

Sivert Grønli Amundsen, Ole Ragnar Randen &
Mons Stenbråten

The Yggdrasil Project

A digital solution to make pilgrimage and local stories around Gjøvik more engaging.

Bachelor's thesis in Bachelor Web Development

Supervisor: Eleftherios Papachristos

May 2023

Sivert Grønli Amundsen, Ole Ragnar Randen & Mons Stenbråten

The Yggdrasil Project

A digital solution to make pilgrimage and local stories around Gjøvik more engaging.

Bachelor's thesis in Bachelor Web Development
Supervisor: Eleftherios Papachristos
May 2023

Norwegian University of Science and Technology
Faculty of Architecture and Design
Department of Design



FOREWORD

We write this thesis as a part of our bachelor's degree in web development at NTNU in Gjøvik, Norway. The subject for the thesis is pilgrimage travel, where we explore how to increase the interest in it. We delve into the technical aspects of developing an application as well as the design aspects of creating an interesting solution.

We would like to express our gratitude to our tutor Eleftherios Papachristos for his valuable guidance and recommendations throughout the process. Additionally, we extend our thanks to Ole Edward Wattne for providing us with essential resources for our application. Finally, we want to thank everyone at BWU who assisted us in conducting user testing.

A handwritten signature in black ink, reading "Sum Groth Amundsen". The signature is written in a cursive style with a large initial 'S'.A handwritten signature in black ink, reading "Ole Ragnar Rander". The signature is written in a cursive style.A handwritten signature in black ink, reading "Mons Stalviken". The signature is written in a cursive style.

ABSTRACT (NO)

Tittel: The Yggdrasil Project – A digital solution to make pilgrimage and local stories around Gjøvik more engaging.

Dato: 15.05.2023

Deltakere: Sivert Grønli Amundsen, Ole Ragnar Randen, Mons Stenbråten

Veileder: Eleftherios Papachristos

Oppdragsgiver: Ole Edward Wattne for *rurAllure*

Fagfelt: Webutvikling

Stikkord: Pilgrimage, Gjøvik, map, POIs, story, audio narratives, local areas, rural, interest, SUS

Antall sider: 72 (Kapittel 1-6) + 38

Antall ord: 20394

Antall vedlegg: 15

Denne oppgaven beskriver hvordan en applikasjon med flere egenskaper, som for eksempel et interaktivt kart som viser forskjellige lokale historier med lyd narrativer, og en informasjonsside om litt grunnleggende info angående pilegrimsreise, vekker interesse for både pilegrimsreiser og lokalhistorie. For å måle interesse, engasjement og brukskvalitet ble det administrert en System Usability Scale (SUS) spørreundersøkelse med 10 deltakere. Analysen viste en økning i interesse, selv om resultatene kan ha blitt påvirket av bias.

ABSTRACT (EN)

Title: The Yggdrasil Project - A digital solution to make pilgrimage and local stories around Gjøvik more engaging.

Date: 15.05.2023

Participants: Sivert Grønli Amundsen, Ole Ragnar Randen, Mons Stenbråten

Supervisor: Eleftherios Papachristos

Employer: Ole Edward Wattne for *rurAllure*

Subject: Web development

Keywords: Pilgrimage, Gjøvik, map, POIs, story, audio narratives, local areas, rural, interest, SUS

Number of pages: 72 (Chapter 1-6) + 38

Number of words: 20394

Number of appendices: 15

This thesis details how an application with several features, such as an interactive map displaying various local stories accompanied by audio files, and an information page about pilgrimage in general, generates interest in both pilgrimage and local stories. To assess interest, engagement, and usability a System Usability Scale (SUS) questionnaire was administered to 10 participants. The analysis showed an increase in interest, although the results may have been influenced by bias.

CONTENT

Foreword	1
Abstract (NO)	2
Abstract (EN).....	3
1 Introduction	10
1.1 Solution.....	11
1.1.1 The different pages.....	11
1.2 Problem Statement.....	13
1.2.1 Context for our Problem Statement	13
1.2.1.1 Gjøvik	14
1.2.1.2 Pilgrim.....	14
1.3 Shortcomings	14
1.4 Scope	15
1.5 The report structure	16
2 Background, theory and early development	18
2.1 Early planning and prior work.....	18
2.2 From concept to finished design theory.....	18
2.2.1 Target audience	18
2.2.2 Personas	19
2.2.3 SWOT	19
2.2.4 Usability Competitive analysis	20
2.2.5 Mobile first design	20
2.2.6 User testing.....	21
2.2.6.1 The different user testing methods.....	21
2.2.6.2 Bias.....	22
2.2.7 Questionnaire	23
2.2.7.1 Standardized questionnaires	23
2.2.7.2 The System Usability Scale (SUS)	23
2.2.8 Prototyping	24
2.2.8.1 Low fidelity prototyping (Lo-fi)	24
2.2.8.2 High fidelity prototyping (Hi-fi).....	25
2.2.9 MERN	25
2.2.9.1 Frontend, React.js.....	25

