A screenshot of a cell phone

Description automatically generated

# Project Exam 1

Kristine G. Alexandersen

Word count

I started this project by first figuring out what kind of blog I wanted to create. I explored several topics but eventually settled on a food blog because I wanted to create something I could benefit from at a later time. I enjoy cooking and baking, even though I'm not the best at taking pictures of food. Once that was decided, I started creating a simple wireframe in Adobe XD, using basic squares and circles to get a sense of how I wanted the layout to look.

After I was satisfied with the wireframe, I started thinking about which colors to use. I thought that a food blog should be colorful rather than having dark, neutral colors. I found an image on Unsplash that I used as a starting point for creating a color palette using Adobe Color. I made sure that the palette passed various accessibility tests for color contrast to ensure that the site could be read by those with visual impairments.

Then the fun began. Creating an API via WordPress was an exciting task. Setting up a headless Rest API on WordPress initially posed some problems for me. I couldn't access the wp-admin on my new WordPress site and was constantly redirected to my main wp-site when I tried to create a subfolder. I did a lot of searching on Google, contacted support on one.com, and chatted with ChatGPT. I tried deleting and creating a new wp-site a couple of times before I finally got it to work.

I entered the recipes and added the pictures I had of the food recipes I wanted to display in the blog posts, and then I started with the coding. I wrote the structure of the pages I wanted in HTML, where I included the necessary elements in the head section to ensure that icons could be displayed and that the site was responsive. I wanted a simple navigation between pages that would be displayed horizontally in the header section on larger screens, while a hamburger menu would be placed in the top right corner on smaller sizes, such as iPads and mobile screens. I implemented this using CSS.

Of course, a blog needs a logo, so I created one using 'canva.com' and edited it to fit the formats and sizes I wanted in Adobe Photoshop. It took a few iterations before I was satisfied. To start, I wanted to get the layout of the site going, so I hard-coded images into HTML to get a proper picture of how the site would look. I started with the home page and wrote CSS code to create the carousel we had been tasked to build. We were given the option to create a simpler version for smaller screens, but I chose to keep the carousel and make it responsive. I watched several YouTube videos for inspiration and help in creating this, and ended up with a version where I used some other people's code but adapted it to fit my content and the appearance of the site. I was quite satisfied with the end result. I wanted the site to be somewhat interactive without going overboard, so I created a small transform effect on hover.

I prepared the other pages to add content from my API, including the contact page where I wrote a contact form with specifications for the number of characters required. I used JavaScript to display an error message if the user does not enter the correct email address or enough characters in the various inputs. <-- write about getting information into WP -->.

I removed the hard-coded images I had in HTML and began the process of trying to retrieve information from my WP API. I tried to view the code by entering <https://unipop.no/bloggapi/wp-json>, but it was very messy, so I put it into <https://www.postman.com/> to see the code in a neater format. There, I could get an overview of what I had and get an idea of how I could retrieve this information and display it on my site. I used the 'fetch' method to retrieve it with JavaScript and also displayed the code there in the console log. This was enjoyable work, but when it came to retrieving the images, I encountered some problems I didn't understand. Again, with intense searching on the internet and questions to ChatGPT, I found a method that allowed me to display the images as I wanted. Then it was just a matter of styling them to have the size and layout that I wanted for my site.

I added Hotjar code to the head section on all pages, deployed my site on Netlify, and sent the link to several friends and colleagues. I asked them to click around the site and provide feedback if there was anything they didn't like. I received good feedback from some of them, and when I watched the recordings of how they used the site, I uncovered a couple of significant errors that I had not noticed in the mobile version. This was very useful for me to fix the errors and also to get tips on things my friends pointed out were missing or good with the site.

The most important thing for me in this task, and generally when I write code, is that if I find hints on how to do things online, I want to understand what I am doing and how different things are connected. For example, I struggled with how to focus on the image in my API and searched for how to do this. I received some advice, but nothing that I managed to make work with the code I had. ChatGPT helped along the way, and I got it to work, but that's not enough. I had to study the code I wrote. I made changes to understand how the different functions work because without understanding, one cannot become a good coder in my view. This has been an exciting task; I have found that design may not be my strongest suit, but rather that the coding part and the ability to retrieve information from APIs are very enjoyable.

## References

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