Henrik Johan Olsen

# The development of NORDIF's business website

Bachelor's thesis in Information Processing
Supervisor: Atle Nes
January 2022

Bachelor's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering

Henrik Johan Olsen

# The development of NORDIF's business website

Bachelor’s thesis in Information Processing
Supervisor: Atle Nes
January 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering

NTNU
Kunnskap for en bedre verden

# Abstract

This report expands upon the project "Making a business website for Nordif".
It will cover my experiences working on the project in its various facets.
From the first meeting to the final one, and everything in between. All the processes I have been through; the ideation, the research, developing the design, developing the code, and finally, actually writing about it.

# Sammendrag

I denne rapporten kommer til jeg til å utdype og forklare alle mine opplevelser rundt å jobbe med prosjektet «Lage en bedriftsnettside for Nordif». Rapporten vil dekke hele oppløpet, fra første møte til det siste, og alt imellom. Alle de forskjellige prosessene jeg gikk gjennom; ideskaping, forskning, utvikling av designet, utvikling av koden, og til slutt, å endelig skrive om det hele.

# Foreword

A quick thanks to tea and energy drinks. Thanks for providing me with caffeine, both hot and cold depending on my preference at the time.

Thanks to my friends. Who has been my mental and emotional support for years.

Thanks to my family. Who I knew always had my back and could catch me in the safety net should I need it.

And finally, a little add-on, thanks to all my fellow volunteers at Studentersamfundet. For giving me many new worthwhile memories, and making me forget others.

# Table of contents

# Figures

# Acronyms and Abbreviations

| | |
|---|---|
| API | Application Programming Interface |
| BEM | Block, Element, Modifier |
| CAPTCHA | Completely Automated Public Turing test to tell Computers and Humans Apart |
| CSS | Cascading Style Sheet |
| CTA | Call To Action |
| GIMP | GNU Image Manipulation Program |
| GNU | GNU's Not Unix |
| HTML | HyperText Markup Language |
| IDE | Integrated Development Environment |
| JS | JavaScript |
| MUI | Material UI |
| NPM | Node.js Package Manager |
| PDF | Portable Document Format |
| PHP | Personal Home Page (former), PHP: Hypertext Preprocessor (current) |
| SVG | Scalable Vector Graphics |
| SQL | Structured Query Language |
| UI | User Interface |
| UX | User Experience |
| VSC | Visual Studio Code |

# 1 Introduction

In this project, my task was to create a website for Nordif. A company under Jarle Hjelen A.S. Whose main products are EBSD detectors and related services. We quickly established some expectations during our first meeting and opened a dialogue regarding future developments.
The initial specification provided about the website were as follows:

- Userfriendly
- Functional on all devices
- Provides good search results
- Advertises our products and services
- Generates leads
- Establishes contact with customers
- Includes information regarding specific products and services
- We, the business, should easily be able to update the website ourself

These seemed reasonable expectations for either a one or two people bachelor project. Both parties, Jarle Hjelen A.S. and I, discussed how it would be done and some potential outcomes. From there on, the project had started. I looked forward to putting some of the things I have learned to the test.

# 2 Software

Here is a short list of the primary software and tools used throughout the project.

## 2.1 Figma

Figma is a UX design program used for sketching, wireframing, designing, and prototyping. Figma is especially popular because of its ability to have multiple people work on the same design in conjunction with one another.

Within Figma, I have a couple of plugins. The only one of note for this project is Iconify. Iconify is a library of premade vector icons. One of which is the burger menu in the prototype design.

## 2.2 Visual Studio Code

Visual Studio Code is my code editor of choice for this project. Primarily because of familiarity, but also because it is a versatile IDE. There are common add-ons used to make development easier. One of note for this project is the Thunder client.

The Thunder client is a Rest testing API. It can be used to verify connections and pathways. I used this tool to test and work out the connection structure between the website and the e-mail server.

## 2.3 GIMP

GIMP, also known as GNU Image Manipulator Program. This is my image editor of choice and was used to touch up and alter some of the photos provided by Jarle Hjelen A.S.

## 2.4 Google Chrome

Chrome was used as a testing environment for the website. And the build-in dev tools were used to great effect.

# 3 Plan and execution

During the first bachelor meeting with my advisor and business contact, we laid down a basic plan for progress. In this section, I will expand upon this initial plan, how it evolved, how it was executed, and how the project plan ended up.

## 3.1 Initial plan

The initial plan was fundamentally divided into three different phases: the design phase, the development phase, and the writing phase. The idea behind this structure was to give each stage a distinct period in focus and then a transitional period into the next phase. This way, I would have enough time to work with each section properly and a smaller period afterwards to adjust and course-correct.

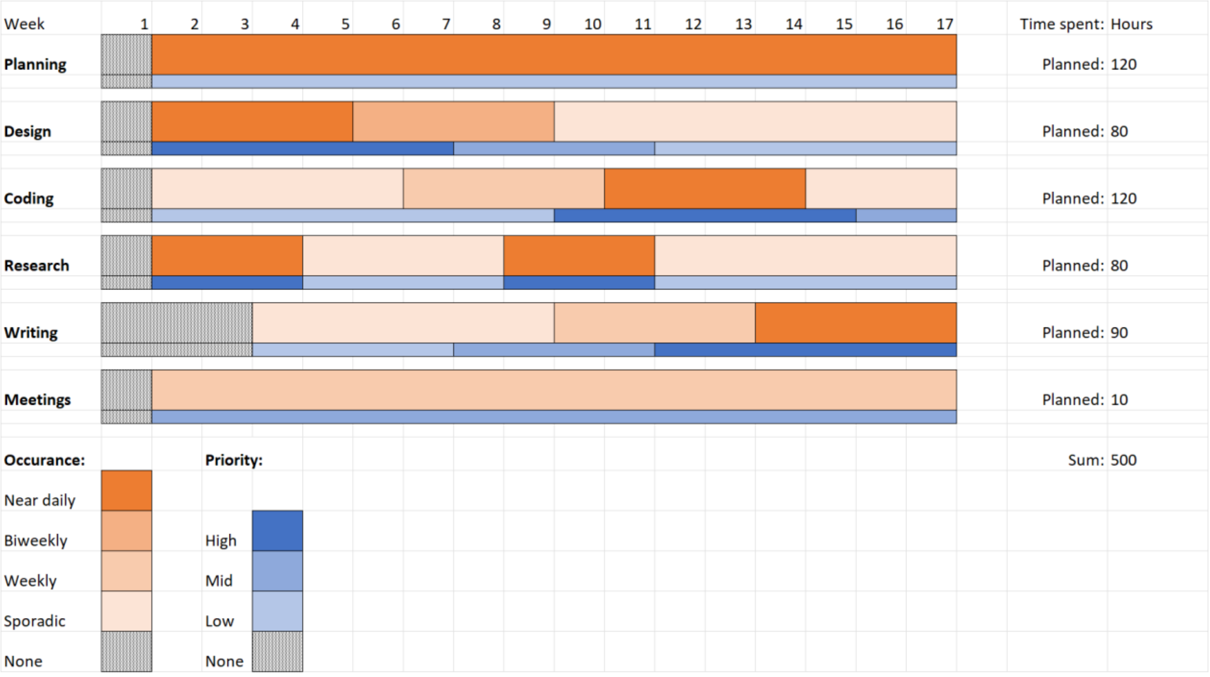


**Figure 3.1: Initial plan**

To expand upon this chart, the intended path would be this:
Start with the design phase, which itself starts with research. And when the initial research was done, focus on the design until about week 7 or 8. At this time, I would begin to transition to the development phase. The development phase, like the design phase, starts with research. Once I had a solid structure and plan, I would start coding. Then I would keep coding until the website is complete, ideally at week 13 or 14. That would leave a reasonable amount of time afterwards to focus entirely on writing.

## 3.2 Execution

In short, there were significant deviations from this plan. In the beginning, I managed to follow the schedule for the most part. But the further into the project I got, the more I had to alter the plan. In the end, I have had to change some of the goalposts due to either lack of or mismanagement of time.

## 3.3 Actualized plan

The actual plan ended up quite different. To summarise the changes, the design phase lasted a bit too long. This pushed the development phase further back and made it overlap with the writing stage. This also meant that I started writing later than I would have liked to. Therefore, neither the development nor writing phase became my sole focus for an extended period.
Beyond structural changes, I also had to refocus my time allotment. Firstly, I underestimated how much of my time would be spent researching. It took way longer to both find and understand the information. Conversely, I spent less time coding than expected. However, that could likely be due to the time spent researching and understanding all aspects of the code beforehand.
As mentioned earlier, the writing process started later than planned. The report framework and chapter structure were set up earlier. No significant writing work was done until the last three weeks.

