



Project Exam 1

Marie Lowzow Morland

<https://innovation-coach.netlify.app/>

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

Words: 12582



School of technology and digital media

Summary

This report will go through the process and reflections I made while making my first blog with Headless CMS and a custom frontend written in HTML, CSS and plain JS. The blog is a student project where I will show my abilities to create a user-oriented product by user testing and applying the knowledge I have gained over a school year. This report will explain the entire process starting from how the design has been developed and ending with the last feedbacks for users who only got to see the result. I'll go through what the main obstacles were, as well as the solutions I implemented and the thoughts behind them. However, the focus for this report will center around user testing, accessibility and how feedbacks in these areas drove the development and led to improved user experience in the end.



Content

Project Exam 1	1
Summary	2
Introduction.....	5
Process.....	6
Design and technology foundation.....	7
<i>Initial Design – The gardener</i>	7
Buttons and shapes:.....	7
Pictures:.....	7
Fonts:.....	7
<i>Wordpress og serveroppsett</i>	8
<i>Revised design - The band aid:.....</i>	8
First web version.....	10
<i>Folder structure</i>	10
CSS structure	10
JS structure	10
<i>Pages</i>	10
Feedback	11
<i>API-calls.....</i>	11
<i>Contact form</i>	11
<i>Search.....</i>	12
<i>Carousel</i>	13
<i>Modal</i>	13
User feedback	14
<i>Who and how.....</i>	14
<i>Design changes – Connecting the dots.....</i>	14
Colors	15
Navigation and Layout.....	16
Loader.....	16
<i>HotJar.....</i>	16
<i>Findings and solutions.....</i>	17
Video on iOS	17
Colors	17
Full height frontpage	18
Contact form verbose.....	18
Redesign DOM order (search in top, showing previous search)	19
Empty state (no search hit)	19
Don't repeat first post in carousel.....	19
Post summary on each post on top (user didn't find search keywords)	20
Accessibility	20
<i>Why all the effort</i>	20
<i>Testing.....</i>	21



Testing with users	21
Testing with tools	21
Engaging a Facebook forum	22
<i>Findings and solutions</i>	22
Skip links.....	22
Video with autoplay	23
Long anchor-tags.....	23
Colors	24
Aria-current.....	24
Aria-labels.....	24
Aria-live (carousel, contact form and user notification).....	24
Menu close after list (not implemented yet)	24
HTML structure	24
Alt-text.....	24
Headings.....	25
Custom focus.....	25
Last round of feedback	26
<i>Some new fresh eyes on the site</i>	26
<i>Test with a blind user</i>	26
Discussion	28
<i>What is innovative design on a webpage?</i>	28
<i>Page headers</i>	28
<i>When and where to split a function</i>	29
<i>When to make a function generic</i>	29
<i>CSS selectors</i>	29
<i>Functionality vs accessibility</i>	29
Future work	30
Conclusion	30
<i>Key learnings</i>	31
User Testing.....	31
Technical solutions.....	31
Accessibility	31
Reference:	33
<i>Accessibility</i>	33
<i>Bugs and function</i>	34
<i>Pictures:</i>	34



Introduction

The assignment for this project was to create a blog with a theme of the students' own choice. It instructs us to use specific pieces of technology that we've learned for this past year, as well as add some requirement to what should be included in the result. As a content management system (CMS), we've been instructed to use WordPress (WP) and to create our blog articles here. Yet the webpage that readers encounter should be custom made and hosted separately. The technique of "hiding" the WP interface and instead hosts another that fetch data with an API is called Headless CMS.

The blog site should be designed, developed and tested by the student, using tools and techniques learned throughout the year. During this report I'll go through how I've used tools like Adobe XD for design and prototyping, Git as a version control system, Netlify to deploy our site, Hotjar to capture user behavior and our skills with the core technologies such as HTML, CSS and JS. Along the way, the page has been tested by multiple users and was adjusted based on their feedback to ensure a user-friendly experience.

This report will cover:

- The initial planning of the project
- How the initial design came to be, and how and why it was adjusted
- The process of setting up WP and deciding which plugins and APIs to use
- How I structured the code and made the first version of the site, describing some of the key functionality.
- How I collected user feedback through interviews and Hotjar
- The tools and techniques I employed to get a good grip on accessibility
- Iterations of the project based to the feedback above
- Thoughts and decisions on how I prioritized areas of improvement
- Future work, where some key improvements are discussed
- User feedbacks late in the process



Process

The topic I have chosen for my blog is innovation and entrepreneurship since I've been a lecturer one the subject for several years. I've also considered to write a book on the subject and have a lot of self-made illustrations that I use in my teaching. I thus based my project on the Noroff assignment description and these illustrations.

While I considered using a tool to plan my project like i.e. Trello, I decided against it and rather made a rough work outline a shared "note" on my mac and iPhone. The outline was to use the first week to form a design, ask for feedback while creating a code foundation. While users tested an XD prototype, I would then use the next week to try and figure out the required functionality that I had not previously encountered and not focusing too much on styling. Then I would style the page according to a revised design and try to get more feedback that would form a basis for future decision, while I could add the last bits of mandatory functionality. In other words: My main plan was to check in with users often and try to save grunt work and JS functionality that could take a lot of time for the times when I waited for user feedback.

All the while I filled in checkpoints in my note for "must have" functionality or bug fixes, as well as thoughts and ideas for improvement. This was particularly used a lot while I had my son in a stroller and listened to a podcast about accessibility. Since WCAG 2.0 is required for all business related sites that targets a general audience (<https://www.uutilsynet.no/regelverk/kva-seier-forskrifta/153>), and since I am fond of the aspect of creating something that is available and user-friendly for everyone, I've weighed learning and employing good accessibility practices quite highly when prioritizing work. One could say that my main order of priority was:

1. Do all mandatory requirements
2. Develop the site user-centric to ensure a good experience
3. Make the site accessible to everyone, and try to conduct a good user test with a person with some sort of disability
4. Expand to optional functionality

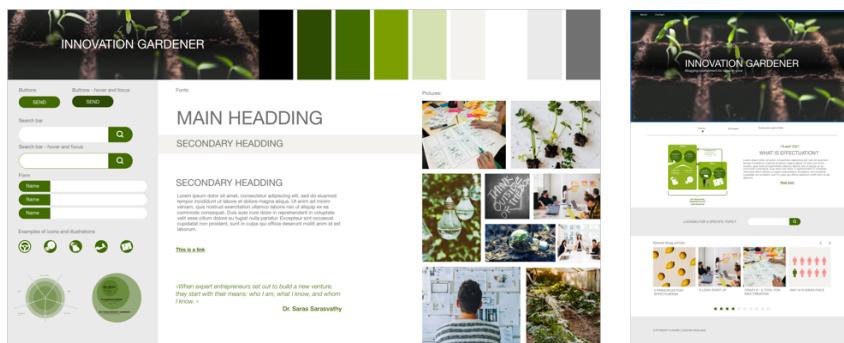


Design and technology foundation

During this project, I have been through several design concepts and I have never been afraid to make changes. I have tried to have an agile approach to the assignment and to create the page with my users by inviting them to join the design discussion. We started to discuss what associations we have to innovation and entrepreneurship, and later I asked them what they thought about shapes, colors buttons, pictures, and navigation and function. Adobe XD has been used both as a tool to create design, icons, and SVG-files that I could use directly on the website, as well as to create a prototype to test navigation and content, and as a tool to sketch and illustrate ideas.

Initial Design – The gardener

My first concept was based on a metaphor that is widely used in the professional community: the idea is a seed that needs nourishment and care to grow and become something big. I named the blog Innovation Gardener and had an idea for a green color theme with images that played on the metaphor.



Style tile and design of initial home-page under the title “Innovation Gardener” with the slogan “Blogging nourishment for your ideas to grow”.

Buttons and shapes:

Since I already had several illustrations, I had a basis for a design language for the page. The illustrations have round shapes and edges, and therefore it was natural that the buttons should be rounded. Labels, background for icons, input fields and search fields are shaped in the same way.

Pictures:

The pictures are picked to match the headings and topics for the blog articles. Some of them illustrate different tools and methods that is naturally to explain in the text. Others are mainly decorative and evocative.

Fonts:

I imagine that some of the posts on the blog can quickly become quite long to cover a topic well enough. The paragraphs are therefore written in the usual Helvetica font which is a font well suited for long texts on digital surfaces. The headlines are in century gothic which is also a san-serif font, yet here the letters are more circular in shape. The letter O, for example, is completely round, which speaks to the circles in the illustrations.



Wordpress og serveroppsett

After I gave some users the link to my first XD prototype, I started to set up WordPress on my web server. Since we had just had an assignment on how to set up WP as a headless CMS, this process was fresh in mind. I added my installation files to a new folder, and installed plain WP with a plugin called “Headless Mode” that I used in my previous assignment. With a small change in wp-config, I now could redirect people to my own Netlify-site if they tried to visit the url for WP, with two exceptions:

- I can still log in to manage content in Wordpress admin tool.
- The API url still goes to Wordpress, so that the API functions properly.

