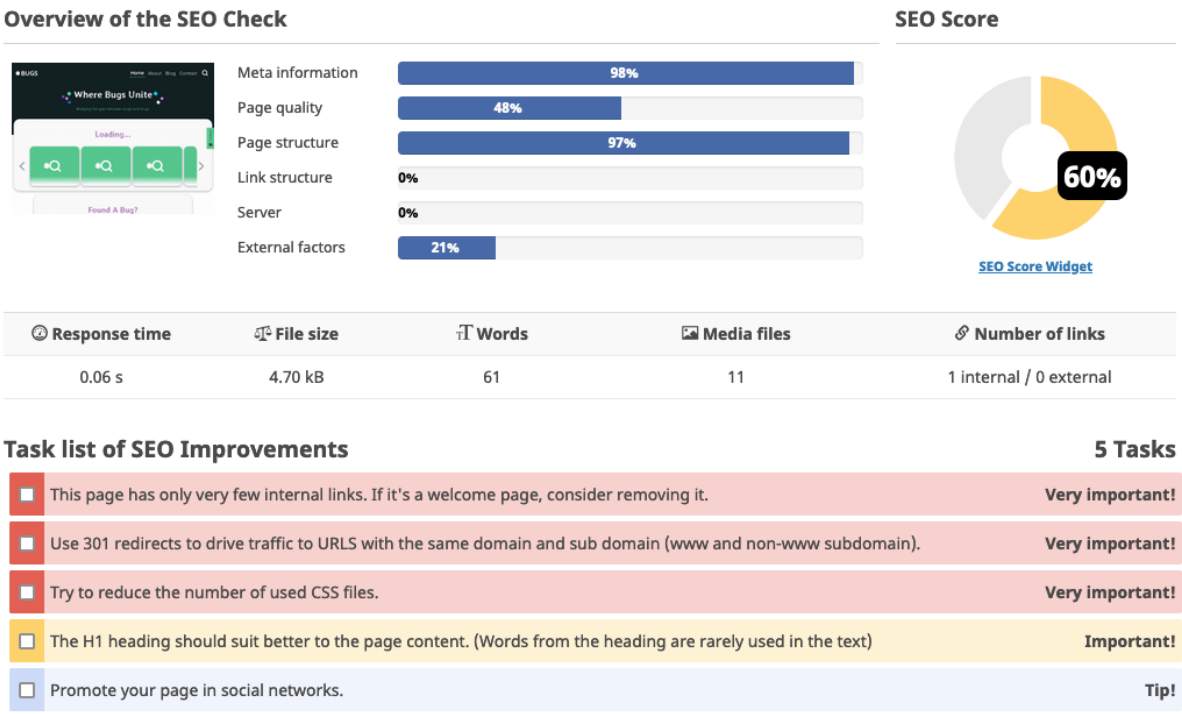

Fig-13

Bibliography

[1] S. Adams, "Layout and Composition: Marketing Collateral," 06 11 2014. [Online]. Available: https://www.linkedin.com/learning/layout-and-compositionmarketing-collateral/determining-the-target-audience?u=43268076. [Accessed 09 2021].

[2] Google, "Google Fonts," 2021. [Online]. Available: https://fonts.google.com/. [Accessed 09 2021].

[3] D. Kadavy, Design <for> Hackers, Chichester: John Wiley & Sons, 2011, pp. 180, 257-364, 280-284.

[4] Adobe, "Adobe Color," 2021. [Online]. Available: https://color.adobe.com/create/color-wheel. [Accessed 09 2021].

[5] Facebook, "Facebook," 2021. [Online]. Available: https://www.facebook.com/. [Accessed 09 2021].

[6] S. Krug, Don't Make Me Think Revisited, New Riders, 2014, p. 54.

[7] A. Leonovas, "Bottom Navigation Pattern on Mobile Web Pages: A Better Alternative?," 08 2019. [Online]. Available: https://www.smashingmagazine.com/2019/08/bottom-navigation-pattern-mobile-web-pages/. [Accessed 09 2021].

[8] Wordpress.org, "REST API Handbook," 2021. [Online]. Available: https://developer.wordpress.org/rest-api/. [Accessed 09 2021].

[9] Wordpress.org, "REST API Handbook, Pagination," 2021. [Online]. Available: https://developer.wordpress.org/rest-api/using-the-rest-api/pagination/. [Accessed 09 2021].