2.2.9.2 Backend, Express and node.js.....	26
2.2.9.3 Database, MongoDB.....	26
2.2.10 CSS framework.....	26
2.2.10.1 Tailwind CSS.....	26
2.2.11 Google Maps JavaScript API.....	27
2.2.12 Dependencies – Node packages.....	27
2.2.12.1 React feather.....	27
2.2.12.2 Animate.css.....	27
2.2.12.3 Material UI.....	27
2.2.12.4 Axios.....	28
2.2.12.5 React Google maps API.....	28
2.2.12.6 React H5 media player.....	28
2.2.12.7 React Router.....	28
2.2.12.8 Mongoose.....	28
2.2.12.9 Body Parser.....	28
2.2.12.10 Focus-Trap-React.....	28
2.2.12.11 Multer.....	29
2.2.13 Accessibility.....	29
2.2.13.1 Disabilities.....	29
2.2.13.2 WCAG - Web Content Accessibility Guidelines.....	29
2.2.13.3 WAVE - Web Accessibility Evaluation Tools.....	29
2.2.13.4 Google Lighthouse.....	30
2.2.13.5 Tab index.....	30
2.2.13.6 Aria.....	30
2.2.14 Sustainability.....	30
2.2.14.1 code splitting.....	30
2.2.14.2 Lazy loading.....	31
2.2.14.3 File types and formats.....	31
2.2.14.4 Redundancy in code.....	32
3 Methods.....	33
3.1 Designing Yggdrasil.....	33
3.1.1 Target audience.....	33
3.1.2 Personas.....	33
3.1.3 Usability Competitive analysis.....	34
3.1.3.1 Finding competitors.....	34
3.1.3.2 SWOT.....	35

3.1.4 Low fidelity prototyping (Lo-fi)	35
3.1.4.1 Wireframe.....	35
3.1.4.2 Creating a low fidelity of the map idea.....	36
3.1.4.3 Lo-fi prototype user testing	37
3.1.5 High fidelity prototyping (Hi-fi).....	38
3.1.5.1 Creating the Hi-fi.....	38
3.1.5.2 Hi-fi user testing.....	40
3.1.5.3 Logo and brand name	40
3.2 Developing Yggdrasil.....	42
3.2.1 Frontend	42
3.2.1.1 Frontend structure.....	43
3.2.1.2 Developing mobile first.....	44
3.2.2 Backend	44
3.2.2.1 Backend structure.....	44
3.2.2.2 Database	45
3.3 The Yggdrasil solution	46
3.3.1 Changes from the high-fidelity	46
3.3.2 How we tested the Yggdrasil application	48
3.3.2.1 Choosing a standardized questionnaire	48
3.3.2.2 How we used SUS	49
3.3.2.3 Reducing bias.....	49
3.3.2.4 Making our own questionnaire to measure interest.....	49
3.4 Sustainable measures	50
3.4.1 Code splitting and lazy loading	50
3.4.2 File types and file size	51
3.4.3 Remove redundant code	52
3.4.4 Testing the sustainability of the website	52
3.5 Accessibility measures	52
3.5.1 Design choices	52
3.5.1.1 Colours.....	52
3.5.1.2 Using tab index	53
3.5.1.3 Transcript.....	53
3.5.1.4 Element sizing.....	53
3.5.1.5 Element Hover	54
3.5.1.6 Image Alt text	54
3.5.2 Testing the accessibility of the website	54

3.5.2.1 Using wave.....	54
3.5.2.1 Using Google Lighthouse for accessibility.....	54
3.6 Organizing the project	54
3.6.1 Communication and work mediums.....	55
3.6.1.1 Discord	55
3.6.1.2 Microsoft Teams	55
3.6.1.3 Collaboration using GitHub.....	55
3.6.1.4 Google Docs	56
3.6.1.5 Conducting a questionnaire with google forms.....	56
3.6.1.6 Figma	56
4 Results.....	57
4.1 Design results.....	57
4.1.1 Looking into our competitors	57
4.1.1.1 SWOT results	57
4.1.1.2 Usability Competitive analysis results	58
4.1.2 Low fidelity prototyping (Lo-fi)	58
4.1.2.1 Results from lo-fi user testing.....	59
4.1.3 High fidelity prototyping (Hi-fi).....	59
4.1.3.1 Results from hi-fi user testing.....	59
4.2 The Yggdrasil project	60
4.2.1 User testing the final product.....	60
4.2.1.1 SUS-questionnaire	60
4.2.1.2 Gauging interest	61
4.2.1.3 Observing how the users navigate the final website.....	62
4.3 Sustainability & Accessibility results.....	63
4.3.1 Applying code splitting and lazy load.....	63
4.3.2 Removing redundant code	67
4.3.3 WebP as image format	68
4.3.3.1 Reducing file size further by compressing WebP files.	68
4.3.4 Mp3 as audio format	69
4.3.4.1 Reducing mp3 files with compression	69
4.3.5 The final folder size.....	70
4.3.6 Lighthouse testing the accessibility and sustainability of the website	70
4.3.7 WAVE test	71
5 Discussion	72
5.1 Design and development	72