For this assignment, I read through the WP native API endpoints, and found the standard API to be sufficient for the level 1 tasks and did not need to add any php functions. Just to be sure, I tried to add a few Lorem ipsum posts with a few images and paragraphs to test the API with. Going back to my design to see what I would need, I also made sure I could easily fetch the post date and found out how to include a featured image as well as a summary called excerpt in WP.

During this process I also made an initial commit to GitHub that said, “Hello exam!”, and connected the GitHub repository to a Netlify-site just to see that all the “wiring” worked, and that I could now start focusing on my custom frontend site.

Revised design - The band aid:

Although I think the idea of my first design and name “Innovation Gardner” was good, I discarded it after a round of user feedback since it seemed that the analogy didn’t work for someone who is not familiar with the jargon. In addition, it was difficult to find good free photos that covered the topic. When I tested the prototype on users, someone suggested that the call to action (CTA) button should link to the about page. I took this as a clear signal that my design didn’t convey that it was a blog about innovation and entrepreneurship.

In a new attempt to create a design concept, I used the shape of my buttons to create my own cover illustration. I picked blue as a color to give associations to technology, professionalism and trustworthy.



2nd Style tile and design of home-page, with different color options that I tried out



The Feedback from two of the test users was that the site gave associations about health-related activities. On the one hand, a health-related company should give the feeling of competence, security, trust, professional and science, which is not very different from some associations I also want to give even though my site is not health-related. As a disclaimer both reminded me that they work as nurses and may be a little colored by their jobs. However, after they said that the shape of the main illustration resembled a band aid, I could no longer stop seeing it that myself. In conversations it emerged that they not only wanted more colors, but first and foremost something that conveyed a creativity and innovation. Preferably something that surprises and maybe challenges the reader in some way.

At the same time, it was mentioned that the use of white space was good and that the page was easy to read. One of the users was disappointed that the posts contained mostly "lorem ipsum" and had become curious about the content. I interpreted this as a sign that the design was able to visually convey the content of the blog in a greater extent than in my first attempt. The test also showed that the navigation on the blog worked well, and the users felt they got a good overview of the content. The dot under the menu that signals the current page, was very well liked by the users and something that is included in the final design together with the overall structure from this design iteration.



Conversation where users associate the design with health care and a health care company



Conversation where users give feedback on colors, the amount of air/whitespace and that they like the current page marker.



First web version

Since the design and prototype feedback came quite quickly, I revised the design and created the band-aid theme before I coded anything more than “hello exam!”. Thus, I changed my title and the URLs to both Netlify and WP to match the new name “Innovation Coach” rather than the older gardener analogy. The new name seemed to convey my content better, and also hinted at a potential for a side business for me where I can fill in actual content later and use this blog as a way to sell courses and workshops.

Folder structure

My folder structure is a result of how my earlier projects have been. I try to have my HTML-pages on the root level, then I have a corresponding CSS and JS file located in CSS/pages/ and js/pages/, respectively. These are the files referenced directly from the HTML-page, and other CSS files and JS files are imported inside of these as needed, so that my HTML files are kept small and tidy with a single entrance point for styles and scripts. I.e.: The HTML file “about.html” has a corresponding “css/pages/about.css” and “js/pages/about.js”. This structure has helped me maintain order even as the number of pages grow.

CSS structure

I have a hierarchy where files located in “common” are imported first, then “components”, then “templates” and lastly the rules in the page-specific CSS file itself. I found this a good way to make the cascading part of CSS prefer the most page-specific rules. Also, I found that I personally used a lot less time to find a set of rules when I used multiple files rather than fewer yet longer ones.

JS structure

Templates

This folder house JS-files that primarily contains HTML-templates. In this project those files are the header, footer and SVG icons. These are repeating pieces of HTML that are the same for multiple pages. Instead of copy-pasting the HTML into each page, I found that I prefer to use a template that I insert with JS. Thus, changes made to this page is instantly reflected in each page with no risk of forgetting one. During a previous assignment, I discovered an extension for my VS code that highlight such HTML templates the same way it does in the HTML files, which makes these template easy to read.

Pages

As mentioned above, each HTML-file has a corresponding JS-file that import and trigger other pieces of code like the template for the header. In addition, they house logic and page-specific rules that doesn’t repeat across other pages.

Components

These are pieces of code that have some logic tied to them, and possibly some HTML as well. They are separated from the page-specific code mainly because they:

- Repeats across multiple pages, such as the search bar
- Contains a piece of code that has a concrete function that could be easily separated and potentially used somewhere else, i.e., the modal

The latter reason is somewhat related to my preference to rather have multiple files with a name describing their purpose than to have longer files.



Data

This folder only contains the code for communication with WordPress API, so that my other files does not need to know the details of how the API works.

Feedback

I've put a great deal of effort in finding a good JS cod structure before, and therefore I did not actively look for people who could give me feedback on the code in the same way I've looked for test subjects for how the user experience is, yet I got a nice message from a fellow student who was inspired after I posted the page min on discord:

The website looks REALLY nice, I'm not sure about the color yellow, but I am more impressed by the js strucrtion. I was in the middle of organizing and removing repeating code myself, but your code structure inspired me to adapt and update my code the same way you structured it. Nice work

API-calls

In previous assignments, we've never encountered the standard WordPress API for reading posts. Yet reading the about possible arguments from

<https://morlanddesign.one/innovation-coach-admin/wp-json/> gave me the information I needed to paginate, search and to filter on categories. In addition, fellow students mentioned the “_embed” flag that were not listed on the endpoint, yet made WordPress embed information about featured images, which saved me from having to do multiple requests in order to retrieve the image URL.

In addition, I later noticed two response headers “x-wp-totalpages” and “x-wp-total” when I inspected the network calls where I retrieved posts, using google dev tools. I ended up using the former to remove the “more posts”-button when I had fetched the last page, instead of my first approach where users could press it until the result was empty.

Contact form

I reused my own “formValidation.js” from two earlier assignments, and once again had to make some adjustments. First, I moved user notification logic to a file of its own since several imports in my previous assignment included formValidation just to use the notification part.

Later on, when I did a lot of visual improvements and fixed accessibility, I also added a step to my component that rewrote the form HTML around input elements so that I did not have to write repeating code to add i.e., error-message div tags. An example is showed below:

Code in HTML	<div> <label for="fullname"> Full name (More than 5 characters) </label> <input type="text" name="fullname" id="fullname" required>
--------------	--



	<pre> data-minlength="6" /> </div></pre>
Resulting HTML	<pre><div> <label for="fullname"> Full name (More than 5 characters) </label> <div class="input-wrapper"> <input type="text" name="fullname" id="fullname" required="" data-minlength="6" aria-describedby="fullname-error"> <div class="field-icon"> <button aria-label="Toggle visual error message." class="error-info-toggle"> <svg xmlns="http://www.w3.org/2000/svg" width="36" height="36" viewBox="0 0 36 36"> ... </svg> </button> <div aria-label="Field valid." class="field-valid-icon"> <svg xmlns="http://www.w3.org/2000/svg" width="25.331" height="30.61" viewBox="0 0 25.331 30.61"> ... </svg> </div> </div> <div id="fullname-error" class="form-error"></div> </div> </div></pre>

As you can see, the amount of HTML now needed to write manually for is centered around the label text and input rules, while the form script adds what is needed to style and function properly. Both the functionality changes and the need for these changes are further discussed in the accessibility section.

Search

The search started off as a plain HTML where the user could choose between their own textual search criteria and a few hard-coded categories. These values would then be sent to the “all posts” page as query params, that the page would pick up and use to search and filter posts fetched from WP.

Since both inputs were optional, and none had any validation rules, I decided to create the



form without any JS support. Later on, I made a template so that I could reuse the HTML on both the home page and on the “all posts” site. To avoid manual maintenance, I also found an API to fetch categories defined in WP and filter them so that only the ones in use would display in the list. This added an asynchronous step to the template where it needed to fetch the categories before I could display the form to users. Yet after I gave it some thought, I figured I could do so in the background since the list would likely be fully retrieved before any user would have the time to click on the select anyway, reducing load time.

Carousel

One part of the requirements was to add a Carousel to the home page. Thankfully, I had made a carousel on the previous project and many of the challenges had already been figured out. Much of the code and principles could be reused, which was very time saving.

While there are probably a lot of way one could go about implementing a carousel, the main idea for me was to create a list of all elements that could be showed in the carousel. Then I calculated how many elements I would display, and which index in the list that should be showed in the leftmost space. When someone click the “next” button, I would then increase that index by the number of elements showed such that the user would see all new items, staring from the left. I decided to make it work in a circular fashion, so that one did not have to alternate between the control buttons.

In order to only show the correct number of elements in the DOM, wipe the carousel content and repopulate it only with the posts that should be showed to the user. Because I use “*beforeend*” to retrieve html for the posts, I get the posts in chronological order. This ensures that screen readers don’t read ahead, or that one could i.e., use tab to move along regardless of the carousel state. The advantage of this is that you wouldn’t have to click or listen to a high amount of post cards if you’re not that interested in this section. On the other hand, it was annoying that you now had to tab back every time you pressed a button that swiped the cards. To make more user-friendly for a user that relies on screen reader to get an overview of the new content in the carousel without having to tab back all the time, I therefore added an aria-live attribute to the carousel content so that it is read out without the user having to change focus away from the button.