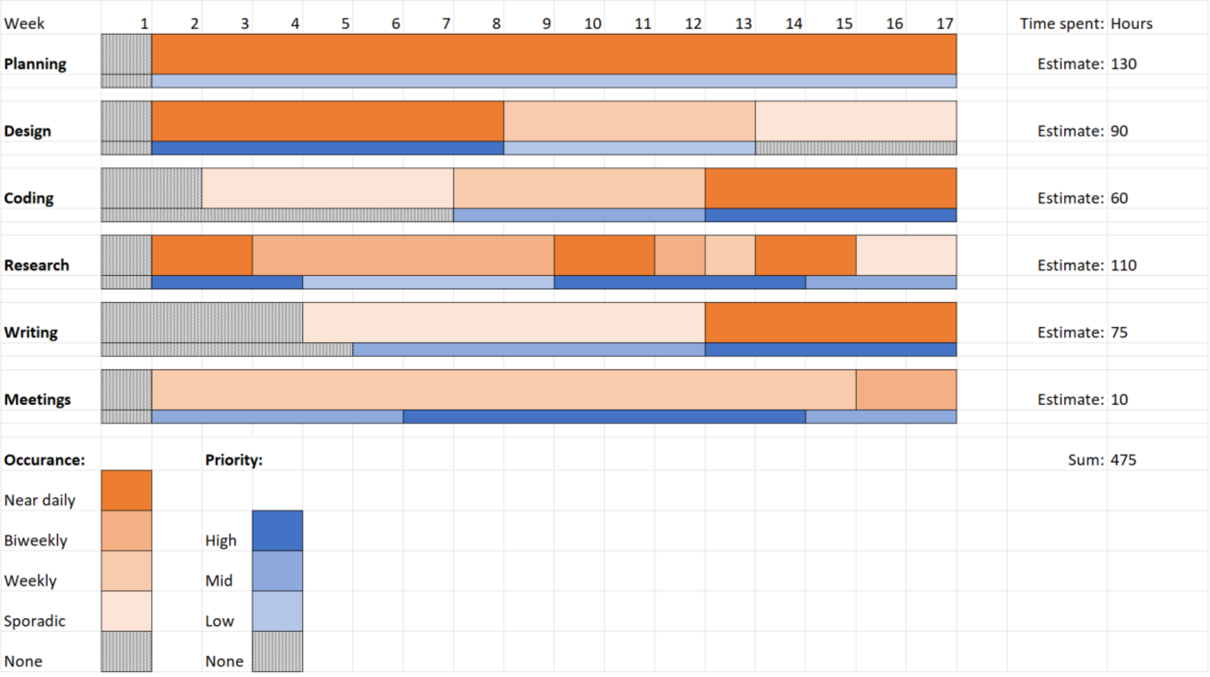


**Figure 3.2: Actualized plan**

# 4 Research

The research process was probably the one that surprised me the most. Here I'll show what my expectations were initially and how they were continually altered.

## 4.1 Initial expectation and changes

To begin with, I thought research would be straightforward. I need to figure out what I want to know more about and then find that information. However, almost every time I found something to research. It led me to a deeper field. A place where I had to make sure I even chose the right thing to read about in the first place. If I wanted to understand a dependency I would use, I found four others that could do the same job.

## 4.2 Time spent searching

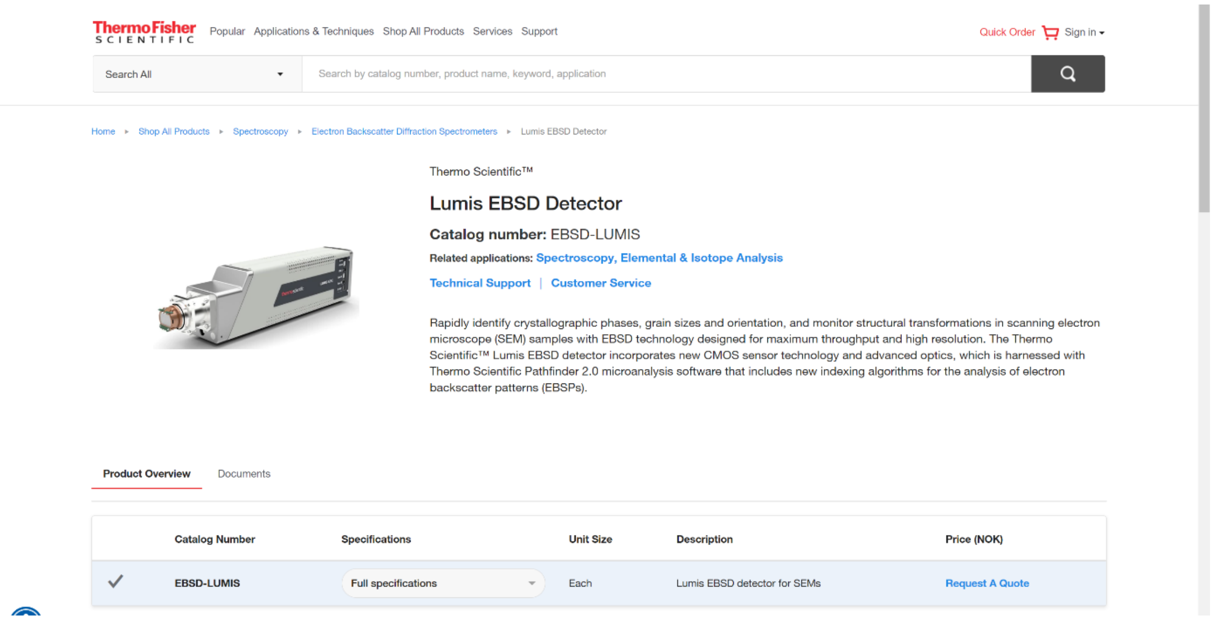
One of the major differences from my original plan, was time spent researching. But only a part of that difference was time spent reading. A lot of time was spent just searching for material. In a project like this one, where code is a significant part. Finding documentation, especially when two different systems interact, is very difficult. To the point where I could only find a singular source of documentation relevant to me on NodeMailer.

# 5 Visual design

In this chapter, I will go through the entire design process. From the initial ideation and planning, through the iterative design work, and finally, the realised design.

## 5.1 Initial planning and ideation

During the first few weeks of meetings, our topic of discussion was the design. To begin with, we looked at some competitor sites to see how they presented their website. As one might expect, it was a sea of whites, greys, and dull colours. All sites were squared and segmented. There was little consideration towards leading the eye line or following a 60-30-10 colour rule.



Home ▸ Shop All Products ▸ Spectroscopy ▸ Electron Backscatter Diffraction Spectrometers ▸ Lumis EBSD Detector

Thermo Scientific™

Lumis EBSD Detector

**Catalog number:** EBSD-LUMIS

**Related applications:** Spectroscopy, Elemental & Isotope Analysis

Technical Support | Customer Service

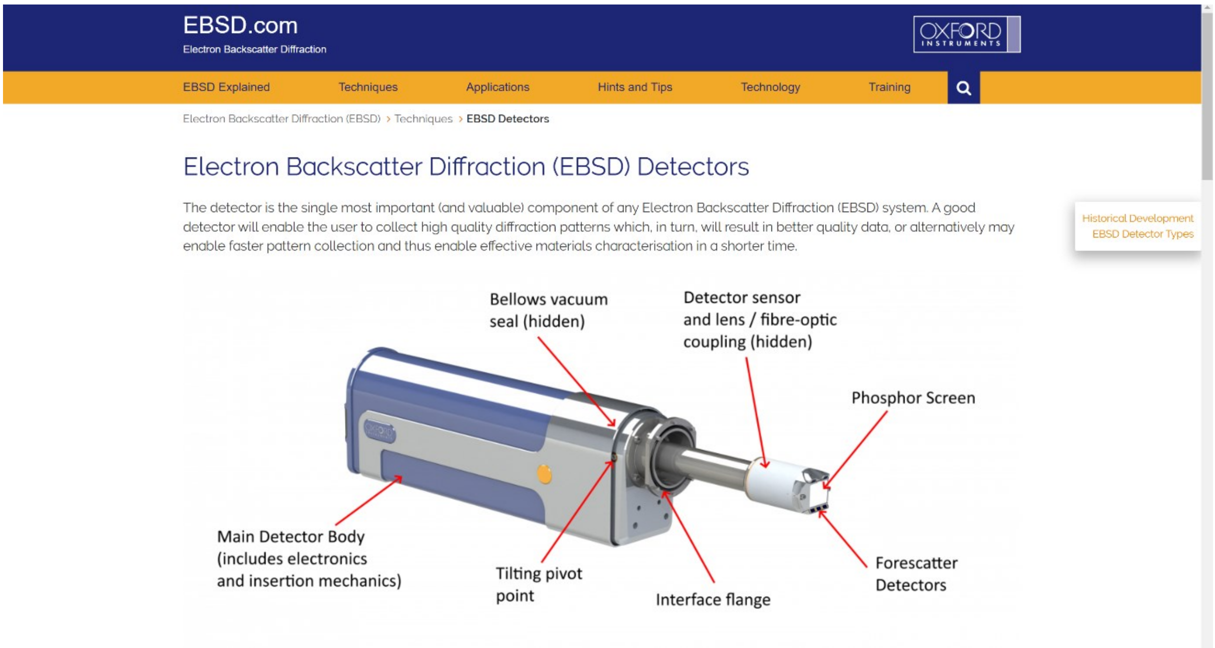
Rapidly identify crystallographic phases, grain sizes and orientation, and monitor structural transformations in scanning electron microscope (SEM) samples with EBSD technology designed for maximum throughput and high resolution. The Thermo Scientific™ Lumis EBSD detector incorporates new CMOS sensor technology and advanced optics, which is harnessed with Thermo Scientific Pathfinder 2.0 microanalysis software that includes new indexing algorithms for the analysis of electron backscatter patterns (EBSPs).