5.1.1 Strengths and weaknesses of our design and development	72
5.1.1.1 Communication with our tutor	72
5.1.1.2 Target audience	72
5.1.1.3 Competitor analysis	72
5.1.1.4 Theme	73
5.1.1.5 Scalability and quality moderation	73
5.1.1.6 Why MERN?	73
5.1.1.7 Transcription of the narratives	74
5.1.1.8 Components.....	74
5.1.1.9 Expansion of map feature.....	74
5.2 Sustainability and accessibility	74
5.2.1 Is there room for improvements?	74
5.3 Creating interest	75
5.3.1 General thoughts on user testing	75
5.3.1.1 Lo-fi testing	76
5.3.1.2 Hi-fi testing	76
5.3.1.3 Application testing.....	76
5.3.2 Was interest increased?	77
5.4 Further development.....	79
6 Conclusion.....	80
7 Bibliography	82
8 Figure list.....	85
9 Appendix	86
9.1 Our personas	86
9.2 The questions and answers from the lo-fi prototype testing.....	87
9.3 The q and a, tasks and additional note from the hi-fi prototype testing.....	88
9.4 Application name suggestions	90
9.5 Schema Overview	91
9.6 Standard questionnaires question sets	92
9.6.1 PSSUQ.....	92
9.6.2 SUS.....	92
9.6.3 UMUX.....	93
9.6.4 UMUX-LITE.....	94
9.7 SUS data.....	95
9.8 Our questionnaire question set.....	96
9.9 The answers from the small questionnaire we created	97

9.10 User habits when exploring Yggdrasil.....	98
9.11 The entire low-fidelity prototype	99
9.11.1 Wireframe.....	99
9.11.2 Map page	99
9.12 The entire high-fidelity prototype	100
9.12.1 Desktop.....	100
9.12.2 Mobile.....	102
9.13 Name suggestions for the nav links	103
9.14 All the pages of Yggdrasil	104
9.15 Lighthouse results before and after	105

1 INTRODUCTION

For our thesis we developed a website in partnership with rurAllure. The main goal of the website was to increase the engagement for the local part of the Saint Olav's pilgrimage path. Among other techniques we utilized local stories to provide more background and enrich the visitor experience. The website was made using the MERN stack due to the website's needs and our familiarity with the stack.

The MERN stack is a framework for developing websites and applications using primarily JavaScript as programming language, combined with MongoDB as a database, Express.js for backend framework, React.js for developing the user interface and the components, and lastly, Node.js for running JavaScript code outside the browser. By combining these technologies, we can effectively create up to date, advanced web applications.

RurAllure is a Horizon Europe funded project that aims to use modern information technologies to promote rural museums and heritage sites located near major European pilgrimage routes such as the Saint Olav ways here in Norway. The project aims to connect cultural heritage collections and venues to showcase Europe's cultural heritage in a broader historical and geographical context. It aims to collaborate with national and transnational associations to develop and promote cultural tourism, while fostering sustainable cooperation between museums and heritage sites (rurAllure, 2023).

As of today, rurAllure is trying to achieve these goals by having a website where they have gathered information about the project, by displaying content such as pilots, partnerships, and media. One of their projects is a "pilgrim portal" where they link to their mobile apps focused on several different pilgrimage destinations like Trondheim and Santiago de Compostela. These apps have interactive maps that contain POIs and a route planner.

Our product owner is Ole Edward Wattne who is NTNUs local member of the rurAllure project consortium. As requested by Wattne they wanted us to create a solution that utilized the stories they had gathered, via audio clips to tell local stories around the many pilgrim paths that stretches through every nook and cranny of Norway. Through this, they want to create interest in the population to explore and invest in the rural and local communities.

Further on in the thesis we will state how we developed our Yggdrasil application, our design process, how we utilized the stories and the importance of designing applications with sustainability and accessibility in mind.

1.1 SOLUTION

1.1.1 THE DIFFERENT PAGES

Now to further explain our solution. It currently exists of a landing page, an information page about pilgrimage, a map page with different marker types ranging from interesting views, official stories made by rurAllure, user created stories and other interesting locations to visit. Lastly, we have a form page where the user can upload their own content to the map.