Mostly for my own convince, I added a piece of code at the end of the carousel that listened to a window resize event and recalculated how many posts should be visible. The carousel would then redraw with the correct amount if necessary and helped me avoid the need to refresh a lot while I was working on it.

Modal

The modal was a technique that I thought would be a lot of work to implement yet proved quite straight-forward. In an earlier project, I had experimented with a shopping cart overlay. For this exam I could use the same concept with an overlay that spans the entire screen, where the modal is placed in the center. It turned out the JS-part in my implementation just added the HTML-element to the DOM and removed it again when it should be closed.



After listening to a podcast called “Universellt utformet”, I discovered that this version was a bit too simple. I therefore had to revisit it later to make sure that the user cannot use the keyboard to focus elements outside the modal window, nor let screen readers read the content behind the modal.

User feedback:

Having created a first version for a fully functional site, I now “switched mode” to where user feedback and insight in user behavior would drive my task priority. This section will describe how I collected information and some of the key findings.

Who and how

To test my blog site and ensure a user-friendly product, I’ve tested with different types of people and done both quantitative and qualitative methods to collect user data. I started with user testing the first week to test the design, and continued testing as I coded the page. As the site was getting more and more refined, I expanded my pool of test users to get some “fresh eyes” every now and then.

Even though I learned a lot, and did several changes based on insights about user behavior from Hotjar, user Interviews were my main source of valuable information once again. I started with 5 different users that could test the page while answering questions and tell me their honest opinion on different aspects of the site. While these initial users have tested the page several times during the whole process and given me good insights along the way, it has also been very valuable to test the page with new people to get some fresh eyes on the project.

Some of my user testing has been asynchronously, where I send the link and wait for a reply. After the initial reply, we’ve chatted back and forth with follow up questions or I’ve notified them when a new version is available.

There have also been a few users that I had physical access to and could observe while they used a computer or a mobile phone to test the site. I could then ask them to look up on something, or to fill in a contact form etc.

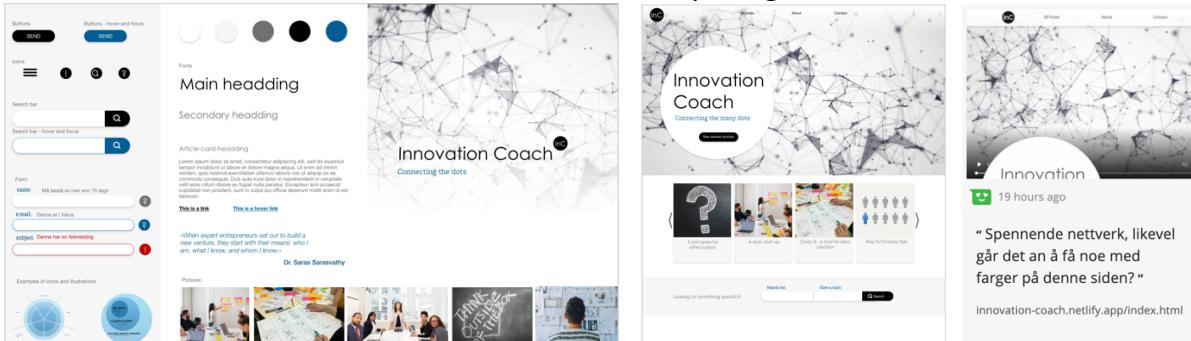
The last form for user interview has been to set up a meeting in Zoom, and let the user share their screen while conduction a test with their own equipment. One such interview with a blind test subject will be further detailed later in this report.

Design changes – Connecting the dots

As mentioned previously, I had users that got very strong associations with health care and band aids from my design. Every now and then when I got stuck with some task I would take a break and loot for more inspiration for new design. <https://meshcommunity.com/> came to be my main inspiration during these searches. I though the page was aesthetically beautiful and the animation / video on the front page gave an abstract illustration of development, process, and movement. After looking in the code for the page, (which is not very exemplary in terms of tidiness or accessibility) I saw that this might not be so difficult to achieve myself. Maybe this could be the extra creativity factor my website was missing?



When I found a video on Pixabay that could serve as a calm background and that also could be seen as an abstract way to illustrate some aspects of an entrepreneurial process, I went all in for a new design. The slogan changed along with the design once again and was now: "Helping you connect the dots", metaphor for seeing the whole picture and connecting different ideas and tools an entrepreneur should know about. A dot got to shape the logo as well, and I realized that I now had a dot-theme for my design.



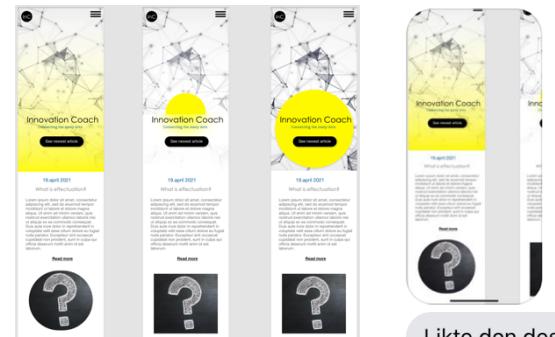
Style tile, design of home page and Hotjar- feedback on lack of color

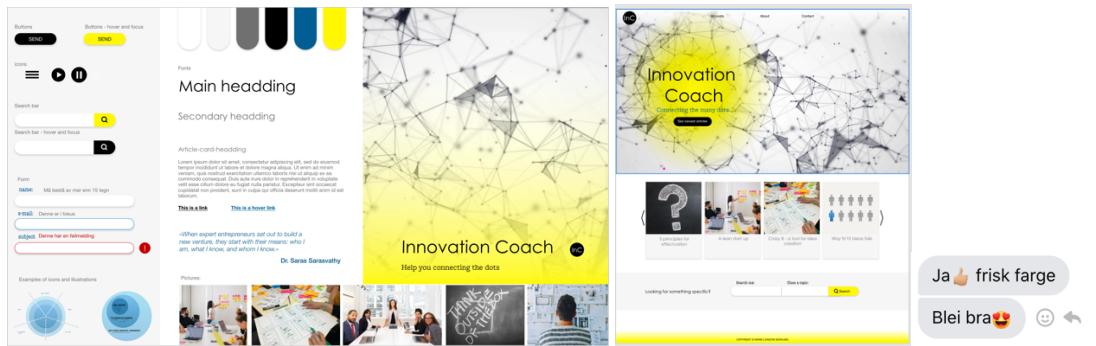
Colors

Even though the new design may be seen as more creative and innovative, users pointed out that it lacked colors. First, I had some version with different color on some headings. However, to get color into the design in the texts, I quickly found out that contrast requirements against a white background make it difficult to find a fresh color.

Therefore, I quickly switched to changing colors on different backgrounds such as buttons, the heading on the home page, the footer etc. One of the users mentioned post-it notes that are widely used and associated with creativity and agile methodologies. This gave me the desire to use a color that tended to neon and which then gave a fresh breath.

By working with both mobile and desktop design in parallel, I came up with a gradient to bring the color into the main page. The feedback on the yellow color was a bit mixed at first, as one of my nurse-users mentioned that yellow was the infectious color and that she, who currently works with covid patients, got a little itchy when she saw the color. A fellow student also mentioned that he was unsure of the yellow. He also did not like the video in the background. To accommodate these views, I tested various other colors and asked the test users which alternative they liked best. Based on feedback from several other users that liked the color very well, I chose to disregard these two views. The nurse is also in a unique situation and is not necessarily representative of her association. When it comes to my fellow student's desire for a still picture in the background, I must anyway make it possible to pause the video in terms of WCAG and should ideally start out with a still picture if the user is on a device where he or she specifically asks for reduced movement with their user settings.





Style tile, design of homepage and positive feedback on color-choice

Navigation and Layout

The only change in the navigation was I removed a page from my design called "Tools and links" which should contain tools such as various canvases and templates used a lot in the field, as well as some useful links to websites regarding certain topics. Yet since I received feedback that one did not necessarily know how to use the tools, and whether the search here should also include hits from this page or not, I decided to remove the page and rather treat the tools as topics in the posts.

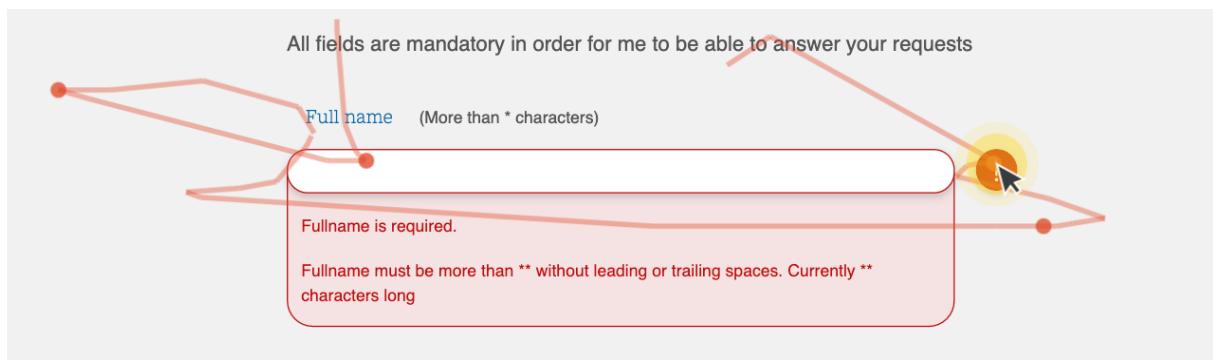
Loader

The loader also got designed as two dots with a line that moves back and forth and where the dots grow as they get connect. The loader is meant to be a small abstract illustration of the slogan on the homepage: "Helping you connecting the dots". The idea came first, and the loader is a typical example of where the author of this report severely overrated her own prowess with CSS as several hours went into understanding animations and making the animation work in a fluid manner.