**Product Overview** Documents

| | Catalog Number | Specifications | Unit Size | Description | Price (NOK) |
|---|---|---|---|---|---|
| ✓ | EBSD-LUMIS | Full specifications | Each | Lumis EBSD detector for SEMs | Request A Quote |

**Figure 5.1: Thermo Fisher EBSD site**

Here we see Thermo Fisher’s webpage about one of their EBSD detectors. The site is plain white with a bit of colour. A rather basic design, not that a basic design is terrible, but this does not capture any attention.



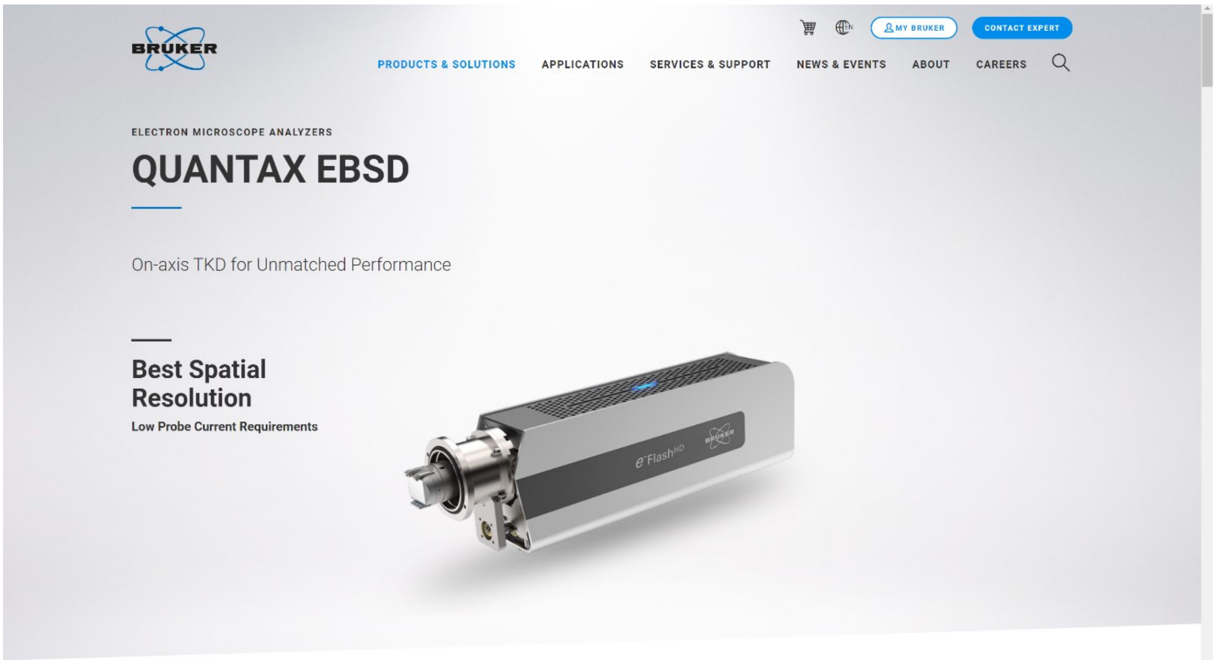
Electron Backscatter Diffraction (EBSD) Detectors

The detector is the single most important (and valuable) component of any Electron Backscatter Diffraction (EBSD) system. A good detector will enable the user to collect high quality diffraction patterns which, in turn, will result in better quality data, or alternatively may enable faster pattern collection and thus enable effective materials characterisation in a shorter time.

Bellows vacuum seal (hidden)
Detector sensor and lens / fibre-optic coupling (hidden)
Phosphor Screen
Main Detector Body (includes electronics and insertion mechanics)
Tilting pivot point
Interface flange
Forescatter Detectors

**Figure 5.2: Oxford Instruments EBSD site**

Oxford Instruments’ design is likewise basic but better, in my opinion. The big centre picture captures attention better while fluently explaining the detector's traits. Colour use is minimal, but the colours themselves feel connected to the brand. Compared to Thermo Fisher, where the colour felt like the default website light blue.



QUANTAX EBSD

On-axis TKD for Unmatched Performance

Best Spatial Resolution

Low Probe Current Requirements

**Figure 5.3: Bruker EBSD detector site**

Once again, an improvement over the other two sites. Like Oxford Instruments, we have a large central image to draw your attention. The colour use is still not as bold or present as I would like, but it is better integrated where it is. While the site is greyscale, it is no longer plain white. And there is a legitimate reason why a site should not be plain white backgrounds and pure black text. For both comfort and readability for the user, you should reduce the contrast between text and background. The only two times you want high contrast on text is when the background obscures or makes the text difficult to read. Or when it is used as a warning. Here we see a light grey in the background and deep grey text.

With all our discoveries about competitor websites, we now knew what kind of direction we wanted to move in. Nordif wanted to differentiate itself via its website. So, we decided to utilise these opposites of the competitors. Rounded instead of squared, darker compared to lighter sites, and using more colours and in more exciting ways.

Another place where the Nordif website could stand out was in structural differences. Their other competitors have a more expansive selection of different products and services. Nordif is exclusively focused on EBSD. This meant that we could give these EBSD detectors and services more of the limelight.

## 5.2 Thoughts and intentions

With the direction of the design chosen. I wanted to explain why I recommended some of these design elements. There was intention behind many of these choices beyond simply what looks good.

- Homely
    - One of the unique selling points of Nordif is its small scale and longevity. It is comparable to a long-running family business. This is proof that Nordif can provide and compete against much larger firms while having more personal support and customer relationship.
    - Making the site look distinctly "home-made" but well put together emphasises this small family business feel whilst also showing capability.
- Professional
    - With what was said earlier, this might seem contradictory. However, if the website does not present information in a professional manner. It will not inspire enough confidence in potential buyers to make them actual buyers.
- Rounded
    - The rounded design helps in the homely department, but it is also a current design trend to make more elements rounded. Especially on mobile units. All this, together with the fact it makes Nordif stand out as more unique and individually, made it a good choice in my mind.

A fundamental part of visual- and UX design is to convey information purely from visual and interactive experiences. With that in mind, I wanted to share the unique strengths of Jarle Hjelen A.S and Nordif through the design of the website.

## 5.3 Designing in Figma

Figma is a design tool with incredibly many uses and exceptional versatility. From being very customisable for each user to the ability for multiple users to cooperate on the same design document simultaneously. Figma or another tool like it is just about a must in graphic design and visual development today.

### 5.3.1 Wireframing

One of the first steps in designing a website. During this step, we worked out the fundamental structure and layout of the website. This is the first time you can get a feel of how the site will operate.