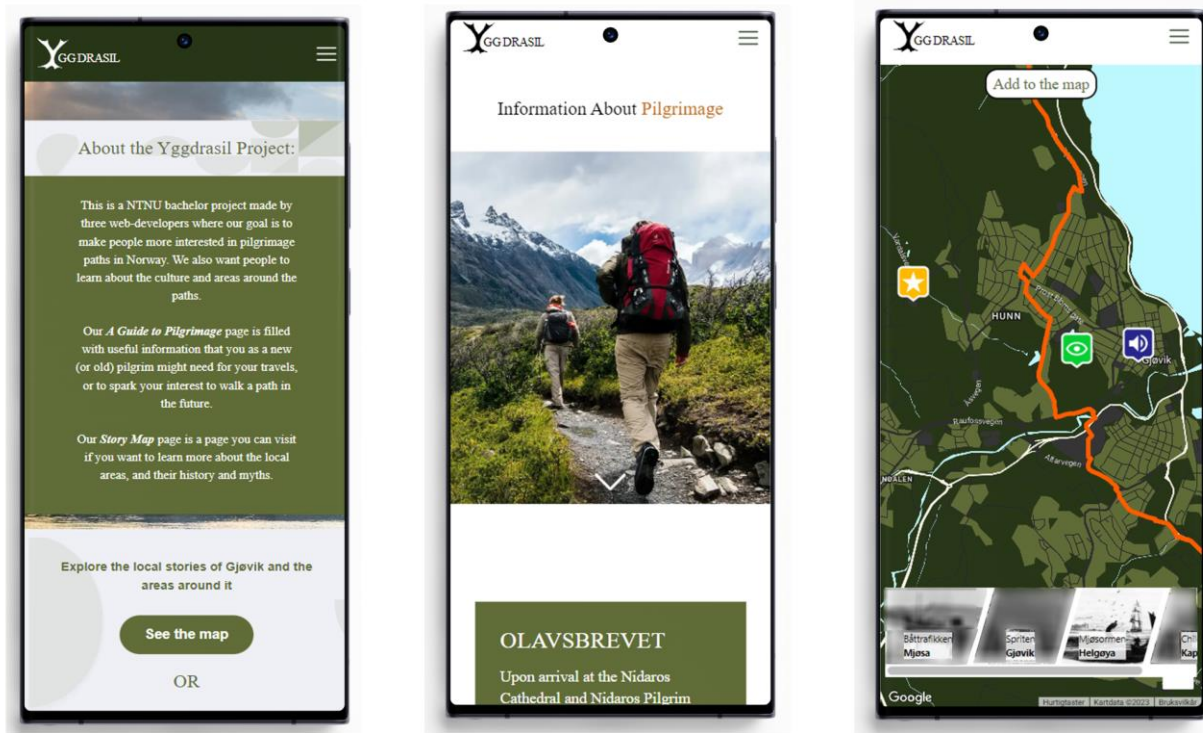


FIGURE 1 MOCKUPS FOR OUR 3 MAIN PAGES

Upon the user's initial arrival to the website, they will be automatically directed to the landing page (picture on the left). This page serves to provide users with a foundational comprehension of the website's purpose and additionally get presented with the choice to navigate to either the information or story map pages, based on which button they choose to press.

Should users opt to proceed to the information page, they will be presented with a page filled with information regarding pilgrimage. The thought behind this decision was to impart basic knowledge about pilgrimage to users, which may potentially result in higher user engagement. It should be mentioned that the information presented on this page is primarily intended to showcase the concept and is subject to change.

If the user decides to further explore the story map page, they will encounter a page that showcases a map with various types of markers. Additionally, a list detailing all the stories provided by rurAllure on the map will be presented alongside the markers. Users are granted the ability to interact with both the list and markers, and doing so will trigger the appearance of a modal. If a rurAllure story is selected the modal provides the users with information regarding the selected story, including the story itself, an audio file reading the narrative aloud, and an image carousel displaying fitting images corresponding to the story. The story “When the Sea Serpents ruled the lake Mjøsa” has a transcription accompanying the audio file.

To expand on the map feature, we created multiple ways users could add to it by submitting a form, which is accessible by clicking the “Add to the map” button on the map. This form allows users to submit either a story of their own, an interesting view or anything else that they may want to highlight for other users to see. The user is prompted to provide their email address, the coordinates of the point of interest, what type of content they would like to add, and the content itself. Once the submission has been completed and the submit button has been clicked, the addition will be sent to the database where it will have to be approved before it shows up on the map. It is assumed that quality control and approval of these additions would be the responsibility of an admin user.

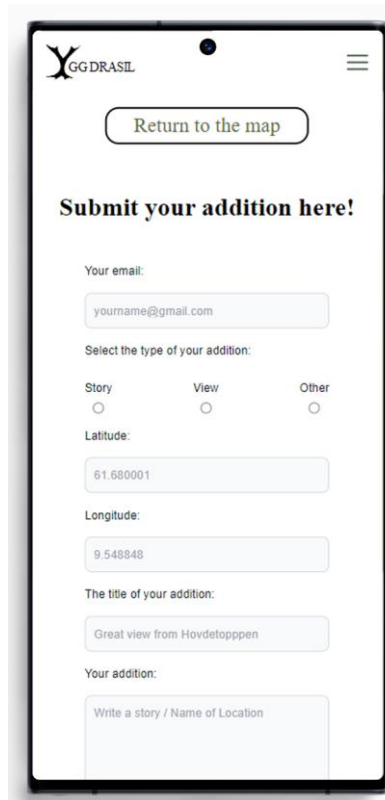


FIGURE 2 ADD TO MAP PAGE

1.2 PROBLEM STATEMENT

This is the problem statement we decided upon based on the project owner's goals as well as our own interests:

How could one design a captivating and sustainable web application that integrates a map feature, gathers helpful information for pilgrimages, and incorporates stories from Gjøvik? Would developing such an application increase interest in the topic of pilgrimage and the local history of Gjøvik?