HotJar

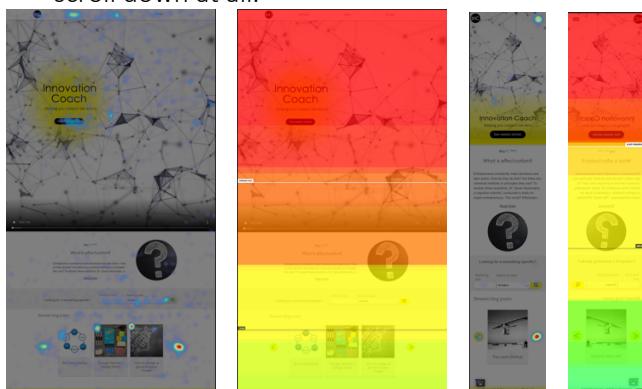
Hotjar is a great tool that provides surveys, heatmaps, and recordings to show how users interact with your site, what they click on, where they hover etc... But as with all quantitative surveys, they are most valuable with a large quantity of participants. With short amount of time to collect data, I published my page to fellow students, friends and family. While this gave me a lot of clicks and page views, I guess it also affected the data I got in the sense that they might not mirror the behavior of "real users that want to read about innovation and entrepreneurship". Naturally, the posts have a tough time to capture user interest when most of the text is lorem ipsum, yet it seems as some of my fellow students have been more curious as to how I have solved different aspects on the page, than to find and read an article. (i.e., some recordings showed users that went straight to the carousel clicking like mad, resized the window a lot, and did not really look at any other pages).

Nevertheless, I did get some useful feedback from Hotjar as I discovered that none of the users clicked on the "exclamation mark"-buttons in my contact form. After looking at the page more closely I saw that the buttons lacked the shadow-effect I had added to all the other buttons. I tried again with shadow and a more prominent hover effect and was rewarded the very next day when two Hotjar recordings showed users that clicked the buttons. In addition I had a test subject that day who tested for the very first time and said out loud, "Oh, now it popped up a button when I filled in incorrectly".



I also learned from Hotjar:

- Most users clicked on “All Post” in the menu as the next step after the index-page. Almost none took a short cut in to a specific post article even though they clicked on the buttons for the carousel.
- Surprisingly many users read the text in “about”. Also, they choose the section about me over the text where the site is in the focus, even though they would need to scroll down the page. I guess this might be related to the fact that my test users know me, and my students might be more curious in me than in my site, compared to “real users”.
- 68% scrolls down to the bottom of my home page on desktop. On a mobile phone 50% of my users does the same. While I think it’s great that so many scrolls down, this might also be because students are biased towards looking at the Carousel that they know should be on the home page. When I tested the end design with a new user, he didn’t realize that one could scroll down at all.



Findings and solutions

Video on iOS

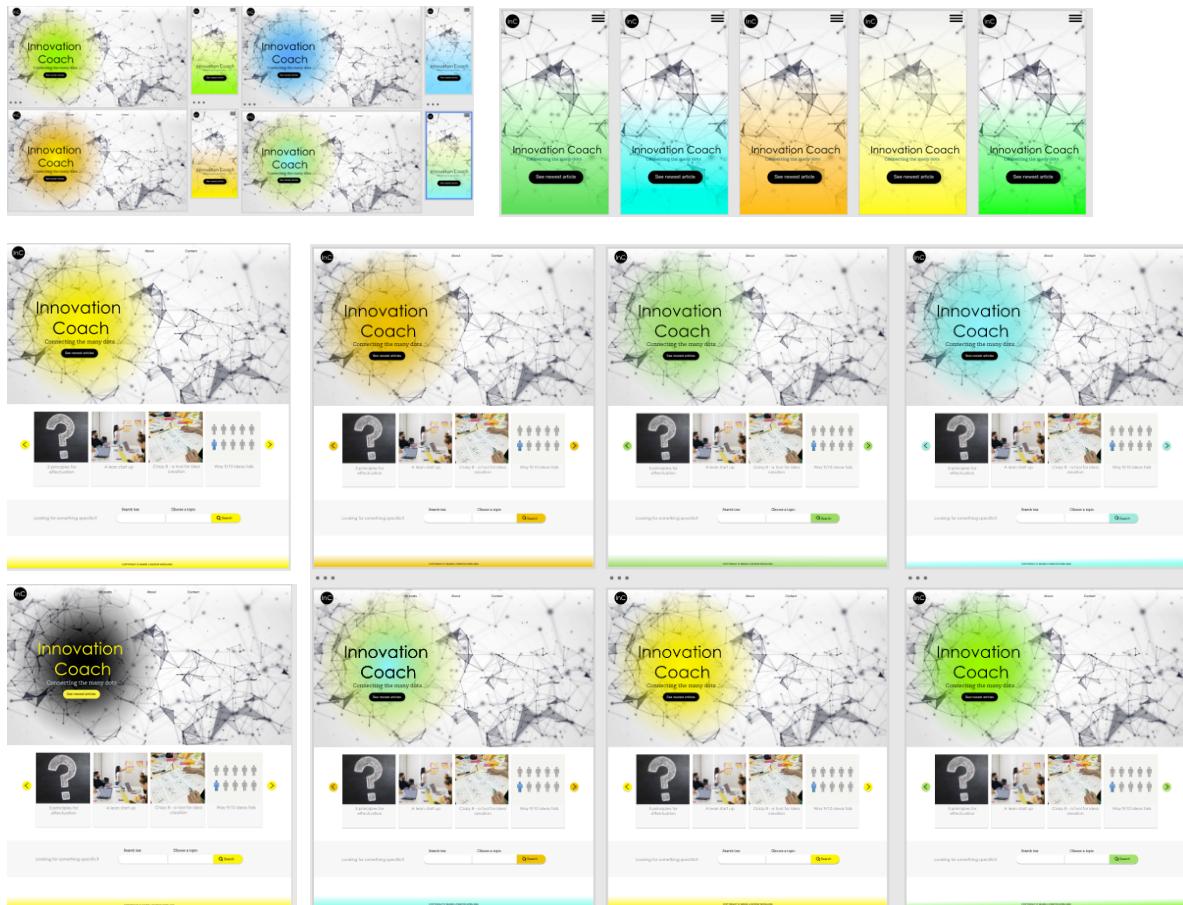
While it was surprisingly straightforward to add video capabilities to my site, it turned out that it was not as easy to get safari to automatically play the video when you opened the page. I had read through w3c description of the tag and attributes

(https://www.w3schools.com/tags/tag_video.asp), yet did not find anything. Also, the video was blown up to a full-size video when I clicked “play”. However, I went back to the site I got the inspiration from and found out that they had an attribute called “playsinline”, that solved this problem.

Colors

After testing many different colors on my users and seeing which options got the best response, I ended up with a bright yellow. With a gradient, it gave associations to the expression "a bright idea" and was described as fresh, brave and fun. The color is used on small surfaces on backgrounds to meet contrast requirements and is a small splash of color

that is repeated a bit here and there. I also had this color on the footer to create a splash of color at the very bottom, but this was removed when I received feedback from two users that a rather unimportant footer took a little too much attention despite the lack of content.



Testing different colors on the design

Full height frontpage

In my designs I always created the first part with video as taking the entire height. Yet testing proved that some users saw the site through Facebook browser, some had ios where there is a button menu when you navigate to a page the first time etc.

To account for these issues, I found out that I could style the page with “height: 100vh;” as I did initially, yet also use JS to overwrite this height to the number of pixels actually available by reading “window.innerHeight”. Thus the page finally took on the correct amount of space for different test users on mobile phones.

Contact form verbose

My contact form is built after reading from uu-tilsynet (Norwegian institute for accessibility) that it is a good idea to give a summary of all the errors in a form in a section located above the form itself, if any user tries to submit an invalid form. In addition, it is preferable to include an error message per input, that explains to the user what needs to be done to correct it. (https://www.uutilsynet.no/wcag-standarden/skjema/38#formidle_feil_i_skjema_1)



At the same time, it is also a good practice to prevent the user from making or submitting a mistake in the first place, so I have tried to follow a technique that was mentioned in w3c to give an error message when the user moves away from an input field:

<https://www.w3.org/WAI/tutorials/forms/notifications/>

Here I became acquainted with aria-live="assertive" (and aria-live="polite"), which I add to the error message on focus change so that a screen reader should read this aloud before reading anything else.

Early feedback from graphic users said that it felt a bit repetitive with visible error messages per field while they filled in, since the label instructed the user what to write anyway. I therefore made the text visually invisible until you clicked a button, while leaving it available to screen readers. This solved the problem for my users that complained about the amount of text, while it still could instruct people with visual impairments. Thus, the button is of no use for people who rely on screen readers and was consequently named: "Toggle visual error message."

