A screenshot of a cell phone

Description automatically generated

# Project Exam 1

Anders Styve

Word count

## Summary

This report covers the written part of the Project Exam 1 at Noroff’s Frontend-development class.

The purpose of the exam was to create a blog site from scratch, putting together all the skills learned over the first year of studies. The blog is created with HTML, CSS and JavaScript, but with the main content hosted on a headless WordPress site. Using the WordPress REST API the content of the blog (posts, comments etc.) is to be fetched and displayed live on the finished website.

In the first section of the report all things regarding the design of the blog is discussed. Included are brief explanations on what my thoughts were during the design process, especially what parts of it went well, and what parts I found more challenging. Finally, I reflect upon what I would do differently, or what changes I would make the next time engaging in a project like this.

Similarly, the next section of the report covers the technical part of the exam. This section also discussed the parts of the assignment that went well, as well as the parts where I found myself struggling to put the pieces together. This sections covers most of the actual programming of the blog and all its connected contents, and is also finished by some thoughts on the things I would do differently another time.

The last section covers the processes and steps that were taken throughout the process of creating the website regarding the sites accessibility and optimization. This section covers the work done to follow the WCAG-guidelines, as well as search engine optimization. This section is also made up like the others, with a part covering what went well, a part reflecting on the more difficult matters, and final part discussing what changes could be made on a future project like this.

Link to final website: https://objective-brown-e68010.netlify.app/

## Intro

## This report will be divided into three sections, each covering a specific subject relating to the exam submission. The three different sections are: “design”, “technical” and “WCAG guidelines, content management and SEO”. In each of those three sections I will elaborate and discuss what I thought went well on the project, what didn’t go so well, and also what I’ve would have done differently a next time.

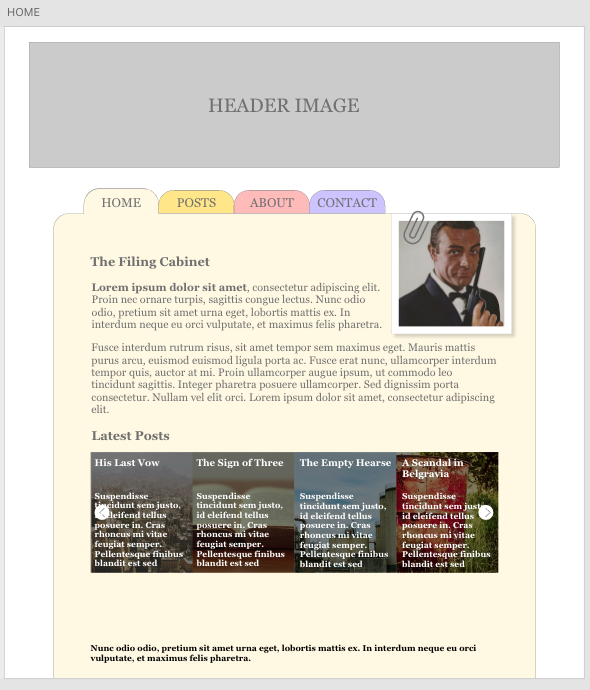
## Design

In this section of the report I will explain my thoughts and processes behind the design of the website. This includes both the initial layout created before the actual programming of the site, and the design choices that were taken throughout the process of developing the site.

### ***What went well on the project***

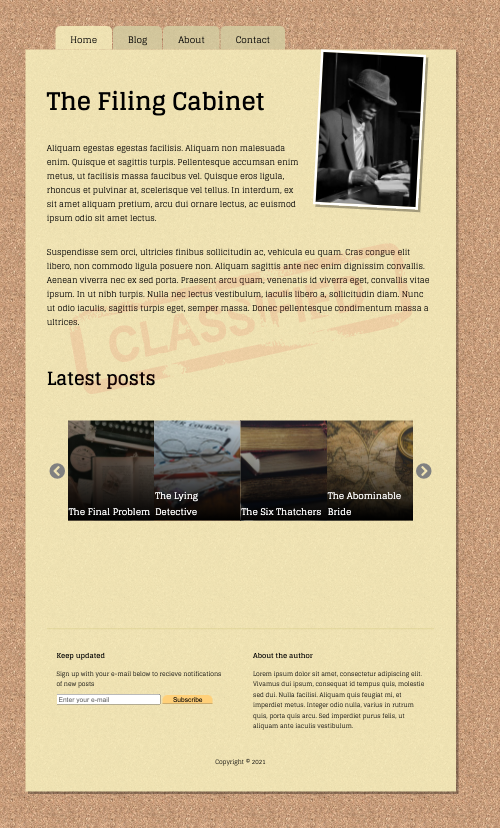
The inspiration for this blog website initially came from older James Bond movies and video-games, as well as classic movies or shows involving FBI or older investigation or detective units. I wanted to create a blog that looked and felt like an older filing cabinet consisting of an amount of blog posts, each covering an unsolved mystery.