This problem statement can be divided into three areas of concern:

1. Designing a web application that will feature a map, and a place to store helpful information for people that want to start their pilgrimage journey.
2. Develop features that increases interest in pilgrimage and the local history of Gjøvik.
3. Ensure that the application is accessible and sustainable.

1.2.1 CONTEXT FOR OUR PROBLEM STATEMENT

To grasp this thesis and its contents it is important that we elaborate on some concepts from the problem statement. The following words will be defined:

1.2.1.1 GJØVIK

Gjøvik is a city in Norway where our university lies, where the users who tested our solution are situated and where the narratives provided by rurAllure take place. We concurred that our testing participants would be interested in the stories from their local community, and it is therefore our area of focus.

1.2.1.2 PILGRIM

The concept of pilgrimage can be defined in two distinct ways. According to the Merriam-Webster Thesaurus (n.d.), a pilgrim is either *"one who journeys in foreign lands"* or *"one who travels to a shrine or holy place as a devotee."* Although the latter definition is commonly associated with the term, this thesis will primarily employ the former definition.

1.3 SHORTCOMINGS

There are some shortcomings that our solution has, that we want to bring to light.

The idea was to utilize the stories and their corresponding audio files that rurAllure were creating for this project. There were, however, some delays and we ended up not receiving the audio files in time to implement them in our solution. To combat this predicament, we recorded the stories ourselves and used that as a placeholder for the audio. This means that the quality of our product might not be to the same standard as what it could have been with the official recordings.

There is also the problem with the inconsistency of the language of the stories with some being written in English and some in Norwegian. Ideally, we would have versions of both with corresponding audio that we could toggle between.

The stories we have might not reflect how some official stories might look. Additionally, our stories are confined to Gjøvik, and therefore lack the ability to convey a story across multiple locations, which is something rurAllure might be interested in doing as users explore different areas.

The project uses the trial version of Google Cloud to access the Maps JavaScript API. This means that after 90 days or after using 3000\$ worth of credits, we would have to pay to keep our map accessible in our project.

Some of our components are not built to be expanded upon, but to work by themselves for this version of the product only. Yggdrasil does not have a login system incorporated in the design at this time, this means that all users can add new content to the map and the quality control of said content must be handled in MongoDB Atlas, instead of in the application itself.

1.4 SCOPE

The scope is crucial in ensuring that the project remains focused and that we, the project creators, have a clear understanding of what is expected of us. Our scope will be determined by the entire work process being constricted to one university semester, where we as project creators also have limited knowledge of the subjects we work with. The following list will detail the limitations we decided for the project:

1. We will utilize the stories we were given by rurAllure as is, with no rewritings.
2. We will not be using any money for any services when creating our product.
3. We will only focus on the city of Gjøvik and the surrounding area.
4. Our website will not be hosted, and we will therefore not be able to track any engagement/interest from market users, nor focus on it.
5. We are limited in our ability to gather large user testing groups and are generally restricted to only small groups of eager campus students.
6. Our solution will not be designed to substitute any already existing products on the market, but to work in tandem with them.
7. The Web Content Accessibility Guidelines (WCAG) will be considered in design choices, but not fully designed for specifically.
8. We will also try to reduce the carbon footprint of our product to the best of our abilities, but we will not prioritize it as a part of the solution.
9. We will develop our application with the MERN-stack.
10. Our solution will aim to target new pilgrims as a priority before seasoned pilgrims.

The limitations set in this project serve as a guide for the group, enabling us to prioritize our efforts and ensure that we meet the requirements of the project. The limitations also help manage expectations, preventing the project from becoming too complex or unmanageable.

1.5 THE REPORT STRUCTURE

In this section we will do our best to clarify the structure of the upcoming chapters, each of which will contribute to addressing our problem statement described in [1.2](#).

The structure of our report is inspired by the bachelor thesis of Glenn Eirik Hansen, Tom Shrier and Cornelius Ottar Sandmæl, who were the 2022s representatives from BWU to compete for “Eurekaprisen”. Their bachelor report was written thoroughly and was easy to follow as a reader. They also wrote a report that has a similar content type to ours, which made some of their structure resonate with what we wanted to write in our report (Sandmæl, Hansen, & Schrier, 2022).