Redesign DOM order (search in top, showing previous search)

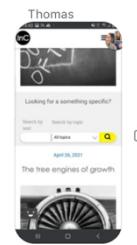
In my earlier designs I had the first post above the search form, before all the other posts came below. Feedback from one user that thoroughly tested the search functions was that it was confusing to place the search box secondly in the layout as he was unsure if the first post was a static thing or included because of his search. To make it more readable and intuitive, I moved the search box to the top of the “all posts” page

Empty state (no search hit)

Through user testing, it quickly emerged that I had completely forgotten to have an idea of what would meet the user if you did not get a hit on your search. A little embarrassing when I was asked for what I meant with the text “undefined”, yet I got great feedback on how I ended up telling the user that there was no result.

Don't repeat first post in carousel

The first post "What is effectuation" was shown both as a separate post at the very top as the first post in the carousel. This became somewhat repetitive, especially for small devices that only showed a single post card at a time. At the same time, there were several who liked that the very first post was above the search bar on the home page. Thus I let it stay on top and removed it from the carousel.



Vet ikke helt hva jeg synes å ha en artikkel over og under søkerlinjen. Ble litt forvirra om den på toppen var siste post og det under hørte til søkerfeltet, men så etterpå at alt hørte sammen



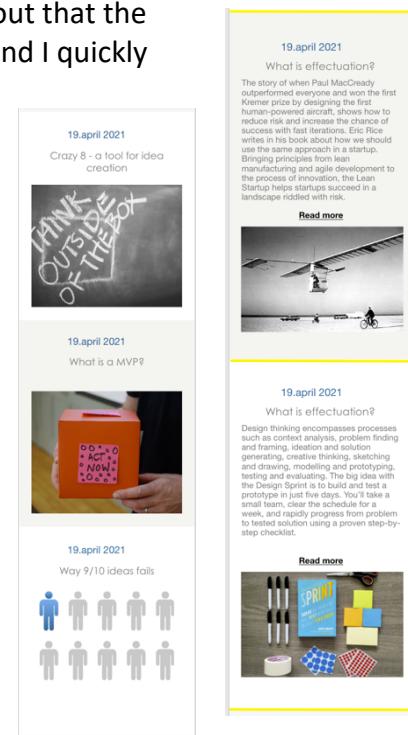
User feedback where he says the post above the search bar is confusing



Post summary on each post on top (user didn't find search keywords)

After a user tested the search function, I received feedback that it did not work. The user had searched for "invest" and then clicked on one of the articles that appeared. Then he had searched for the word in the text and not found it. It turned out that the word appeared in the summary text on the "all posts" page and I quickly realized that the same text should appear at the top of the article as an introduction. In addition, the user was surprised to see that the text appeared on the "all posts" page only on desktops and argued that it was much more interesting than the image. I therefore added summaries to all devices and noted for future reference that layout options might have been an interesting feature to investigate later. To better separate the posts on mobile phones, I also added a yellow line as a clear separator between each post.

Design of posts displayed in allPosts on mobile before and after iteration.



Accessibility

Why all the effort

As a front-end developer, I want to create good user experiences, and it is important to me that I do not exclude anyone. Accessibility is personally something that engages me as I want to contribute to an inclusive society. With this motivation in mind, simply making my site compliant to Norwegian law is more of a minimum requirement, as it is important to me that the user experience for someone who is i.e., blind, should not be worse than for someone who can see well, even if the experiences will be different. User-oriented design and development for persons with disabilities is exciting, and perhaps even more challenging as the user experience differs greatly from my own experience. Therefore, I decided I would use this project as an opportunity to gain greater user understanding within the topic of accessibility.

To learn what are best practices and how I can ensure good accessibility on my site, I have done the following:

- Talked to UU-tilsynet
- Read the UU-tilsynet's website for tips on good solutions
- Sent email to Blindeforbundet (society for people with sight impairments), in hopes of recruiting user testers
- Read at www.w3c.com
- Listened to the podcast "Universelt utformet" while I'm on my way to pick up at the kindergarten or while making dinner
- Read blogs about accessibility



- <https://inclusive-components.design/cards/?fbclid=IwAR2jWc4034p3ZZIVav69pPVMDzD1d-G4U55YwC85D8UPcLMmZh5oRPxzM9E>
- https://axesslab.com/alt-texts/?fbclid=IwAR0AWDbFT2RJP44W14mJNaaWiN3jUkbZ_F82KFktqGFFT_gUM6w5o_LCK58
- Engaged in a forum on facebook where I have used my page as a case for discussion about best practice
- Looked for new test users with disabilities

Testing:

Testing with users

User testing is necessary for websites to work well. People with different backgrounds and needs need to test the site to make it user friendly for everyone. Finding and conducting tests with new user groups has been both challenging, useful, and exciting. The accessibility authority recommends finding test users who represent several user groups with different challenges and needs. It can be:

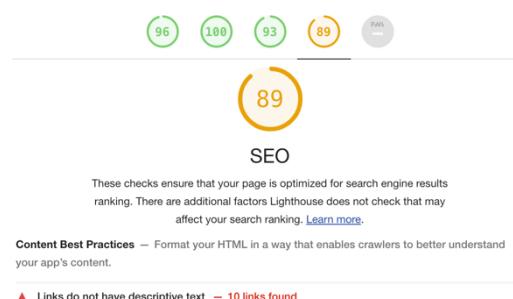
- blind
- visually impaired
- mobility impaired, especially disability in the arms
- people with reading and writing difficulties, such as dyslexia
- hearing impaired
- people with cognitive impairment or impaired perception

I am missing a good number of these users in my testing. But I have managed to recruit someone:

- one who is blind
- two with dyslexia
- one who never remembers her reading glasses

Testing with tools

To test the site, I have used Wave and Google lighthouse to get feedback on my code. I have done this regularly throughout the project and tried to pick up problems at an early stage. Mostly this has been helpful feedback, with follow up links on the topic where my code is found lacking. Yet I was a bit confused when Google Lighthouse's gave me penalty on my "read more" links. I can understand that if you read a list of links with a screen reader and the whole list consists of repeated "read more" it would be a bad user experience, but since I added an aria label with a full description, I felt it should suffice. This is mentioned as a suggestion for a good solution in such cases at www.w3.org. My score for SEO therefore contains a penalty, yet I've added an issue to Google Lighthouse as seen below:





MarieLowzowMorland commented 2 days ago

...

From WCAG, link texts are covered in this article, that describes some of the motivation behind demanding descriptive links.
<https://www.w3.org/WAI/WCAG21/Understanding/link-purpose-in-context.html>

One of the techniques mentioned here for passing is to use i.e aria-label to provide a good description in the cases where the author of the page for some reason wants to use a common repeating text such as "read more":

<https://www.w3.org/WAI/WCAG21/Techniques/aria/ARIA8>

However, Lighthouse has some code that blacklists common terms in link texts regardless of the what the accessible name provided with i.e aria-label says, in stark contrast to the accessible guidelines above:

```
lighthouse/lighthouse-core/audits/seo/link-text.js  
Line 127 in 545176a  
127     return BLOCKLIST.has(link.text.trim().toLowerCase());
```

Shouldn't a SEO use the accessible name over the text for anchor elements?

If so, then the lighthouse check should also use check for an accessible name before adding a penalty here

I have also used the keyboard and the screen reader VoiceOver to test how the page is works for those who rely on screen readers. I must admit that I struggled a bit to navigate and even though I learned a few shortcuts, I'm far from a native user. Nevertheless, it has been instructive and useful. Among other things, there is a lot of unnecessary repetition when you constantly must get through the menu-list every time you go to a page. While my site does not have the longest menu, it is still annoying. However, I learned about concept of skip links though the podcast I listened to, got excited and made one myself.

Engaging a Facebook forum

To get feedback on Accessibility, I posted my page on a forum for accessibility enthusiasts on Facebook. In less than ten minutes I had several comments from people who took their time to test my page, and it resulted in an exciting discussion about best practices on various topics with my project as a case. Here I learned a lot from experts and got several good tips for improvements that were not picked up by tools like Wave and Google Lighthouse.

I also got a comment from a blind person who said he'd investigate the site this final weekend, and while it's a bit late for this report, I'm looking forward to his judgment. All in all, I got a lot of positive remarks about how accessible the site was, and that what they came up with was mostly small picks. (I posted it on Facebook less than a week before submission, so I had done quite a lot of digging and work already). One of the participants put it this way: *"I would say that your page is accessible, but that it can probably be smoother."* I take that as a big compliment and an encouragement to continue my engagement for accessibility.

Findings and solutions

Skip links

The concept is to create shortcuts down the page for someone who uses tabs to navigate. Just like you can press "skip intro" on a series on Netflix, this button can skip repeating or optional content such as a list of menu options. Since people who navigate with a mouse does not need these links, the anchor tag only becomes visible if you use keyboard to focus



School of technology and digital media

it. This allows you to skip i.e., straight to the main content, or (for those pages where it is relevant) to the search bar.