Based on those initial ideas and concepts, the first design layout for the blog ended up looking like the following image:



As shown in the initial design, I wanted the blog to consist of a home page consisting of a brief introduction of the blog, an image of the author, and a section showing the latest blog posts in an image carousel. The navigation bar is designed to look like the kind of navigation one could see from filing cabinets. This would also be the inspiration for the title of this fictional blog.

When transitioning the design from that initial layout and into a functional website, some changes were made, but the initial concept stayed somewhat the same. The picture below shows the final version of the index page for the blog.



After taking inspiration from my own actual desk, I decided to change the background of the blog from plain white to a pattern made to look like a cork board. This was done both because it was more aesthetically pleasing, but also because it gave the blog section a nice contrast against the background. The colour of the blog was also changed to a more yellow tone trying to mimic the real life colour of old documents. At last, the header image was removed, as it served no real purpose, and only pushed the blog further down, making it longer and more in need of scrolling.

**What was difficult/didn’t go well on the project**

To further develop the idea of this archive or filing cabinet concept, I initially wanted the blogs posts looking like they’ve been hand printed on an old typewriter, just like old reports from the movies and shows that was the inspiration for this site. After finding and trying multiple typewriter fonts, each with its own design and functionality, I unfortunately had to fully discard this design element. The main reason for this was that even though how authentic the typewriter fonts looked and behaved, they we’re just not readable enough to be used in paragraph after paragraph on a blog like this.

Also finding a correct cork board background that had the right colour and rules of licensing, as well as being repeatable multiple times without showing toke some time and effort. Having had difficulties regarding the file size of images used as website backgrounds before, I also had to have that in mind when trying to find the perfect pattern.

**What would you do differently the next time**

Even though I did some research and collected inspiration from various other blogs and websites, a thing I would do different another time is taking enough time to design different initial drafts and layouts. Especially on a relatively large time frame like this. When first designing the layout for this blog I was so focused and excited about my first initial design thoughts, that I may not have spent sufficient amount of time exploring other ideas. Even though you have one initial favourite design or layout, using time trying to create something entirely different could be time well spent as one could stumble upon ideas or solutions that could be implemented into the initial design.

**Technical**

This section of the report includes all the technical parts behind the creation of the blog. This includes the programming of the website, and the transition from an initial design to a fully functional website, or blog, with all necessary features.

**What went well on the project**

The scope of this assignment was to create a website that uses WordPress as a kind of database to store all it’s content, and by using API calls this content would be displayed on the website. Not only should the website be able to make a call to the WordPress database to fetch and display information, the website should also be able to send, or post, information directly to the database to be stored there.

Firstly, after setting up my WordPress site and making it headless and connected to my website, I created a number of blog posts directly in WordPress. These posts consisted of a title, a body of text, and an image. To display these posts on the blog an API call to the WordPress database was made, which returned a set number of posts and displayed them accordingly on the site. The first issue was to only show the 10 first blogs, and then having a button underneath that, when clicked, showed more results, or blogs, underneath the initial 10. The way this was solved was by utilizing the WordPress REST API’s built in “per\_page” argument, where the default value is 10. So when the initial API call is made, by default only 10 results will be shown.

So by binding a function that makes a call with the “per\_page = 2” argument to the “Load More” button, the returned posts will be every post from number 11 to 20. These posts will then be displayed underneath the initial 10. By solving it this way, one also secures that the newest blog posts will be displayed at the top of the site, as a new post will automatically be number 1 of the 10 first shown posts.

To implement a search functionality to the site, one could also use one of the built in arguments in the API called “search”. How I practically implemented this into the site is that the site makes a new API call with a search argument that is gathered from an input text box on the site for the user to write a search query in.

One of the main challenges involved in this assignment was the creation of a comment section on each blog post that allowed users to view, write and submit comments directly to a specific blog post. The way this was solved was by creating a function that gathers the values of the input elements that the user would fill in. After these values are validated and gathered, a fetch request with a “post” (rather than “get”) method is sent to the API and then, if successful, a body containing the string values of the input is posted to the WordPress database on the correct blog post ID.

Two other main challenges in this assignment was creating functionality for the user being able to subscribe to a newsletter, as well as sending in a “contact us” form to get in touch with the developers behind the blog. The main task regarding this was creating a way to store the inputted information from the users directly in the WordPress database to then be accessed by the admin or author.