Chapter 2 - Background, theory and early development features a brief section discussing the outcomes of our project in IDG3101 In-Depth Project, along with explanations of the various technologies, methods, terms, and theory we employed during the Yggdrasil project's development.

Chapter 3 - Methods provides a comprehensive account of the methodologies that were utilized to design and develop our project, building upon the insights presented in chapter [2](#). The chapter is divided into six sections: “Designing Yggdrasil”, which outlines the process by which we arrived at our design, “Developing Yggdrasil”, which describes the methods employed in the frontend, backend, and database development, “The Yggdrasil solution” which summarizes the final modifications made to the project, “Sustainable measures”, where we elaborate on what measures were done to increase our applications sustainability, “Accessibility measures”, which elaborates on steps taken to improve the project's accessibility, and lastly, “Organizing the project”, where we explain how the project was structured and managed.

Chapter 4 – Results is presented in a format that mirrors the structure of chapter [3](#) and it displays the results gathered from the methods we used. The various data we collected is presented in many ways to help give a clear visual to the insights.

Chapter 5 - Discussion is where we analyse and interpret the results presented in chapter [4](#), in addition to reflecting on the methodologies employed in chapter [3](#).

Chapter 6 – Conclusion provides a summary of the process and efforts undertaken to bring the project to fruition. Additionally, we discuss whether we have been able to answer the questions outlined in the problem statement.

2 BACKGROUND, THEORY AND EARLY DEVELOPMENT

It is important to address what led us to our current solution and look at what early steps we made in our project. Pilgrimage is very under-explored when it comes to what research has been done on it. And therefore, in this chapter, we will primarily emphasize the extensive research conducted on web technologies.

2.1 EARLY PLANNING AND PRIOR WORK

Our bachelor's project began to take shape in December 2022, as we aimed to pursue a topic that captured our interests. During the IDG3101 In-Depth Project course that semester, we collaborated with rurAllure on their vision to encourage travellers to explore local shops and communities along the St. Olav's Path. Our team produced a comprehensive report, identifying the problem statement as follows:

How can we make St. Olav's Path and the areas around it more motivating for students at NTNU?

We agreed that we wished to explore this topic more for our bachelor and started thinking of other ways to explore this. Our research in the In-Depth course revealed that many people were unfamiliar with rurAllure and the St. Olav's Path (Amundsen, Randen, & Stenbråten, 2022), which prompted us to formulate a new problem statement in [1.2 Problem Statement](#) which was meant to tackle this.

2.2 FROM CONCEPT TO FINISHED DESIGN THEORY

When developing a product or service it is important to incorporate a few concepts to ensure that the result becomes the best it can be. In this section we will include some information about what we have used along the way to get to today's design.

2.2.1 TARGET AUDIENCE

Your target audience refers to the specific group of consumers most likely to want your product or service, and therefore, the group of people who should see your ad campaigns. Target audience may be dictated by age, gender, income, location, interests or a myriad of other factors (Marketing Evolution, 2022).

Identifying and comprehending your targeted audience is vital to achieving success with your product, as it leads to numerous amounts of benefits. According to Marketing Evolutions (2022) such benefits include improved customer relationships and communication leading to increased product sales and usage.

Marketing Evolutions (2022) outlines seven methods for determining your target audience. To mention some, it is to conduct market research and identifying industry trends, analysing competitors, creating personas, and defining who does not fall within your targeted audience.

2.2.2 PERSONAS

As a designer or developer, you must design a product that is going to fit with what a user expects. When a design team is trying to implement a solution, it is important that they don't design for users who don't exist. It is then important that the team decides on a group of qualities and interests based on research, to make a hypothetical user of their product (Pesot & Plantenberg, n.d.).

Personas can be divided into multiple categories depending on the needs of the group/company (Dam & Siang, n.d.). If we want to understand what a user wants to do with a product we can utilize **goal-directed personas**. You then make a workflow that will lead to certain goals that a user might be interested in achieving. The next one is **role-based personas** which are used to find out what role a user is going to take in the overall architecture. **Engaging personas** are personas made to be less of a stereotype and more in line with a regular user. They have interests, emotions and are there to help the design team think about them just like they would do real people. Lastly there are **fictional personas** which are personas that are not made based on user research but are based on a design team previous experience with a user group. This however can lead to some flawed assumptions and is very disputed.

2.2.3 SWOT

“SWOT analysis is a technique for assessing the performance, competition, risk, and potential of a business, as well as part of a business such as a product line or division, an industry, or other entity” (Kenton, 2023).

A SWOT analysis consists of the four main components, strengths, weaknesses, opportunities, and threats. According to Kenton (2023), **strengths** are the qualities that distinguish an organization from its competitors and showcase its areas of excellence. These could include a powerful brand, a devoted customer base, a solid financial standing, innovative technology, and more.