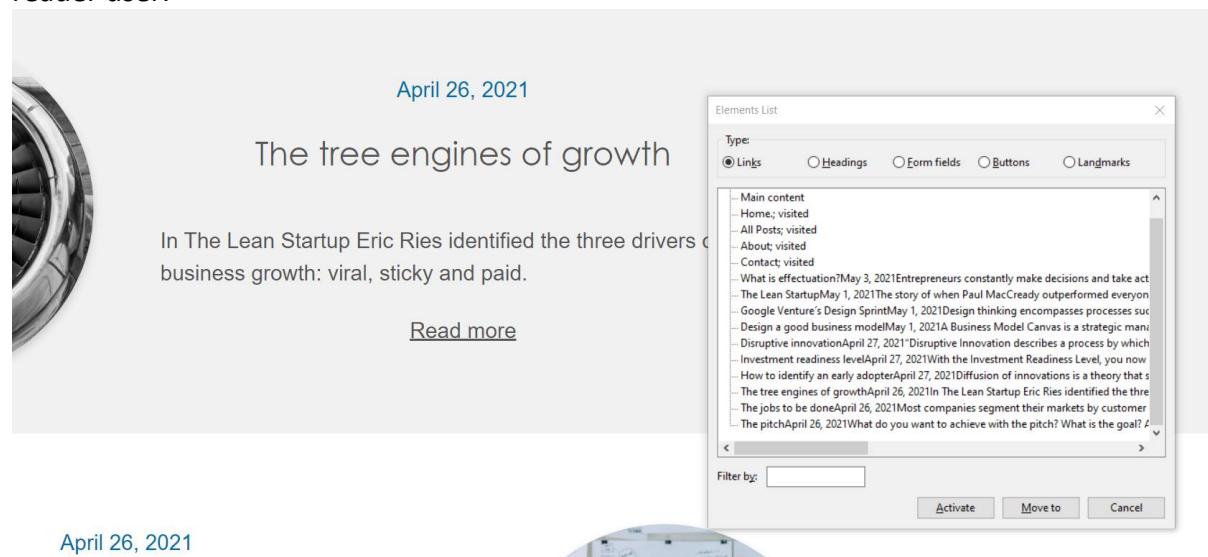
Video with autoplay

Some people have issues coping with fast animations, or just a lot of movement. Also, people with i.e., dyslexia or a concentration problem might feel that there the text is more than enough to pay attention to. It is therefore not only required that one can pause/play video content. While I felt that a play / pause button does not fit into the design visually, I've added the ability to tap or click the video to pause it. User testing shows that users find this feature easily even if it is not very visible, and for those who navigating with a keyboard, a play/pause button pops up when you focus the banner.

Some people use an option in their laptop or mobile to say that they want reduced movement. For example, there may be people with crystal melanoma, difficulty concentrating or vestibular motion disorders who do not tolerate movement and animations very well. The prefers-reduced-motion CSS media feature is used to detect if the user has requested that the system minimize the amount of non-essential motion it uses. (<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/prefers-reduced-motion>). For my part, I therefore implemented a check in JS for this media query to turn off the automatic animations for people who use this setting. The loading animation is on the other hand left intact as it is neither fast nor big, and conveys a meaning.

Long anchor-tags

Before, I had the whole post as a link in the All-post page. From the Facebook forum, I received feedback that the link was very long for a screen reader and that it became difficult to get an overview of the topics when so much was to be read. "Read more" was probably enough as a link but should have an aria label such as. "Read more about Google Venture's Design Sprint", so that the various link can be distinguished from each other for a screen reader user.



Colors

WCAG 2.1 requires that non-textual content has a contrast ratio of at least 3:1 to adjacent color (s). The cursor that shows which page is active on the page can be difficult for people with impaired vision to spot. WCAG 2.1 contains future requirements that do not apply today, a recommendation is nevertheless to start getting acquainted with what is to come. I did like to have the cursor bright yellow, but to meet upcoming requirements, I have changed the color to black to match the loader and homepage video.

Aria-current

I got a tip on Facebook that aria-current = page be nice to have on the cursor of the current active page. Aria-current tells the user that there is a cursor here that tells you what page you are on.

Aria-labels

Aria-labels is used many places on the project:

- On some buttons
- On input fields
- On the skip link
- On links

For example: I have added an aria label to the "read more" links on the "all posts" page so that a screen reader, for example, will read "read more about The Lean Startup" in the article about Lean startup and Eric Rice.

Aria-live (carousel, contact form and user notification)

Aria-live can be used to give screen readers a hint that the section should be read out loud. As mentioned, the carousel will read the new posts out loud if you click "next" or "previous". Also, both new error messages and the user notification use this attribute to be read out loud as well. This usage makes sure that the user will get notified when things happen.

Menu close after list (not implemented yet)

I got feedback that it was tricky to close the hamburger menu with screen reader, because the user had to navigate back through the menu to close it. Therefore, it was recommended to make a close button at the end of the menu as well.

HTML structure

I have tried to be aware of the order in which the screen reader reads the. For instance, I coded the date for each post after heading tag but used CSS to place it visually in the opposite order. I thought the same way about the carousel's post-cards where I first made sure that Voiceover read out the headlines in the post-card before the picture is presented.

Alt-text

At first I made sure that every image had an alt-text. But then I read a blog written by a blind man about how he prefers alt-text to be used, and that it can be experienced as annoying in some cases. Listing of articles in a blog was explicitly mentioned as an example. Since the



pictures are first and foremost decorative in the lists, I therefore chose to go for a blank alt in the carousel cards and “all posts” but describe the pictures when you are in an article.

On another hand: As I have discussed the issue with others one someone said: “what if someone talking about the page are telling you to find that blog post with a picture of an old plane?” I guess the best solution is to consult with real users although I have a feeling that their opinions are also divided. One solution is to go back to have alt-text, yet I would instead consider a solution that might be beneficial all users: Just like one can change layout in a folder between a list of files with data columns, or the graphical version with names underneath, one could image to display modes for the articles:

- One with pictures
- One with just headers and text

In a user test, it was mentioned that the images might be of less interest, especially when searching for a specific topic, (screenshot on the right). As a possible solution, he suggested having different list views. In a blog with articles such as this one, the written content is much more important than the picture. (As opposed to how I would feel about i.e., a food blog)

Hvordan er det med listevisning og slikt? Er det litt utenfor kravene? Synes det kan være greit på mobil av og til. Altså uten bilde kun tittel med en ingress. Når en søker og slikt. Som et alternativ

Er jo greit hvis dette var en blogg med mange innlegg

Taking this one step further, the mode you elected could be remembered on the browser using i.e., localstorage. Then, the list version would omit the images and consequently the alt texts anyway, while they would be included in the display mode with images.

Headings

I have also learned to be aware of my headings. On one of the previous designs, I had “innovation coach” as a kind of visual and decorative headline in the header. Almost as an alternative to a logo. But for someone who uses a screen reader, it will be terribly boring to read if the main headline on all pages was the same. The thought of reading a book where all the chapters had the same headline sounds ridiculous, impractical when it comes to navigation, and uninspiring. Although this did not become a relevant issue at the final design, the experience has made me aware of making exciting headlines – also for a screen reader.

Custom focus

From the accessibility forum on Facebook, I got a tip regarding defining a custom focus in the CSS code. For once I had not implemented my own because I thought maybe people liked to see their “usual” focus style. Yet at this person pointed out, the default is not always a particularly good match for the design. In addition, while Chrome on Mac has a clearly visible blue outline as default, it was much less prominent on Windows. I thus learned that best practice is to define your own focus in order to be in control of contrast conditions and presentation across browsers. As someone commented about Chrome on Windows: *“Chrome's default focus would have failed WCAG if not for the fact that native focus is explicitly excepted from the rules about focus contrast.”*



Last round of feedback:

Some new fresh eyes on the site

The last week, I recruited brand new participants for user testing. In total, I have had 17 people who have tested the site and given me feedback, without counting feedback from students. (They are a slightly special user group with abnormal user behaviour)

Standing behind one of the new users while he was testing the site, gave me new insight that would be hard to get through a chat on internet. For instance, I saw that the user did not scroll down while he was at the homepage. He was not used to sitting on a mac, and thus he did not ever try to scroll as he didn't see any visual scrollbars. The solution I went for was to change the CTA button and add an arrow to indicate that the button would take you further down the page and that it was more to see.

He also noted that he thought he'd use the search a lot if the blog had been real. While he was pleased with how it worked, the comment made me include a skip link directly to the search bar for applicable pages.

During this last round of feedback, I have also been asked if the page should perhaps have a comment field under the articles, seeing as these types of articles might be particularly well suited to start a subject discussion that could later have inspired new blog posts. Also, a comment section where one could add anonymous comments thought of as easier to do than to fill in a contact form with a question. While I did not get around to building a comment section, it is noted as a sensible improvement.

Test with a blind user

While I tested the site multiple times with a screen reader myself, I figured that it was hard for me to determine if the site was well suited since I was so new to screen readers in general. I therefore got in touch with a blind acquaintance of mine who agreed to become a test subject and rejoiced that I was interested in the topic at all. Due to illness, this test was postponed and was not completed until the day before the submission.