The way these challenges were solved was by creating a fictional blog posts for each of them, utilizing categories in WordPress to separate them from the actual blog posts. By doing this, I ended up with three categories of blogs posts: Actual blog posts, a contact form post, and a newsletter post. Then, similar functions to the ones created for the users being able to make and post a comment to a blog posts, were made to the contact form and newsletter. The way this worked is that when a users submits the contact form, the information submitted is stored as a “comment” on the blog post that is named “Contact Form” in the WordPress database. This way, is submission is stored in the right place. Similarly, a “comment” with the user’s email address is stored under a fictional blog post called Newsletter.

These fictional blog posts would never get in conflict with the real blogs posts, as long as they remained correctly categorized. This is again because of the built in argument in the WordPress API that is used to display only the posts connected to any given category specified in the API call.

**What was difficult/didn’t go well on the project.**

Working with WordPress as a database did have some limitations and possibly demanded a little more effort when trying to achieve the preferred outcome. This was especially the case when trying to combine the right endpoints and arguments when fetching data from the API. A concrete example was trying to figure out how to fetch a specific blog post related image, which was not included in the standard URL when connecting to the API. To retrieve said images, a string containing “?\_embed” had to be included in the API URL. This was just one example of a situation where retrieving the wanted data from WordPress was not as straight forward as preferred, and demanded some time and effort in configuring the right paths for accessing the API.

**What would you do differently next time.**

One of the biggest drawbacks I’ve had through the programming of this website is that the way it is created now, its sending a lot of API calls back and forth to the WordPress database. Every a user clicks, searches or comments on a blog post a new API call is sent. Of course, most of the calls is surely needed to have the functionality that is wanted, but I do also believe that there could be ways to increase the efficiency, and reduce the loading of the website by utilizing e.g. arrays to a greater degree. To explore this opportunity of making fewer API calls and instead storing the received data into arrays to be sorted and filtered from there is something I definitely would look closer into the next time working on a project like this.

### **WCAG Guidelines, content management and SEO**

This sections elaborates how i worked according to the WCAG guidelines during this assignment. In addition, the thoughts and decisions regarding content management and search engine optimization (SEO) will be further discussed.

**What went well on the project**

One of the first objectives I had when starting to develop this site was the management of the site’s contrast. I initially wanted to create a site with between the elements of the page would be as good as they needed to be to avoid any chance of confusion or miss-clicking from the users. To further optimize the readability and clarity of the site, I decided to use only two different fonts, as well as limiting the number of different font sizes.

One especially challenging feature to implement regarding contrast and readability was the title text overlay on the images in the “Latest posts” image carousel on the front page. Having text on top of images can often be an issue when it comes to the user being able to effortlessly reading the text. I solved this issue by having a gradient overlay on top of the images going from black with low opacity at the bottom, to an almost transparent top. This made the bottom half of the images darker, and subsequently increased the contrast to the white title text, enhancing the readability.

Other important implementations that were made was fetching and creating image alt texts directly from the WordPress API, so that the alt texts always would be up to date and in line with the content of the images. Another correction that was made directly after assessing feedback and recordings from users through the services of Hotjar, was that the titles of the blogs post were made into links, as many users tried to click them, rather than the button at the bottom, to access the blog post.

**What was difficult/didn’t go well on the project**

One specific issue that kept coming back was regarding how to properly size the images that were supposed to be rendered on the site. Downloading and displaying the images on the site is one of the most time consuming processes and high is highly influential when it comes to both the actual and the perceived speed of the website.

The main reason to why this was a challenge, was that the assignment specifically stated that the blog post image should be clickable and, when clicked, opened into a modal giving the user a bigger view of that image. This meant that the original image resolution had to be bigger than the resolution displayed on the site, in case any user wanted to click the image to view it in a larger resolution. To find the balance between the image file size and resolution was therefore a challenge, but in the end it turned out satisfactory.

**What would you do differently next time**

If it’s one thing I’ve learnt throughout the process of making this website, or blog, it is that it’s never too early to involve external users and invite them to have a look and an opinion regarding your site. I felt that it’s very easy to get stuck in ones own ideas of how a site should look and “feel”, and forget about small details that others may discover within seconds of viewing your site. One particular example was the lack of a “back” button to navigate backwards from a specific blog post to the overview of the blog posts. This function did not cross my mind at all during the creation of the site, but was immediately requested by two separate external users during their initial view of the site. This just shows how easy it is to get stuck and “not see the forest for all the trees” when designing or programming sites that are intended for users to interact with.