Following up, Kenton (2023) describes **weaknesses** as factors that impede an organization from achieving its full potential. They represent areas where the business needs to enhance its capabilities to remain competitive. Such areas could include a weak brand, higher-than-average employee turnover, excessive debt, an inadequate supply chain, or insufficient capital.

Kenton (2023) describes **opportunities** as positive things outside a company that can help it be more competitive. For instance, if a country lowers tariffs, a car maker can sell more cars in a new market, leading to increased sales and a bigger share of the market.

Lastly **threats** are described by Kenton (2023) as things that can harm a company. For example, a wheat-producing company can be threatened by drought, which may damage or reduce their crop yield.

2.2.4 USABILITY COMPETITIVE ANALYSIS

Unlike traditional competitive analysis, a usability competitive analysis is centred around the user experience of a product. The features, user interface and overall usability is important parts of a usability competitive analysis. The goal of a competitive analysis is assessing the strengths and weaknesses of competitors to improve your product (Baxter, Courage, & Caine, 2015, p. 32).

2.2.5 MOBILE FIRST DESIGN

Mobile first design thinking is an approach to web design that focuses on creating a better user experience on mobile devices. This is important because more than half (54%) of all web traffic now comes from mobile devices, according to Unadkat (2022).

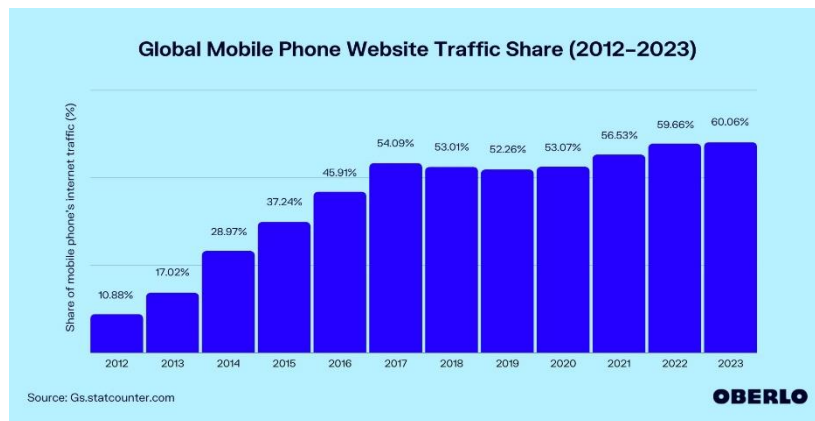


FIGURE 3 [OVERVIEW OF THE PERCENTAGE OF INTERNET TRAFFIC THAT COMES FROM MOBILE] (OBERLO, N.D.)

Unadkat (2022) argues that designing for mobile devices is important because smaller screens have limited space, and design teams must ensure that the most important elements of the website are clearly visible to users.

The way this design approach is utilized is to make designers begin developing for mobile devices first, and then gradually expand to larger screen sizes rather than the traditional approach of designing for desktop first. This process encourages the removal of any unnecessary design elements that may hinder website rendering and navigation, resulting in a smoother user experience. In addition, having a mobile-friendly website can enhance its search engine optimization (SEO), due to Google gives priority to such websites in its search results (PageFly Academy, u.d.).

2.2.6 USER TESTING

2.2.6.1 THE DIFFERENT USER TESTING METHODS

User testing is a research method used to assess the usability, functionality, and overall user experience of the product. This involves observing and collecting feedback from users as they interact with the product, the reason being to find any issues or improvement areas.

There are different types of user testing, this includes usability testing (UX research) and surveys among other things.

A survey is one of the types of user testing that enables you to obtain feedback from your users about your product. Generally, the user will give constructive criticism as feedback, which can be used as material for improving your product. As Omnicconvert (2023) expresses

in their article, surveys are one of the easiest ways to gather data, as users can complete surveys through any device and from anywhere.

When testing your product, you can also test its usability. In usability testing, we confirm that the website fulfills the needs of users. An example of a usability test could be evaluating the design of a website's mobile version to find out if it lacks some critical components in its layout.

Observation is a very effective method to test your product's usability. The method involves giving users a task or instructions, and then observing the user's initial actions, without interfering. By observing the participant and taking notes, the researcher can use the gathered data to improve the product.

Since usability testing both can be done remotely and in-person, there are different ways of testing, however there will be some aspects you will have to sacrifice. For example, you could observe remotely, although by doing this, you would not be able to observe body language and the facial expressions of the participant. However, by doing testing remotely, a larger quantity of user tests can be done, as this will be more accessible for more users (Omniconvert, 2023).

2.2.6.2 BIAS

When planning to conduct user testing with a survey it is important to consider how different biases can affect the outcome. According to Baxter et al. (2015, p. 267) systematically excluding certain units from a dataset, known as selection bias, can result in less accurate data. This type of bias can occur when resorting to convenience sampling while looking for users to test. Convenience sampling is when you recruit people for testing based on your convenience.

