Project Exam 1

Jonne Martin Krosby

Word count: 1524;

Design

What went well on the project

Since the task is to make a blog, I began surfing on other blogs for inspiration and to find out what is expected from a blog. I then went on drawing a wireframe by hand on a whiteboard, to get the sense of what I want it to look like. After finally figuring out what I wanted the blog to be about I made a style-tile and then went on to build the blogsite in Adobe XD.
I like designs that are simple and informative, so I ended up with making a design with that in mind. I made my own logo that I'm really satisfied with. I'm satisfied with the colour-palette, prototype, and the end-result. It ended up looking quite close to the prototype and what I pictured in my head.

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

My first big challenge was to figure out what to blog about. After making a short list of candidates I ended up blogging about my hobby, 3D-printing. Since it is a very niche hobby it thought, I would contribute to show others about 3D-printing. The colour palette was something I used way too much time on. At first, I went the safe way with a very white and light colours. I did not feel like it fitted the theme, so I decided to challenge myself to make a dark theme for the blog. After some testing of many colour palettes, I suddenly realized I could try the colours that are used on my 3D-printer. After some minor adjustments I found the colour palette I saw fit for this blog. I used my own photos for the posts, and it was a lot of work and very time consuming to find, crop and resize all the images. But it was worth it. Finding fonts that matched the theme for the site was a harder task than I had planned, but after some testing I ended up using Roboto for headers and CTA and open Sans for text.

What would you do differently next time

I do not believe I would do so much differently because I think the process that I have learned and used is working quite well. In the planning phase I will adjust more time to the designing part, because I used more time than I planned initially.

Technical

What went well on the project

I think that the technical aspects of this assignment went surprisingly well. I also decided after reading the assignment text that I wanted to do the optional level 2 process, just for the challenge and that I like to learn new things.

I like the way of using reusable functions making the code look cleaner, so I dived into using import / export JS module-based JavaScript for the first time on a project.
I gave myself a goal to make an extra effort to make as clean code as I can make, I spent a lot of time after getting some functions to work on refactoring them to be DRY and make them accessible through import/export if the function can be reused.
Setting up the Api and fetching data went well after reading some documentations and watching some YouTube videos. Getting the data, I needed from the WP rest Api was quite different from other API`s I have been practicing on, but in the end, I liked it because it has some functions that I decided use from it, like sorting posts and pagination.
Making the site responsive took less time than I have used on earlier projects, due to more experience and learning that I can use my phone and iPad with URL from the localhost and make adjustments according to a real device much more efficient.

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

The carousel slider was a bit of a hassle to get working as I would like it to do, I tried making it on my own with various poor result. The only decent result I made was using pagination from the Api on the slider part, but I wasn't satisfied with the result of that either, so I needed some inspiration for that part. I found a video that was precisely what I wanted, so I used that and made some changes to it. It's credited in the code and on the reference list.
Getting the right setup on the posts in WP was a time-consuming and kind of difficult to get right, I could have used custom endpoints, but I found out that wasn't necessary. The comments on the posts part was bit of a challenge, I struggled with getting the data accepted, so I had search all corners of the web to find out how I can send comments,(I didn't want it to be sent anonymous) I found out that I can make user on WP and setup an application key for authentication when sending the comments data.
Sending the form-data was almost same deal as the comments, but I used the contact form 7 plugin to getting the endpoints that I could send the data to, the formatting was a new one for me, the FormData() was the answer. I also struggled a bit with the load more function, because I wanted it to be flexible to the number of posts on the Api. But after some trial and error on using the postsPerPage endpoint and getting the header info about how many pages the Api has, it almost solved itself. I now got what I believe a code that can handle as many pages as possible or posts that the WP rest Api can contain. I wanted to add a search function, but I didn't get it to work as I wanted, and I struggled with error handling, so I gave up because of shortage of time, but I will look into this and make it work in the future when have more time to spare.

What would you do differently next time

I will use some time looking into how the different technical aspects would be done, before actually just start on them and maybe ending up not using them at all. I'm positive that I would save a lot of time and some frustration with some more carefully planning.

WCAG guidelines, content management and SEO

What went well on the project

I'm always taking into account the WCAG guidelines when designing a website.
By following the material from this course and WCAG check Tools.
I used tools like Wave validation tool and Adobe Color Contrast analyzer for checking colour contrast and overall site accessibility. I personally tested the site to check how it is to navigate with only a keyboard. After the test I needed to a lot of fixes because there were a lot of elements that didn't get focused via keyboard inputs, I made adjustments so that everything now is accessible and focused when the site is by only a keyboard.
I made sure that all images have an alt-text, and those other elements got an aria-label if I thought it needed it for the screen readers.

Setting up the CMS went ok in my opinion, I just followed the guides from WP, but using the Json from the CMS went well in the end after a lot of testing. I found out about some plugins that could make the CMS easier to use with custom endpoints, but I managed fine without them, so I decided to not use it.

For the SEO I made sure that each html page has a unique description and tags that fits the site.

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

I think everything about the WCAG is difficult because I use a lot of time on figuring out how all users that can access the site are going to be able to navigate the site. I am left with an impression that I have done ok on the accessibility part, but it I find it challenging. I have used a lot of time on the next button on the carousels Slider on the homepage. I have not managed to figure out how to make it not move when a user is tabbing trough the posts.
I decided to use my own personal pictures and some from prusa3D.com. There was time-consuming and difficult to make all the pictures fit for using on the web. The plus side is that I learned more on how to use photoshop.

What would you do differently next time

I would like to more research on available tools that can be used to be more efficient on the WCAG parts and the CMS part. I am confident that there is a more efficient solution to my WCAG and how I can setup the posts from the CMS.

References

(Place references to websites, books, forums etc. that helped you in the project)

- *https://www.youtube.com/watch?v=yq4BeRtUHbk*
- *https://code-boxx.com/javascript-fetch-auth/*
- *https://www.youtube.com/watch?v=e_thybKPKHc*
- *https://www.tetchi.ca/how-to-post-comments-using-the-wordpress-rest-api*
- *https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch*

- https://www.youtube.com/watch?v=RO40Opyizeg&t=12s
- https://cssloaders.github.io/
- Noroff course material
- https://www.prusa3d.com/ for some of the used images.
- https://stackoverflow.com/
- https://jigsaw.w3.org/css-validator/
- https://color.adobe.com/nb/create/color-contrast-analyzer
- https://wave.webaim.org/
- https://jasonwatmore.com/post/2021/09/05/fetch-http-post-request-examples
- https://cssloaders.github.io/
- https://webdevpuneet.com/how-to-remove-hover-on-touch-devices/#gsc.tab=0
-
-