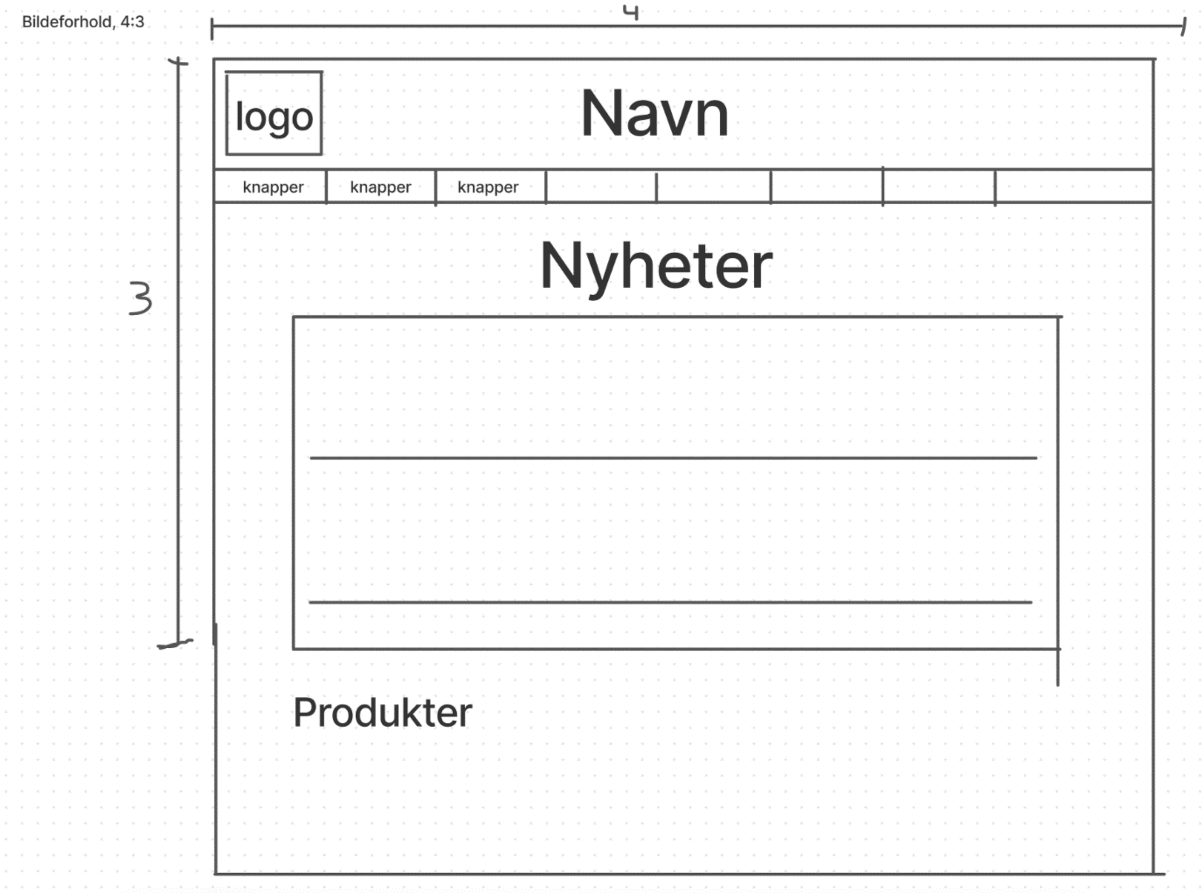
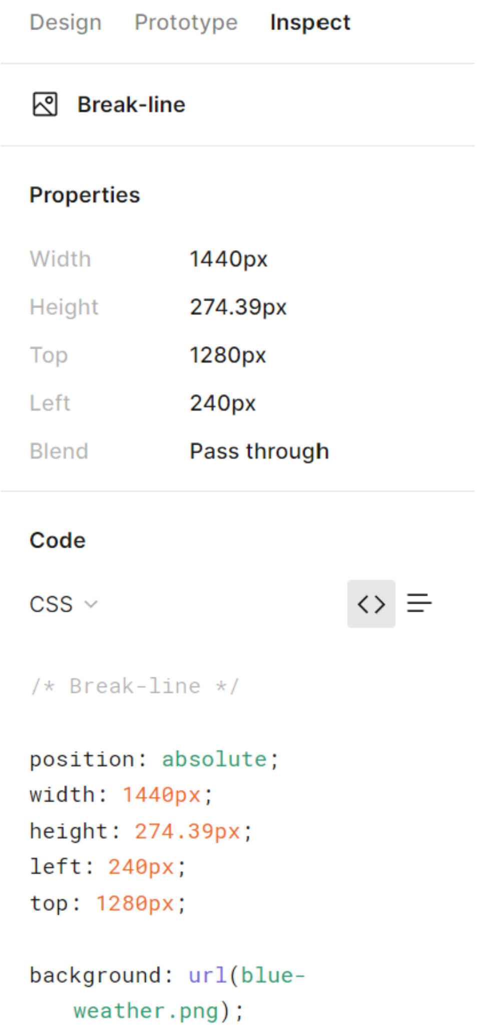


**Figure 5.4: Wireframe example**

During this step, you should think about the end-user. Is this the way they will use it? Is it designed with their use pattern in mind? To begin with, who are your users?
When Jarle Hjelen A.S and I looked at this question, it stood out. During the early meetings and until now, I gave the mobile site and desktop equal priority. Because that was part of the initial spec. But upon asking, we both realised that the vast majority of users would be researchers and technicians doing professional work. It was unlikely they browsed for business solutions on the phone. This train of thought led to more priority shifting over to the desktop site and only maintaining compatibility on mobile.

### 5.3.2 Design

Designing in Figma is a lot like most visual design software. But Figma does cater to web developers specifically. One way it does this is, E.g., Showing the CSS tags and properties that should produce a similar result.



| | |
|---|---|
| Width | 1440px |
| Height | 274.39px |
| Top | 1280px |
| Left | 240px |
| Blend | Pass through |

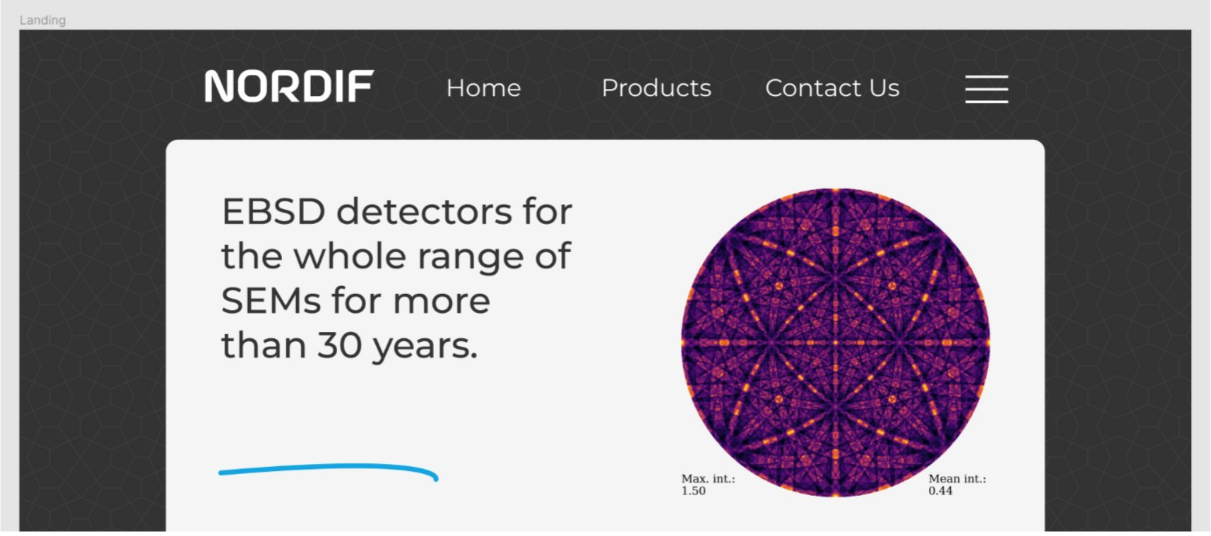
```
/* Break-line */

position: absolute;
width: 1440px;
height: 274.39px;
left: 240px;
top: 1280px;

background: url(blue-
   weather.png);
```

**Figure 5.5: Figma CSS output**

Another way Figma works well for web developers is in its ability to draw and export SVG elements. SVG elements are ideal for websites, especially for logos, icons, and other images that need to be scalable no matter the resolution.