Another feature that I most certainly would like to spend some more time exploring next time is everything related to the total file size of the total website. A good example is looking into the next-gen formats for saving and displaying images or pictures on a website. With the end users demanding faster and more streamlined websites than ever before, I think looking into this would be a good long term investment when it comes to frontend development.

## References

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

**Text References**

* Contrado Digital. «How to Post Comments to the WordPress REST API Version 2», 2016. Internet: <https://www.contradodigital.com/2016/04/06/post-comments-wordpress-rest-api-version-2/> [Accessed 27 August 2021]
* MDN Web Docs, Mozilla. «Using Fetch», 2021. Internet: <https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch> [Accessed 31 August 2021]
* REST API Handbook, WordPress. «Comments», 2021. Internet: <https://developer.wordpress.org/rest-api/reference/comments/>Accessed [27 August 2021]
* Section.io. «How to Make an Image Carousel Using Basic HTML, CSS and JavaScript», 2021. Internet: <https://www.section.io/engineering-education/how-to-make-an-image-carousel-for-your-website/> [Accessed 25 August 2021]
* Tetchi Blog. «How to post comments using the WordPress REST API», 2019. Internet: <https://www.tetchi.ca/how-to-post-comments-using-the-wordpress-rest-api> [Accessed 27 August 2021]
* Wikimedia Foundation. «List of Sherlock Episodes», 2021. Internet: <https://en.wikipedia.org/wiki/List_of_Sherlock_episodes>[Accessed 19 August 2021]

**Photo References**

* Bulbfish. Pexels, 2021. Internet: <https://www.pexels.com/photo/vintage-music-business-money-9264012/>[Accessed 29 August 2021]
* Classified stamp. FreeSVG, 2021. Internet: <https://freesvg.org/classified-stamp>[Accessed 20 September 2021]
* Cork board pattern. Tiling Textures, 2021. Internet: <https://www.tilingtextures.com/cork-board/>[Accessed 15 September 2021]
* cottonbro. Pexels, 2021. Internet: <https://www.pexels.com/nb-no/bilde/mystisk-forstorrelsesglass-forskning-retro-7319068/>[Accessed 29 August 2021]
* cottonbro. Pexels, 2021. Internet: <https://www.pexels.com/photo/geometric-building-on-street-with-tramway-5235178/>[Accessed 06 September 2021]
* cottonbro. Pexels, 2021. Internet: <https://www.pexels.com/photo/geometric-building-on-street-with-tramway-5235178/>[Accessed 06 September 2021]
* cottonbro. Pexels, 2021. Internet: <https://www.pexels.com/nb-no/bilde/teknologi-musikk-datamaskin-lyd-8382621/> [Accessed 06 September 2021]
* Gonullu, M. Pexels, 2021. Internet: <https://www.pexels.com/nb-no/bilde/ute-gammel-park-anlegg-6907994/> [Accessed 29 August 2021]
* Hazelwood, S. Pexels, 2021. Internet: <https://www.pexels.com/photo/stacked-books-1333742/>[Accessed 29 August 2021]
* igovar. Pexels, 2021. Internet: <https://www.pexels.com/nb-no/bilde/statue-historie-innendors-militaeruniform-9306496/>[Accessed 29 August 2021]
* Ilina, Y. Pexels, 2021. Internet: <https://www.pexels.com/photo/white-book-page-on-black-and-gray-typewriter-9107393/> [Accessed 29 August 2021]
* Ilina, Y. Pexels, 2021. Internet: <https://www.pexels.com/photo/white-printer-paper-on-brown-wooden-table-9107394/>[Accessed 29 August 2021]
* Koppens, Y. Pexels, 2021. Internet: <https://www.pexels.com/nb-no/bilde/vintage-retro-retning-guide-697662/> [Accessed 29 August 2021]
* Koppens, Y. Pexels, 2021. Internet: <https://www.pexels.com/photo/letters-and-an-eyeglass-on-table-1809342/>[Accessed 29 August 2021]
* Kozlov, A. Pexels, 2021. Internet: <https://www.pexels.com/photo/geometric-building-on-street-with-tramway-5235178/>[Accessed 29 August 2021]
* Roseclay, D. Pexels, 2021. Internet: <https://www.pexels.com/photo/close-up-photo-of-black-typewriter-977930/>[Accessed 29 August 2021]
* Smith, J-M. Pexels, 2021. Internet: <https://www.pexels.com/photo/brown-paper-envelope-on-table-211290/>[Accessed 29 August 2021]
* Smykova, D. Pexels, 2021. Internet: <https://www.pexels.com/nb-no/bilde/bord-reise-vindu-bok-8369435/> [Accessed 29 August 2021]