Gaining the experience of setting up such a user test and gaining insight into her use of websites was incredibly valuable. The test was done online with Zoom where we could talk, and I could see her screen while we conducted the test. It did not work as well as intended, since the screen reader sometimes removed the visual focus on the screen, and sometimes it did not scroll in the same way as it does when I use my own browsers. Still, it was convenient to at least see which page she was on and what happened as she filled out the contact form. She also explained well out loud what she heard and how she thought. Also, she told me how she used the screen reader, what other tools she used on the side and a little about her own skills versus what is possible and what you can expect from someone who uses screen reader and braille terminal in everyday life.

When she entered the site, the skip link was the first thing she discovered, and she was very thrilled about it. Soon after, she said it did not work, until she again came up with an "Aha! There is a list below here!". Even though it was on a nav, the screen reader had read out loud that it was an element called "Skip to content" and she had thought it was a link. I had



originally thought one would only try to activate buttons or anchor tags, yet this led me to change the aria label on the nav element to "Skip link Menu". All in all, she was very happy with the skip links and commented on the new pages that it was great to be able to jump straight to main content.

I asked her about her preferences around alt-text in the pictures on the all-posts page as this has been a topic on the Facebook forum and I was unsure of what would be the best way to solve it. It was clear she was not very spoiled with good alt-texts in pictures, and just the fact that there was a thought about it, was more than enough for her. She felt it was nice and effective without alt-text on the all-posts page and did not feel any need for more explanations of what pictures were shown there if it appeared on the specific post. When I put forward the argument that someone talks about "the post with a big plane"? she laughed and thought I was over-complicating it a bit. She said that it was up to the person she talked to give a clearer description of the content of the article. After all, it was the content that was important, and she would just use the search bar to find the right post instead navigating around looking for the correct alt texts. She also found it funny that I had picked up a detail about using a period at the end of an alt text in order to provoke a small pause in the screen reader.

When it came to alt texts in general, she suggested nrk.no as a good role model because they are short and informative. She also explained that she had two braille terminals. A long one that she had lying at home with room for 80 characters, and a shorter one of 40 characters that she could carry around and connect to more mobile devices such as mobile phone and laptop at the university. The braille terminal with 40 characters made her very much appreciate if alt texts and other short messages stayed within 40 characters as she then did not have to read multiple lines on the braille terminal.

When it came to navigating the site, she orientated herself smoothly and said that the page was neat and that it was easy to get to know the page and understand the navigation. On the "all posts" page, she quickly went through all the headlines and read a few summaries. She explained that she often used headlines to orient herself, and that those who were a little more adept with screen readers could get lists of links, headings, and other functionality to navigate effectively.

The search function was understandable and useful. She also thought it was nice with the page that told that there was no article with the topic she was looking for. She particularly liked the alt text for "sad face" which she described as "cozy". One thing she mentioned as a potential for improvement was that H1 in this case could have been changed from all posts to "No results" or similar. Since I have made this H1 a screen reader only, I had forgotten this detail as I didn't interact with it myself.

When we came to the contact form, she became very enthusiastic, as she had filled in many less accessible forms before. I deliberately asked her to write something wrong to test the error messages. Aria-live informed her of the error and she thought it worked very well. When she deliberately tried to submit the form with errors, she expressed: "Wow! This works really well! Here I get listed the errors and it takes me directly to where I have written errors! Not all forms work so well! ". Yet my bloated ego came to an abrupt halt when she pressed the "send" button and became a little unsure. I turned out her screen reader did



not get the user message displayed on the screen. The solution I went for was to add role = dialogue and aria-live=assertive, which again forces the screen reader to read the text out loud. Another solution would be to modal window, that she also had complimented earlier.

All in all, she was very happy with the site. Her highlights were that accessibility was clearly thought of, and that the site was easy to navigate. She met no obstacles and accomplished everything she wanted. The user message about the confirmed sent message was perhaps what she missed the most but emphasized that she still understood that it had probably been sent when she discovered that the form was emptied. The form otherwise was a highlight by itself.

Discussion

During this project I have had several discussions with myself, users, and others on a lot of issues. This section describes a few:

What is innovative design on a webpage?

When creating a page about innovation, one quickly discovers that there are expectations around the design. Users have expressed a desire to see something they have never seen before, something exciting and creative that also comes with a lot of color. At the same time, the page should be clear and easy to navigate to be user-friendly. To achieve that, it is easy to apply common patterns and norms for layout so that the user almost recognizes themselves and immediately knows how to use the page. In many ways I would say that the design on my page is not very innovative at all, but rather very conservative with a layout that is easily recognizable in terms of how you navigate around. Instead, I have tried to give the site an aesthetic and a mood that gives associations to a creativity, development, innovation, and bright ideas. With an abstract video with movements as a picture of an innovation process, and a bright yellow gradient, I have tried to create an element that users have not seen before and which may surprise a little. Whether I have succeeded in creating an innovative design, I'm not sure, and there may not be a final decision on what is good enough to be able to get such a label.

Page headers

While I had the "band aid" design on the page, it was tempting to put innovation coach as an H1 on all pages as this visually looked like a headline. But in terms of accessibility, this is a hopeless way of dealing with headlines. As many screen reader users navigate by headings, similar headings on all pages could be compared to trying to navigate in a textbook where all the chapters had the same heading. In the same way as a book, the headlines should provide an overview and make it exciting to read. What is best? Should the header one the contact page be "Feedback or coffee?", "Contact", "Tell me what you think?". In the end, I feel like the most important part is that structure makes sense. I used Wave to get an easy overview over how the landmark and headers looked like in a compressed manner.

On the "all posts" page, there was also no natural H1 headline visually, so I made a "screen-reader-only" class that visually hides the H1 tag using CSS. This means that someone who uses a screen reader, or a Google bot could still read that the H1 was "All posts". Even though I have understood along the way that



The screenshot shows the Wave interface with the 'Structure' tab selected. The page structure is outlined as follows:

- Header (with three navigation items)
- Main
 - All posts (H1)
 - What is effectuation? (H2)
 - The Lean Startup (H2)
 - Google Venture's Design Sprint (H2)
 - Design a good business model (H2)
 - Disruptive innovation (H2)
 - Investment readiness level (H2)
 - How to identify an early adopter (H2)
 - The three engines of growth (H2)
 - The jobs to be done (H2)
 - The pitch (H2)
- Footer



headings isn't something that should be underestimated, I see that I have not managed to remember all my headlines in all situations. For example, it turned out that the screen reader notes "all posts" as H1 for the search results. This seems especially strange the times you do not get hits on the search and there are no posts on the page, such as was the case in my user test with a blind person.

When and where to split a function

While I made the header in a certain manner for the design with the band aids, I had a lot of text on all pages for the desktop mode, while on mobile it was only the main page that had a banner. When I experimented while creating the existing layout, I felt really complex to make this work in a clean fashion. In the end I decided the header template could rather start with a split based on whether it was the homepage or not.

Seeing as how the result looks like, I'm quite pleased with how it looks, and that I split the CSS for the home page out to index.css. Yet it seems a bit fragile all the same, in the sense that a new layout probably would mean that code needs to be moved again in several different places. I guess that like with so many other of my struggles, it is one of those decisions that comes easier with experience.

When to make a function generic

Several times during the project I have been in a situation where I had to choose whether I should tailor a solution or whether I should create a function generically so that it can be used several times on a page. The advantage of making the function reusable is that you can save time if you need the function later (for example the search bar), or that you can reuse it in another project later, like I did with the first version of the form validation. On the other hand, you may have to make it a little more complex to pull it out of context and make it reusable. At the end it is not certain that it pays off if it turns out to be used only once. Then one can ask oneself if it was worth it and if you should have simplified the code for better readability. When I started coding the carousel, the plan was not to have more carousels than one, and thus it is also probably not entirely generic. What if user feedback shows that another was needed? Or what if I need another form on contact.html later so that it can be possible to register an interest in a workshop or similar? In general, it is difficult if not impossible see the future and know whether you should make the code reusable or not. The cheapest advise would probably be to make it generic when you see the need, and not to do it up front.

CSS selectors

While not as often as in the beginning of my studies, I still find myself falling over self-made traps in CSS. In this assignment, I originally had a single nav-tag, and consequently used this as a selector instead of naming it with an id or a class. Problems arise when I added a new nav for skip links, and the styling and JS functionality broke. I make a mental note to myself that I should at least name it when I also has JS functionality that ties up to the element, yet it is still hard to distinguish when I can style based on structure versus the need for classes or IDs.

Functionality vs accessibility

The assignment has some optional tasks, users request features and accessibility demands a lot of research and thought. Deciding which feedbacks and ideas one should prioritize is an ongoing process, yet for this exam I felt made the list mentioned earlier:



1. Do all mandatory requirements
2. Develop the site user-centric to ensure a good experience
3. Make the site accessible to everyone, and try to get conduct a good user test with a person with some sort of disability
4. Expand to optional functionality

While it seemed rather straightforward, issues quickly emerged: Not all user feedback is something to act on right away. Users may suggest functionality or design changes from the top of their head, even though it might not necessarily be an improvement. When I asked a lot of people what color they would've chosen from a palette I'd made, I got 4 different answers from the 7 first people I asked. Which indicates that one should take care to verify a change before going through with it.

Also, I sometimes asked the user if they felt like a specific new feature would be an improvement. Usually, they would agree, even if they had not themselves complained about the form, the search bar or on a design decision. Thus, it seems most users doesn't necessarily think that something is lacking until it is pointed out.