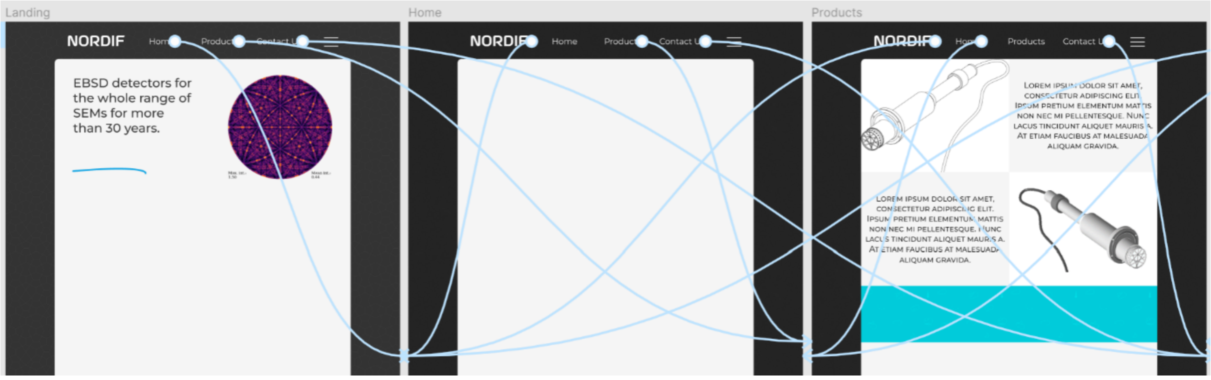
EBSD detectors for the whole range of SEMs for more than 30 years.

**Figure 5.6: Figma prototype landing page / Hero section**

Here is the design I made in Figma for the Nordif Site. Some of the elements here were still subject to change as the design would have to be transformed into a website. But I did feel this design encompassed the ideas proposed earlier, with maybe the exception of colour use. I did want to achieve a 60-30-10 ratio of colour. One thing in particular I liked about this was the logo design. It uses a free licence font called "a Astro Space". This font was inspired by NASA's worm logo from the mid-1970s. A design which fits in just right with the long-running family business feel whilst still being techy.

## 5.3.3 Prototyping

The final part of Figma. Prototyping. Prototyping in Figma uses multiple pages in conjunction to act like an actual website. You can establish connections, links, and buttons, which all have full function.



EBSD detectors for the whole range of SEMs for more than 30 years.

LOREM IPSUM DOLOR SIT AMET, CONSECTETUR ADIPISCING ELIT. IPSUM PRETIUM ELEMENTUM MATTIS NON NEC MI PELLENTESQUE. NUNC LACUS TINCIDUNT ALIQUET MAURIS A. AT ETIAM FAUCIBUS AT MALESUADA ALIQUAM GRAVIDA.

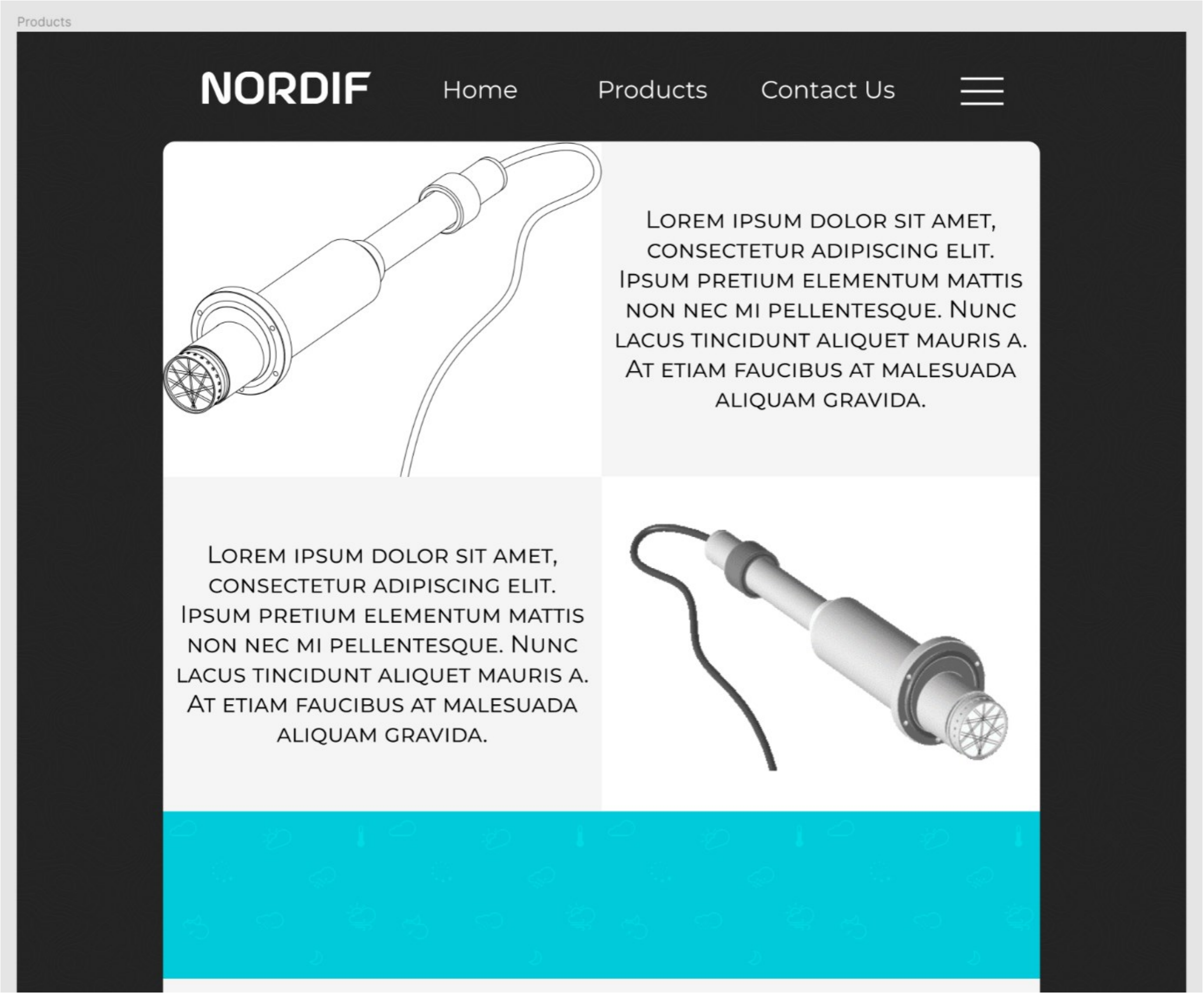
LOREM IPSUM DOLOR SIT AMET, CONSECTETUR ADIPISCING ELIT. IPSUM PRETIUM ELEMENTUM MATTIS NON NEC MI PELLENTESQUE. NUNC LACUS TINCIDUNT ALIQUET MAURIS A. AT ETIAM FAUCIBUS AT MALESUADA ALIQUAM GRAVIDA.

**Figure 5.7: Figma prototype connections**

This might look confusing and complex, but it is simply a visualisation of how the connection between pages and their navbar buttons is set up. By running this in Figma, you get to test the same experience you will have on the website. Beyond this point, you should be content with all design and user experience aspects.

## 5.4 Converting a Figma design into a website

Converting a Figma design into a website is at times straightforward, and at others, quite complex. It all depends on how much content is on the site and how specific you need it.

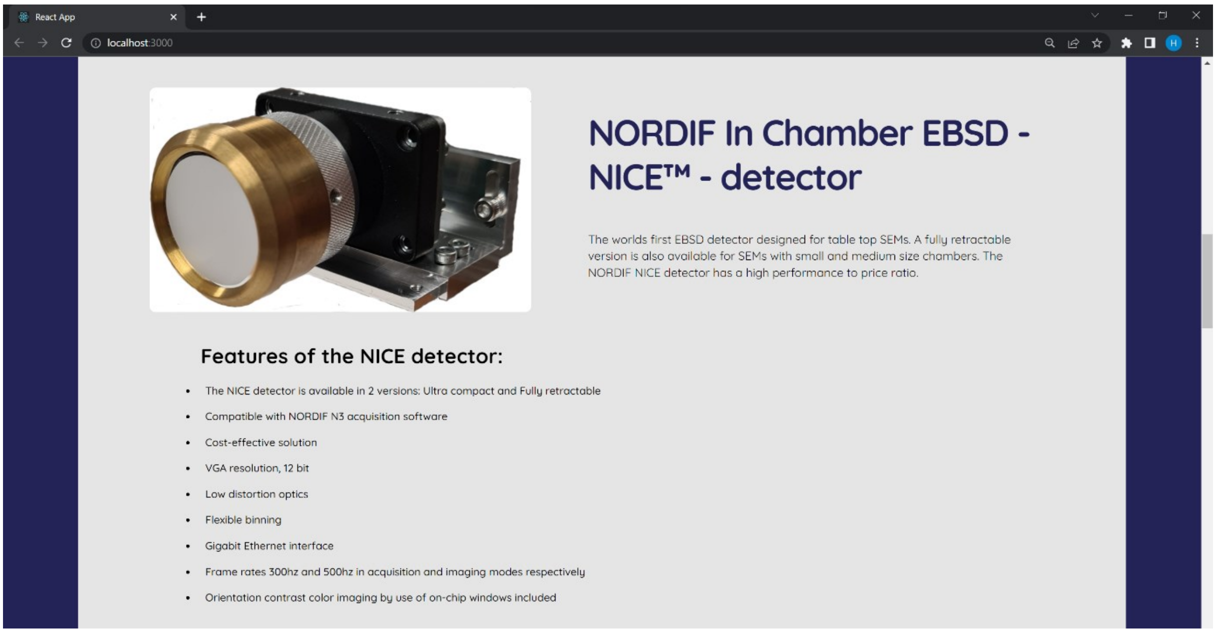


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ipsum pretium elementum mattis non nec mi pellentesque. Nunc lacus tincidunt aliquet mauris a. At etiam faucibus at malesuada aliquam gravida.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ipsum pretium elementum mattis non nec mi pellentesque. Nunc lacus tincidunt aliquet mauris a. At etiam faucibus at malesuada aliquam gravida.

