



# Project Exam 1

Dennis Alekseev

1138

GitHub repo:

<https://github.com/Noroff-FEU-Assignments/project-exam-1-mrDennisA>

Web deployment:

[https://yunitto.online/noroff/FEU1/AW33-37\\_Project-Exam-01/](https://yunitto.online/noroff/FEU1/AW33-37_Project-Exam-01/)

# Design

Design process went relatively smoothly. There were some compromises made at the end for reaching accessibility standards, but the essence of the design theme where relatively intact.

Since we were not given any style direction for this exam, I always wanted to make a really clean, minimalistic and white design with some contrast elements for guiding users in the right direction. It was more about making a technical design that was easy to adjust and not focusing too much on style and standards of what a blog site should look like. I also didnt want to make a standard navigation menu for the desktop view. To keep everything minimalistic I went with a hamburger menu for all screen sizes.

Design prototype was done relatively quickly, it was meaby not complete but it had all elements I wanted to realize. I had to do some trial and error to find the right font, which was scrapped for another one after a week when i started to finetune the design. Prime colors were also picked quickly. White moste of the site, orange for navigations, icons and buttons and dark blue for anything that is text related. Secondary colors where gradients of primary colors to make some elements stand out or be secondary actions like focus and active states.

Unfortunately at the end I had to scrap my orange color to meet accessibility standards. I went with dark red, it was a nice color too, but I really loved my orange. It's a type of compromise I hate to do as a designer.

Logo for the site was going to be plain text, but I wanted to make it more than just a logo. For the prime site name I went with my initials but I incorporated the title of the page and breadcrumbs directly into the logo. So the logo, title and breadcrumbs were all in one.

Icons were made directly in the code, just as a challenge. At first they were just plain html code with css style, but I did change some of them into svg with the same principle. Goal was to not make any icon assets

but just be a snippet of code I put where it's needed, the only exception was "favicon" that was made into a png file.

All other content for the site were random free stock photo assets combined with lorem ipsum text to illustrate the website theme.

## Technical

"Don't repite code" was my motto for this exam. Clean simple code, avoid repeating it as much as possible with my limited knowledge and learn new tricks along the way.

First task on my agenda was to move as much as possible html code into javascript so most of the html pages would look the same but content would be different. I hope we will go in depth into php code next semester so html will rather be in there then in javascript. But it will do for this assignment.

Most of the coding was straightforward, of course there were exceptions like carousel, show more button and how to make them work with WordPress rest API. But for the most part I had a theoretical idea how to solve them and what to search for online. It was relatively easy to guide my selv true jungle of codes and solutions for things that I could understand and implement. One think that was really important for me was if i find a solution for a problem but i didt understand how or what the code did, i didt use it. I just searched more till I found a solution that I could understand, implement and reuse again for different tasks.

Hamburger menu was one thing I was looking forward to making in javascript. All my previous assignments used checkbox work around with css, but this time I went all in with javascript. I am actually a little proud of myself for making a working hamburger menu on my first try. Without searching for a solution, just me and what I have learned this semester (patting myself on the back).

Carousel function was something I haven't done before. We had a video event about some part of it, but everything else was new for me. I think it was most time consuming from the whole project. I did have an early breakthrough with "pagination" API call using numbers of pages and offset. It did work but the problem was it constantly calling the API for next or previous posts, and it became even worse when i tried to make it compliant with screen sizes. So I had to redo it all with just calling it ones and filter it locally.

Blog page and show more function was a lot easier after i solved carousel function. I only had to filter how many posts I needed to show at any given time. Rather than using the button to show more I moved it to scroll function, so it automatically loads more at the bottom.

Contact form was straightforward, we have done it now a couple of times with different assignments. So now it was more about how clean and not repetitive code I can make.

WordPress Rest API was not enjoyable to use. Felt like I had a restraint jacket on and worked with information that was either hard to obtain or was unnecessary. I really look forward to using other methods to make API, ex. React.

We were also given 3 other optional assignments. Rather than implement them I focused on cleaning my existing code, redoing some part of it if I found a better solution in a later stage of the project.

Only one think that i didnt had time to implement was animated carousel. When clicking the next or previous button it would animate in or out new or previous posts. It wasn't necessary for the assignment, but it would be nice to have.

# WCAG guidelines, content management and SEO

Guidelines, accessibilities and SEO aren't my strongest features. I tried to implement as much as possible, but I feel like I always miss something out. Thankfully one of this year's students, Marie Lowzom Morland is very passionate about it. She guided me and reminded me of some features that I was missing. I think I need to sit down and actually make a proper checklist that I always go to before any code is written.

Content management with WordPress was not that hard to manage. I understand easy accessibility with using WordPress but i still didnt enjoy using it.

## References

Research for most of the code solutions where 3 site frincently used.

<https://www.w3schools.com>

<https://developer.mozilla.org>

<https://stackoverflow.com>

Stock photos were from 2 sites.

<https://www.pexels.com>

<https://unsplash.com>