Therefore, I felt I deviated a bit from my original plan in that I actively sought user feedback, yet unless I could verify and change it quickly, I would rather prioritize making the existing functionality accessible. I feel that something not accessible is only a prototype, and not a fully finished piece of functionality. Therefore, I could experiment and add new functionality for user testing that wasn't compliant, yet I would rather have a site where everything worked properly, than to add a few half-done pieces.

Even so, I got to add most of the optional functionality like search, filter and sending contact form. If this had been a real-life project, I might not have had the luxury to decide on my own, so that makes me extra happy to be able to prioritize accessibility this time around.

Future work

Although the blog has come a long way, there are several ideas and plans that were scoped out, and the blog site still has a lot of potential for improvement. My preliminary list of things to do says:

- Menu close button after listing the menu items
- Change h1 when searching for specific topics
- Add loader when fetching more posts (forgotten and never mentioned)
- A "try again later" message if WP doesn't answer. (Never had any issues and therefore not prioritizes it)
- Next / previous post buttons on each post page
- Comments on post page
- Alternative «all posts» layout, with local storage to remember what you'd like (pictures or not)

Conclusion

All in all, I would say that the goal "to put into practice the skills learned over your first year of studies" has been achieved in many ways. In addition, the project has led to new



knowledge, especially within the field of accessibility, where I've had extra attention during this project. This focus has been at the expense of time and capacity to complete all the functions and tasks that were listed as optional level 2 tasks. Even so, I'm quite happy that I've felt my site was appreciated someone with sight impairments, and that I've now learned so much about a topic that is required almost for every website I'll make.

Key learnings

User Testing

Once again, I sit with the impression that a user-friendly site needs a lot of user feedback to be successful. It seems impossible to think of all the improvement yourself, and the more user feedback you get, the better. It has been challenging to find users with various disabilities to test the site. In retrospect, this is something I would have liked to have started with earlier. Especially since unforeseen things like illness stalled one of the main user tests happen a little late in the process. Although it was too late for major changes, I was happy it turned out that none were needed. The experience was also immensely valuable for me as I both could observe and talked to someone with a quite different take one websites that I have myself.

I've tried several different forms for user testing, my experience is that you get the best result when you invite users to co-create the site with you, and when you can observe how the user interacts with the site. Now and then it is also good to get some fresh eyes on the project, so it very valuable to hold back some user, and not involve everyone right away. In this way, you get to see the reactions of someone who is not biased towards the changes you made based on earlier feedback

Technical solutions

I think I have found a good way to structure code and folders in Visual studio. Although there I ended up with a lot of code lines, I find my system to make it easy for me to find the code I'm looking for. On the other hand, there is always a discussion about whether something should have its own file, or whether it should be coded on the page-specific page.

The fact that I had already worked with similar solutions to some of the functional challenges in previous assignments, resulted in an ability to get a lot of functionality out fast. It is comforting and encouraging that experience helps. Even so, I still have made a lot of adjustments to these parts and feel that I now have a form validation that I will have much use of also in future work.

A good development environment makes it much easier to work efficiently. Plugins in Visual Studio such as Live server and es6-string-html template have helped a lot. Also, I've learned to connect my mobile directly to the computer to get live testing here as well.

Accessibility

I'm mostly struck by how much there is to learn about in accessibility! Also, both users and experts disagree with each other on what is the best solutions, meaning that there is a lot of thought process one must do to decide what is the best solution for a particular site.



I want to highlight what in many ways was the conclusion of our discussion on Facebook:
"WCAG compliance is often too little / too poor. I'm really looking forward to the accessibility declaration becoming standard." This is because then people are forced to actually argue and think through what they do regarding accessibility.

Tools like Google lighthouse and Wave are great for a lot of accessibility testing, yet they don't test everything on WCAG. Regardless, I have experienced that WCAG as a checklist is not enough to ensure a good user experience for everyone. A site with good accessibility and a site with good WCAG compliance are not necessarily the same. As another user said: *"One must look up and aim for accessibility rather than the pursuit of compliance"*



Reference:

Accessibility

<https://www.uutilsynet.no/webdirektivet-wad/wcag-21-standarden/140>

<https://www.uutilsynet.no/wcag-standarden/losningsforslag-nettsider/36>

<https://www.uutilsynet.no/regelverk/brukertesting-med-funksjonshemmede/193>

https://axesslab.com/alt-texts/?fbclid=IwAR0AWDbFT2RJP44W14mJNaaWiN3jUkbZ_F82KFktqGFFT_gUM6w5o_LCK58

<https://developer.mozilla.org/en-US/docs/Web/CSS/@media/prefers-reduced-motion>

<https://inclusive-components.design/cards/?fbclid=IwAR2jWc4034p3ZZIVav69pPVMDzD1d-G4U55YwC85D8UPcLMmZh5oRPxzM9E>

<https://www.w3.org/TR/wai-aria-practices/examples/menu-button/menu-button-links.html>

<https://www.uutilsynet.no/wcag-standarden/skjema/38fbclid=IwAR0G5EhFdd1svhKcRt6aEwCU6wmX8w9P27zctYxaBZ8A550nRNOFYXXIJqo>

<https://www.w3.org/WAI/tutorials/forms/notifications/>

<https://stackoverflow.com/questions/40555111/what-is-the-best-way-to-use-a-background-image-on-a-div-yet-remain-accessible>

<https://stackoverflow.com/questions/52783375/anchor-with-href-not-taking-focus-in-safari-even-with-all-accessibility-keyboard>

https://www.webassign.net/manual/student_guide/t_a_osx_tab_config.htm

<https://developer.mozilla.org/en-US/docs/Web/API/Window/innerHeight>

https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA_Techniques/Using_the_aria-hidden_attribute

https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA_Live_Regions

<https://www.aditus.io/aria/aria-current/>



Bugs and function

<https://stackoverflow.com/questions/41512200/how-do-some-websites-play-video-inline-in-ios-safari> (safari på mobil ville ikke autospill eller la file være i line uten)

<https://help.hotjar.com/hc/en-us/articles/360046960654-Why-does-my-Heatmap-screenshot-look-cut-off-the-wrong-size->

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/entries (hvordan gå gjennom et objekt)

<https://developer.mozilla.org/en-US/docs/Web/API/Node/insertBefore>

<https://uxdesign.cc/how-to-trap-focus-inside-modal-to-make-it-ada-compliant-6a50f9a70700>

<https://stackoverflow.com/questions/50178419/how-can-restrict-the-tab-key-press-only-within-the-modal-popup-when-its-open>

https://medium.com/@mahesh_joshi/wordpress-contact-form-7-rest-api-endpoints-bf45907b571c

<https://stackoverflow.com/questions/21044798/how-to-use-formdata-for-ajax-file-upload>

<https://since1979.dev/respecting-prefers-reduced-motion-with-javascript-and-react/>

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/shift

<https://css-tricks.com/bold-on-hover-without-the-layout-shift/>

Det var utfordrende å få nav bar til å stå stille når hover ble bold. Lagde en kopi av teksten i bold som var hidden slik at plassen alt var satt

Sticky fungerer ikke slik som jeg vil:

<https://developer.mozilla.org/en-US/docs/Web/CSS/position>

https://www.w3schools.com/howto/howto_js_navbar_sticky.asp

<https://css-tricks.com/line-clampin/>

<https://contactform7.com/getting-started-with-contact-form-7/>

Pictures:

<https://unsplash.com/photos/onLbXleikds>

Photo by [Isaac Benhesed](#) on [Unsplash](#)

<https://mycolor.space/gradient?ori=to+bottom&hex=%2354A4EA&hex2=%23006093&sub=>



<https://unsplash.com/photos/Cp-b69OOyN4>

Photo by [Jo Szczepanska](#) on [Unsplash](#)

<https://unsplash.com/photos/Oalh2MojUuk>

Photo by [Leon](#) on [Unsplash](#)

<https://unsplash.com/photos/wusOJ-2uY6w>

Photo by [Amélie Mourichon](#) on [Unsplash](#)

<https://unsplash.com/photos/lZ01rjX0XQA>

Photo by [Thomas Vimare](#) on [Unsplash](#)

<https://unsplash.com/photos/qC2n6RQU4Vw>

Photo by [UX Indonesia](#) on [Unsplash](#)

<https://www.pexels.com/nb-no/bilde/sprak-sitat-kritt-innovasjon-6375/>

<https://www.pexels.com/nb-no/bilde/avgifter-euro-fond-gris-9660/>

<https://www.pexels.com/nb-no/bilde/samarbeid-forhold-tilkobling-forbindelse-3184433/>

<https://www.pexels.com/nb-no/bilde/handtrykk-samarbeid-tilkobling-forbindelse-3184423/>

<https://www.pexels.com/nb-no/bilde/forvirret-rom-test-forskning-212286/>

<https://www.pexels.com/nb-no/bilde/flyplass-fly-luftfart-motor-459402/>

<https://www.pexels.com/nb-no/bilde/sporsmal-hjelp-nysgjerrig-symbol-356079/>

<https://www.pexels.com/nb-no/bilde/inspirasjon-teknologi-lyspaere-ide-3758105/>

Video:

<https://pixabay.com/no/videos/nettverk-koble-internett-abstrakt-45961/>



School of technology and digital media