**Figure 5.8: Figma prototype product page**

This was the general product page template made in Figma. It was based on the idea that an admin could add and remove pages, so the basic layout had to be simple. Hence why, it is four equal squares and a line spacer underneath. Those four squares were supposed to hold enough information for a product page, with some options on how to structure text and images. To me, it looks striking and has potential. The spacer line also adds a much-needed splash of colour.



NORDIF In Chamber EBSD - NICE™ - detector

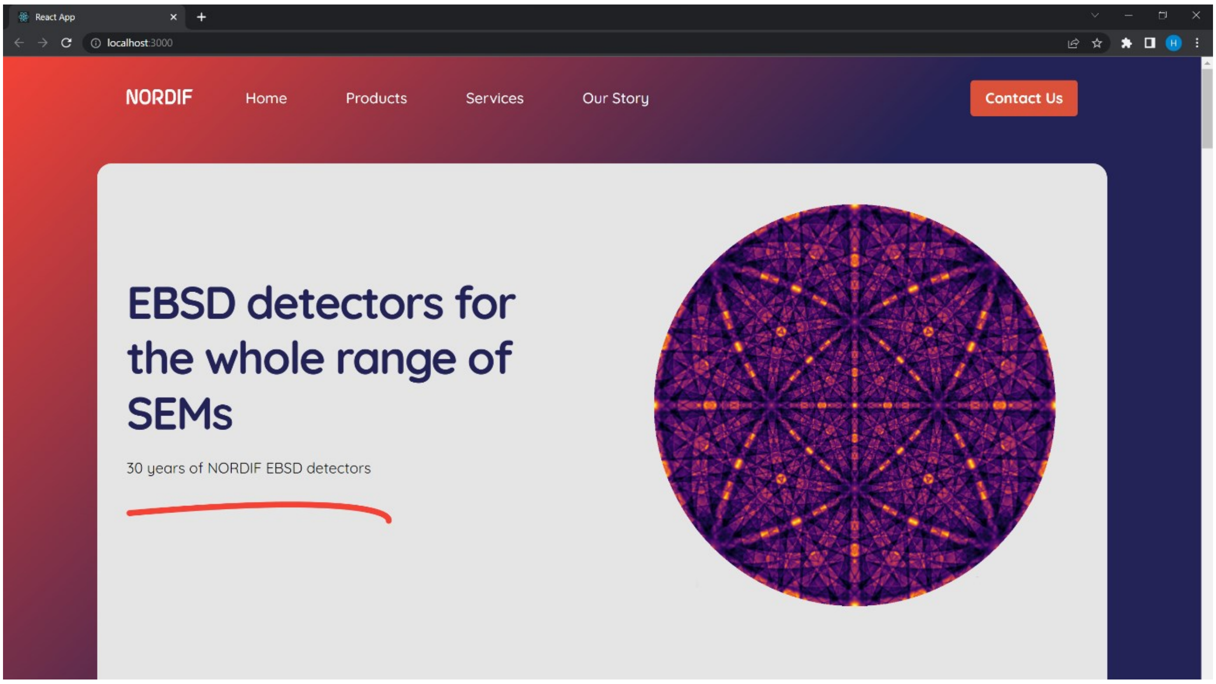
The worlds first EBSD detector designed for table top SEMs. A fully retractable version is also available for SEMs with small and medium size chambers. The NORDIF NICE detector has a high performance to price ratio.

Features of the NICE detector:

- The NICE detector is available in 2 versions: Ultra compact and Fully retractable
- Compatible with NORDIF N3 acquisition software
- Cost-effective solution
- VGA resolution, 12 bit
- Low distortion optics
- Flexible binning
- Gigabit Ethernet interface
- Frame rates 300hz and 500hz in acquisition and imaging modes respectively
- Orientation contrast color imaging by use of on-chip windows included

**Figure 5.9: Nordif NICE detector product page**

The actual product site ended up quite different. One part of this was the imagery provided, but that can be fixed later. The same rings true of the information—a balance between attracting customers with impressive specs but not showing competitors your hand. I will admit that the product page ideas within Figma are better than those on the website. Some of that is because of the statements above, and the rest was due to me not having time to fine-tune the CSS structure. Ultimately, the way it looks now, the structure lacks sharp, defined lines and divisions.

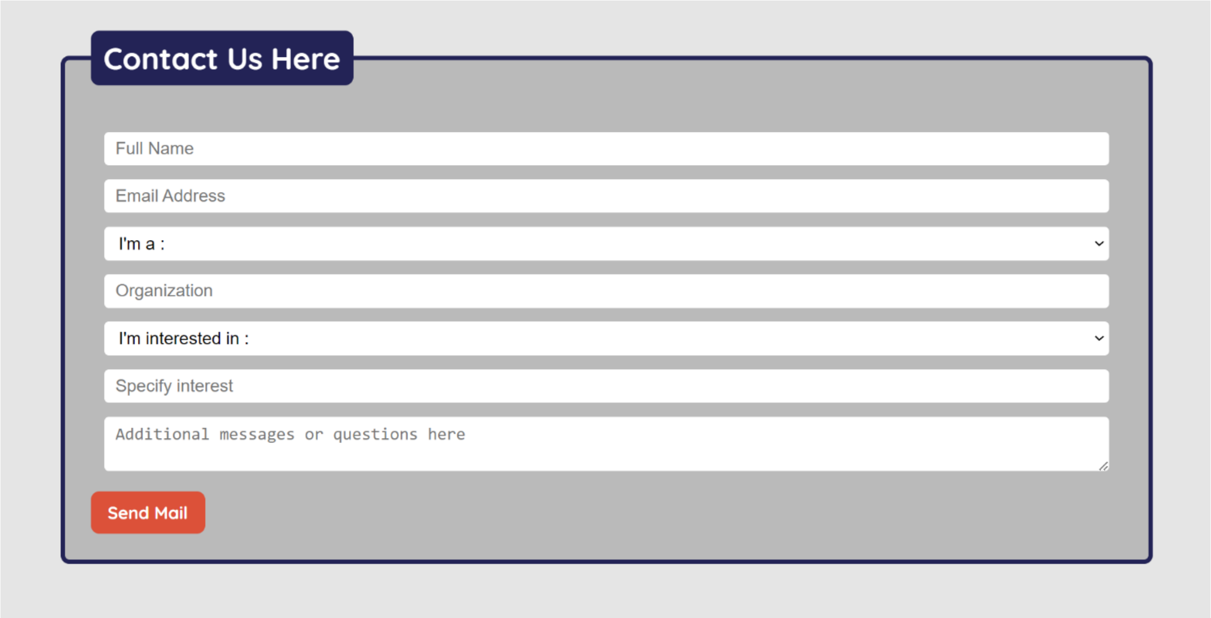


EBSD detectors for the whole range of SEMs

30 years of NORDIF EBSD detectors

**Figure 5.10: Nordif landing page / Hero section**

However, the colours are now themed and unified. Here we can see a warm orange to a deep blue. It was taken from the scanned image in the hero section. This puts us closer to the 60-30-10 rule and makes vital elements like the CTA stand out.



Contact Us Here

**Figure 5.11: Nordif contact form**

Much of the same can be said for the contact form. It’s fulfilling the ethos of the website in general.

## 5.5 Potential changes?

Several changes can be made visually to improve the website. In general, I think some of the scaling might be off. The fonts used should be different in titles than regular text. However, there are more specific changes I had in mind.