[10] Wordpress.org, "REST API Handbook, Comments," 2021. [Online]. Available: https://developer.wordpress.org/rest-api/reference/comments/#create-a-comment. [Accessed 09 2021].

[11] Wordpress.org, "Code Reference," 2021. [Online]. Available: https://developer.wordpress.org/reference/hooks/rest_allow_anonymous_comments/. [Accessed 09 2021].

[12] M. W. Docs, "Date," 2021. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date. [Accessed 09 2021].

[13] Wordpress.org, "Plugins, Contact Form 7," 2021. [Online]. Available: https://wordpress.org/plugins/contact-form-7/. [Accessed 09 2021].

[14] T. Miyoshi, "Contact Form 7," 2021. [Online]. Available: https://contactform7.com/save-submitted-messages-with-flamingo/. [Accessed 09 2021].

[15] Moz://a, "MDN Web Docs, FormData()," 2021. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/FormData/FormData. [Accessed 09 2021].

[16] Moz://a, "MDN Web Docs, sending form data," 2021. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/Forms/Sending_and_retrieving_form_data. [Accessed 09 2021].

[17] CSS-Tricks, "Solved with CSS Colorizing SVG Backgrounds," 05 04 2020. [Online]. Available: https://css-tricks.com/solved-with-css-colorizing-svg-backgrounds/. [Accessed 09 2021].

[18] Z. Dickerson, "Size matters! Accessibility and Touch Targets," 04 2018. [Online]. Available: https://medium.com/@zacdicko/size-matters-accessibility-and-touch-targets-56e942adc0cc. [Accessed 09 2021].

[19] T. Yip, "The Best Way to Embed SVG on HTML (2021)," 01 2021. [Online]. Available: https://vecta.io/blog/best-way-to-embed-svg. [Accessed 09 2021].

[20] bitsofcode, "HTML For Screen Readers - Labelling Elements," 2015. [Online]. Available: https://bitsofco.de/html-for-screen-readers-labelling-elements/. [Accessed 09 2021].

[21] WAVE, "WAVE Browser Extention," 2021. [Online]. Available: https://wave.webaim.org/extension/. [Accessed 09 2021].

[22] Seobility, "Seobility," 2021. [Online]. Available: https://where-bugs-unite.netlify.app/. [Accessed 09 2021].

[23] N. Access, "NV Access," 2021. [Online]. Available: https://www.nvaccess.org. [Accessed 09 2021].

Appendices

Links to open-source images used for my site:

https://www.flickr.com/photos/topgold/2842497804

https://pixabay.com/photos/orange-tip-butterfly-5051334/?download

https://en.wikipedia.org/wiki/File:Wanderer_male_female_philomela_WIKI.JPG#/media/File:Wanderer_male_female_philomela_WIKI.JPG

https://www.flickr.com/photos/bbusschots/5728366550

https://pixnio.com/fauna-animals/insects-and-bugs/butterflies-and-moths-pictures/question-mark-butterfly-on-common-milkweed-flower

https://www.pxfuel.com/en/free-photo-omjrd

https://www.piqsels.com/en/public-domain-photo-fxpcf

https://www.piqsels.com/en/public-domain-photo-jecvw

https://commons.wikimedia.org/wiki/File:Hormiga_panda.png

https://freesvg.org/bald-man-walking-in-a-suit-silhouette-vector-image

https://commons.wikimedia.org/wiki/File:Bee_hummingbird_(Mellisuga_helenae)_adult_male_in_flight.jpg

https://upload.wikimedia.org/wikipedia/commons/5/59/Stio_iks.jpg

https://www.flickr.com/photos/andreaskay/44094238755

https://www.maxpixel.net/Insect-Nose-Macro-Close-up-Filbert-Weevil-Bug-106114

https://www.flickr.com/photos/berniedup/16462044627/

https://www.flickr.com/photos/urtica/28097467522/

https://www.ravelry.com/patterns/sources/sophs-crafts-ravelry-store

https://pixabay.com/photos/beetle-mites-mite-infestation-lice-452994/

https://www.pxfuel.com/en/free-photo-jdcxl/download

https://www.pxfuel.com/en/free-photo-xwyur/download

https://unsplash.com/photos/P9mJHJCSCVc