### 5.5.1 Information density

To me, the site feels a bit too airy. Since the site's purpose is to provide information about products and services, I think the site should be a bit denser. There is also the possibility this "too airy" feeling is because of the structuring and not the information density. Both options should be looked at as a possible improvement.

### 5.5.2 Images

The current images are not ideal. They are not even good. Every image, aside from Jarle's portrait, must be replaced. There should be high-quality 3D renders with proper texture mapping of the products and real-world pictures for this page to shine. The photos currently on the site have no thought put into the lighting, background, framing, or composition. It was beyond this project's scope, but I almost wanted to take some pictures myself. If only to serve as a guide for the actual pro photographer who should take the product pictures.

This also builds into another change I'd recommend. Make the product pages look more like the prototype. I genuinely believe that design is good. And compared to the current ones, it would significantly elevate the site's aesthetic.

## 5.6 What have I learned?

This experience has made me think more deeply about the intentions of visual design. If you use the visual aspect in tandem with whatever message you want to convey, they both amplify in meaning. Before, UX and graphic design were as simple as "does it look good" and" does it feel good". Now there is much more cohesion between all these facets.

Beyond the intangible concept above, I also understand more design principles. From more colour theory to the use of visual proximity. All my fundamental knowledge was solidified, and I expanded it into more aspects.
I have also gained a greater appreciation for how complex a topic design is. When researching, I never knew where to begin or end. What would be relevant, and what would not. Eventually, I gravitated toward whatever interested me and tried to apply that whenever possible.

# 6 Development

In this chapter, I will expand upon the development process. Explain the code's structure, How the actual development went, what can and should be done going forward, and finally, what I have learned from this experience.

## 6.1 Code ideology

Before I started this project, and further deepened during my research, I had some ideas and ideologies around code and coding. One of those fundamental ideologies is to make solutions at the lowest level whenever possible. One example of which I will expand upon later in this report. Another is consistency, which is helpful during development, but critical to making maintenance and further development after deployment bearable. Luckily one of the few advantages to doing this project alone was the ability to implement some of these fundamental ideologies.

There were also ideologies specific to this project. Given that others would handle further development, I tried to keep the project simple. There were times things planned changed due to this principle, and others did not. An example of each would be how MUI was dropped in favour of a CSS only solution. And how the email and contact form were progressively made more complex to produce a better solution.

One principle I picked up during research was BEM. BEM or Block Element Modifier is a guide that can be used to structure and name code components better. You can use BEM as a naming scheme where each part of BEM is named accordingly and connected with underscores or hyphens. An example:

- Block
    - The name of the container that encapsulates all elements.
        - Nordif (name of the site)
- Element
    - The name specific to a function within the block
        - Navbar (a component of the site)
- Modifier
    - The name of specific traits or flags of an element
        - Links (function of a navbar)

Same example from the code:

```
.nordif__navbar-links {

    flex: 1;

    display: flex;

    justify-content: flex-start;

    align-items: center;

}
```

**Figure 6.1: CSS and BEM example**

## 6.2 Coding

### 6.2.1 Node

Node.js is the most fundamental piece of structure in this project. Node.js is a JavaScript runtime, and it is this runtime that acts as the backbone for all add-ons and dependencies I used. It serves as the server component of both the website and the e-mail server.

### 6.2.2 React

Before development started, I had to make a choice. Angular or React. There were a couple of reasons this choice could have gone either way. I ended up using React, but much of said reason was simply that I was familiar with React from before. While I’m sure, there are advantages to both. And either could have worked with the website. The main reason I considered Angular was as a learning experience. I ultimately decided against it because I felt more comfortable with React and thought I could use it to make a better final product.

React is, in essence, a JavaScript library based on components and a compartmentalised structure. There are many reasons to use React and other similar systems to improve UI. The primary one for me is the structure and capability it provides. Component-based code structure allows for many benefits. It makes development more straightforward. By creating versatile components that can be reused, you cut down on development time and code complexity. These same components make expansion easier. Since you now have a framework to work within, expansion is as easy as adding new information and data in a template.
Beyond benefits to a single developer, this system makes development easier for projects with multiple developers. By having an agreed-upon structure, other people can work in conjunction on the same project. This also includes future developers who did not originally work on the project.

### 6.2.3 CSS

Early in the project, before actual development, the plan was to use MUI to implement the design onto the website. However, as I mentioned earlier, I would likely not continue development after the degree. And I wanted to keep the website as simple as possible for whoever had to pick up the code after me. I decided on dropping MUI in favour of making everything in CSS from scratch.

In this project, I ended up using CSS a reasonable amount. And as mentioned earlier, during research, I learned about BEM. BEM was a large part of what kept the CSS part of this project manageable and nicely structured. I have also used several online tools to assist with CSS code. Primarily from angrytools.com (Angrytools.com, n.d.) and Animista (Animista, n.d.). Where I used the gradient tool from angrytools.com to create the background colour. And Animista for animations on elements like the burger menu.

### 6.2.4 Contact form

The email system turned out to be an ever-evolving task. Whenever I thought I had an acceptable solution, new ideas and realisations occurred. Initially, I planned on using the build-in email solution in HTML and PHP. However, it became clear that such a solution did not provide sufficient security for Jarle Hjelen A.S. I could not obfuscate the email address or password used with this set-up. Which meant they could be extracted and exploited. I was considering these problems and the fact that these are older protocols by this point. I decided to look for other solutions.

I then moved on to find a React dependency I could use instead. NodeMailer quickly stood out as my most convenient option. This would also allow me to externalise the email codebase. This alone helped security significantly. Beyond the externalisation, I also used dotenv, another dependency that stores data within an encrypted file. Together, both the externalised email server and the user data encrypted. I had sufficient security on the backend.

```
let transporter = nodemailer.createTransport({ // Connects to the API and verifies the user
    service: "gmail",
    auth: {
      type: "OAuth2",
      user: process.env.EMAIL,
      pass: process.env.WORD,
      clientId: process.env.OAUTH_CLIENTID,
      clientSecret: process.env.OAUTH_CLIENT_SECRET,
      refreshToken: process.env.OAUTH_REFRESH_TOKEN,
    },
  });
  transporter.verify((err, success) => { // Sends a message regarding server status
    err
      ? console.log(err)
      : console.log(`=== Server is ready to take mail: ${success} ===`);
  });
```

**Figure 6.2: E-mail server main function**

However, I discovered that I would need further dependencies to set up a supporting structure upon trying to use NodeMailer this way. That led me to Express and CORS. This is middleware, an in-between code structure used as a web framework. These are essentially dependencies used to create communication paths between the website proper and the now external email server.

```
const cors = require("cors");
const express = require("express");
const nodemailer = require("nodemailer");
const app = express();
require("dotenv").config();

// middleware, used to comunicate between the API and site

app.use(express.json());
app.use(cors());
```

**Figure 6.3: E-mail server dependencies**

While the pathways had been set up on the backend, Jarle Hjelen A.S planned to use a Gmail account to manage email. That meant using the Gmail API and OAuth2 to send emails. A simple enough system to set up via Google's developer playground.
This way of managing email also works in another favour. Nordif can now use this email; nordif.contact@gmail.com as a filter. This is because the form uses this email both as a sender and receiver. By making the account forward all mail from itself to another business email, you filter out all mail sent from other senders.

However, I realised it was still insufficient when all was said and done. While I had set up a good amount of security on the backend, much could be done for the frontend. There are some safety features which should be implemented. The plan was to use react-hook-form to add more features to the form. With react-hook-form, add checks on the inputs so users cannot spam an unlimited number of characters in each field. I would add a character limit on each input and a check to ensure each input has the correct data. Beyond safeguarding from each email being full of useless data and spam, there should be a way to stop users from spamming many emails. The plan was very early in the project to use a CAPTCHA or reCAPTCHA for this. Ultimately, I did not have time to add this, but this is still how I would go about stopping spam emails.

## 6.3 Further development

The website and e-mail server are both in a functional state where they can be deployed. However, there are still changes and further development I recommend before deployment.

### 6.3.1 React router

React router is another add-on for React that I planned to use. It is used to separate elements of a website into multiple pages. While the site is simply a scrolling site now, it should be divided over multiple pages. This is by no means necessary, which is why it was down prioritised. But it does make the page more structured and visually appealing.

### 6.3.2 The admin page

One of the main items in the project description was an admin page. This was going to be a specific page locked behind an admin user. On this page, the admin would have the ability to alter or delete existing pages and add new pages or update the newsfeed.

Unfortunately, due to alterations in the plan and lack of time, this feature was not completed. I did have the logic planned and some structure set up. But I did not manage to complete the system in time and had to drop it to make sure the other features were correctly implemented.

The admin page was to be locked behind the admin user. The plan was to do this with either Firebase or BeeCrypt, as it would allow for an easy way to manage user authentication. I wanted to use either of these over node.js authentication as I was no security expert. Any custom solution made by me; would likely have oversights I wouldn’t have foreseen. A complete pre-made solution would be faster implemented and safer.

The admin system itself was based on SQL. Using an SQL server to store templates for different pages and then sending the updated information and content via the admin page. Once this was done, the site would update with whatever changes were made in the SQL server. This was a similar system to an earlier project I worked on. However, I was primarily working on the front-end of that project and therefore was not intimately familiar with the backend structure. Not wanting to cannibalise code from a former project without entirely understanding it, I had to try and learn as much as possible to create my version of it. I could not do this in a reasonable amount of time, so I had to stop working on the system.

#### 6.3.2.1 Newsfeed

The newsfeed was also one of the initially planned pages. However, this section was also unfinished due to dependence on the admin system. The page was pointless, with no admin system to add new posts to the feed.

## 6.4 What have I learned?

To start with, a lot of what I have learned can be boiled down to expanded skill and knowledge about different code languages. I have learned new ways to use CSS. I have learned about the many ways you can use react, node.js, and their respective add-ons and dependencies.

I have also learned more about the intangible and less defined subject matter. I now know how I work and how to improve the way I work. I have learned that I become productive in almost every other area of my life when working on such a significant task. And how I must do everything else I can to clear my head before I can sit and dedicate my focus to a single task. How much focus and attention coding demands not to make mistakes.

The main reason degrees exist is to prove qualifications and provide a learning experience. Throughout this project, I have learned a lot. From the perspective of coding, I believe I could now make a website of twice the quality in half the time. Doing a complete project singlehandedly has reinforced my understanding of fundamental code principles and expanded my knowledge in multiple ways. So, while I feel that I have more knowledge about web development now, I also have a deeper understanding of how shallow my knowledge is. So, while I'm left feeling a bit inadequate, I'm also looking forward to learning more.

# 7 Final thoughts and conclusion

I wanted to touch on some ideas before I must leave.

## 7.1 Developer or designer

Throughout my entire degree, this has been a question on my mind. Do I focus on UX and the frontend because I like to develop it, or design it? For a long period of time now, I thought it was the latter. I focus on UX so I can design. But now, sitting and finishing the last few words, I think otherwise. I like the development; it's just hard to be good enough to realise the vision in my head. But from now on, I want to be able to affix my vision on the page. I want to create it myself.

## 7.2 My experience

This project has been a more significant learning experience than any other class or project. I now know that I can do so much more after, than I did before. And that in and of itself, will help me onward.

*A human being should be able to:*

*change a diaper,*
*plan an invasion,*
*butcher a hog,*
*conn a ship,*
*design a building,*
*write a sonnet,*
*balance accounts,*
*build a wall,*
*set a bone,*
*comfort the dying,*
*take orders,*
*give orders,*
*cooperate,*
*act alone,*
*solve equations,*
*analyse a new problem,*
*pitch manure,*
*program a computer,*
*cook a tasty meal,*
*fight efficiently,*
*die gallantly.*
*Specialization is for insects.*

*— Robert A. Heinlein, Time Enough for Love (1973)*

Finally, I can check one off the list

# References

Andris Reinman (2010). *Nodemailer :: Nodemailer*. [online] Nodemailer.com. Available at: https://nodemailer.com/about/ [Accessed 18 May 2022].

Angrytools.com. (n.d.). *Online Gradient Generator*. [online] Available at: https://angrytools.com/gradient/ [Accessed 18 May 2022].

Animista (n.d.). *Animista - CSS Animations on Demand*. [online] Animista - CSS Animations on Demand. Available at: https://animista.net/ [Accessed 18 May 2022].

Archibald, J. (n.d.). *SVGOMG - SVGO's Missing GUI*. [online] jakearchibald.github.io. Available at: https://jakearchibald.github.io/svgomg/ [Accessed 18 May 2022].

Bruker (n.d.). QUANTAX EBSD. [online] www.bruker.com. Available at: https://www.bruker.com/en/products-and-solutions/elemental-analyzers/eds-wds-ebsd-SEM-Micro-XRF/quantax-ebsd.html [Accessed 18 May 2022].

CSS-Tricks. (2015). *BEM 101*. [online] Available at: https://css-tricks.com/bem-101/ [Accessed 18 May 2022].

en.bem.info. (n.d.). *Methodology / BEM*. [online] Available at: https://en.bem.info/methodology/ [Accessed 18 May 2022].

Facebook (n.d.). React – A JavaScript library for building user interfaces. [online] Reactjs.org. Available at: https://reactjs.org/ [Accessed 18 May 2022].

FontSpace (2006). Free Fonts | 64000+ Font Downloads. [online] FontSpace. Available at: https://www.fontspace.com/ [Accessed 18 May 2022].

freeCodeCamp.org (2022). *React Course - Beginner's Tutorial for React JavaScript Library [2022]*. [online] www.youtube.com. Available at: https://www.youtube.com/watch?v=bMknfKXIFA8 [Accessed 18 May 2022].

GitHub. (2022). *Cors*. [online] Available at: https://github.com/expressjs/cors [Accessed 18 May 2022].

Interaction Design Foundation (2019). *What is User Experience (UX) Design?* [online] The Interaction Design Foundation. Available at: https://www.interaction-design.org/literature/topics/ux-design [Accessed 18 May 2022].

JavaScript Mastery (2021a). *Build and Deploy a Fully Responsive Modern UI/UX Website in React JS*. *YouTube*. Available at: https://www.youtube.com/watch?v=LMagNcngvcU [Accessed 18 May 2022].

JavaScript Mastery (2021b). *JavaScript Tutorial for Beginners - Learn JavaScript in 2021 | Course*. *YouTube*. Available at: https://www.youtube.com/watch?v=g7T23Xzys-A [Accessed 18 May 2022].

JavaScript Mastery (2021c). *Learn Material UI in One Hour - React Material UI Project Tutorial [2021]*. *YouTube*. Available at: https://www.youtube.com/watch?v=Xoz31I1FuiY [Accessed 18 May 2022].

JavaScript Mastery (2022). *React JS Full Course 2022 | Build an App and Master React in 1 Hour*. [online] www.youtube.com. Available at: https://www.youtube.com/watch?v=b9eMGE7QtTk [Accessed 18 May 2022].

Khan Academy (2019). *Intro to SQL: Querying and managing data*. [online] Khan Academy. Available at: https://www.khanacademy.org/computing/computer-programming/sql [Accessed 18 May 2022].

Long, J. (2021). *How to implement email functionality with Node.js, React.js, Nodemailer, and OAuth2*. [online] DEV Community. Available at: https://dev.to/jlong4223/how-to-implement-email-functionality-with-node-js-react-js-nodemailer-and-oauth2-2h7m.

Material UI SAS (n.d.). *MUI: The React component library you always wanted*. [online] mui.com. Available at: https://mui.com/ [Accessed 18 May 2022].

Motdotla (2019). *motdotla/dotenv*. [online] GitHub. Available at: https://github.com/motdotla/dotenv [Accessed 18 May 2022].

Node.js Foundation (2014). About | Node.js. [online] Node.js. Available at: https://nodejs.org/en/about/ [Accessed 18 May 2022].

OpenJS Foundation (2017). *Express - Node.js web application framework*. [online] Expressjs.com. Available at: https://expressjs.com/ [Accessed 18 May 2022].

Oxford Instruments (n.d.). EBSD Detectors. [online] Oxford Instruments. Available at: https://www.ebsd.com/ebsd-techniques/ebsd-detectors [Accessed 18 May 2022].

w3schools (2019). *SQL Introduction*. [online] W3schools.com. Available at: https://www.w3schools.com/sql/sql_intro.asp [Accessed 18 May 2022].

Scott, D. (2021). *Free Figma UX Design UI Essentials Course*. [online] www.youtube.com. Available at: https://www.youtube.com/watch?v=kbZejnPXyLM [Accessed 18 May 2022].

Thermo Fisher (n.d.). Lumis EBSD Detector. [online] www.thermofisher.com. Available at: https://www.thermofisher.com/order/catalog/product/EBSD-LUMIS [Accessed 18 May 2022].

Yablonski, J. (2018). *Home* | *Laws of UX*. [online] Lawsofux.com. Available at: https://lawsofux.com/ [Accessed 18 May 2022].

NTNU

Kunnskap for en bedre verden