Another bias to be aware of is the Nonresponse bias. Not everyone is going to be willing to answer a survey. The people who are willing to participate in a survey may be very different than those who don't (Baxter, Courage, & Caine, 2015, p. 268).

Social desirability bias refers to the tendency some participants have to give responses that they perceive as more socially acceptable or desirable, rather than the truth (Baxter, Courage, & Caine, 2015, p. 237).

2.2.7 QUESTIONNAIRE

A questionnaire is a form that you can make users answer to gather certain information from them. It is usually in the form of multiple choice, with either different alternatives or a scale of numbers, but can have open ended questions occasionally (Sauro & Lewis, 2016, p. 185).

2.2.7.1 STANDARDIZED QUESTIONNAIRES

As said by Sauro & Lewis (2016, p. 185) a standardized questionnaire is defined as a questionnaire that is designed to be used many times in many different circumstances. It usually has a predetermined list of questions that follow a given order and/or format. They usually also come with a set of rules on how to gather/visualize the user data. It is normal for designers to report on measurements such as reliability, validity, and sensitivity when developing a standardized questionnaire.

There are many types of standardized questionnaires. The main difference between them is their length, types of questions and how many steps a user can choose between when answering the questions. Some are more geared towards users that have just finished tasks as a part of their testing, while others are more geared toward people finished testing a product to gather their impressions of its entirety (Panagiotidi, 2021). Sauro & Lewis (2016, p. 186) gives some examples of these post study questionnaires:

- QUIS – Questionnaire for User Interaction Satisfaction
- SUMI – Software Usability Measurement Inventory
- PSSUQ – Post-Study System Usability Questionnaire
- SUS – System Usability Scale
- UMUX/UMUX-LITE – Usability Metric for user Experience

The one we used for this project is SUS which we will explain in detail in the following section.

2.2.7.2 THE SYSTEM USABILITY SCALE (SUS)

The System Usability Scale is a widely adopted, standardized questionnaire that assesses users' perception of the usability of a system at the end of a study. It is frequently referenced in national and international standards (ANSI, cited in Sauro & Lewis, 2016, p. 186).

It consists of ten items, each rated on a five-point scale. The odd-numbered items are positively worded, while the even-numbered items are negatively worded. To calculate the SUS score, participants must respond to all ten items. The first step is to determine the score contribution for each item, which ranges from 0 to 4. For positive items, the score contribution is the scale position minus 1, while for negative items, it is 5 minus the scale position. The overall SUS score is obtained by multiplying the sum of the item score contributions by 2.5 (Sauro & Lewis, 2016, pp. 198-199).

2.2.8 PROTOTYPING

A prototype is used to convey the intent of a design and to act as a simulation of the final product's interactions between the user and end result of the design. It is normal to conduct prototyping in the early times of the project designing when the stakes are low, and the risk of bad feedback will not ruin or set back a product too much. A prototype can be divided into two categories based on its fidelity, which is its realness compared to the product it is simulating: Low-fidelity and High-fidelity (Babich, 2017). Let us take a look at low-fidelity first.

2.2.8.1 LOW FIDELITY PROTOTYPING (LO-FI)

Low-fidelity prototyping can be used in the early stages of product development as a valuable design technique to test and refine ideas. The ideas can be tested quickly and with little to no cost. According to an article by Usability.gov (n.d.), low-fidelity prototypes involve creating representations that can be incomplete and rough, of a design idea, typically using paper, sketching tools, or basic digital software. These prototypes are used to test basic functionalities, gather user feedback, and identify potential issues before investing time and resources into creating a more polished product.

Low-fidelity prototypes can be especially useful in situations where a project's requirements are still being defined. It can also be useful where there is a high level of uncertainty about what features or design elements will be most effective. By allowing designers to quickly iterate on ideas and incorporate feedback from users, low-fidelity prototyping can help improve the overall user experience and increase the likelihood of a successful final product (Babich, 2017).

Now let us look at the opposite of lo-fi, hi-fi in the next section.

2.2.8.2 HIGH FIDELITY PROTOTYPING (HI-FI)

What separates hi-fi from lo-fi is that while a lo-fi prototype will be as simple as possible, containing only a simple layout with minimal styling, a hi-fi prototype will be as similar to the final product as possible (Babich, 2017).

A prototype designed with high fidelity will usually be created digitally and encompasses both the user interface (UI) and the user experience (UX) (Ibragimova, 2016). Because of its similarity to the end product, it is easier to gather useful data regarding how users might interact with the interface. This is also good for stakeholders because it gives them a good idea of what features a product will have and how the features will work with one another. It is, however, more time consuming and more expensive to create (Babich, 2017).

It is best to use a hi-fi if you have a plan for how the animations and transitions between elements are going to work. It is also great for showcasing your theme and color palette.

2.2.9 MERN

The MERN stack uses MongoDB, Express, React and Node to build web applications. The technologies in the MERN stack are JavaScript based and when used together, they make up a simple stack for web development (Hoque, 2020, p. 51).

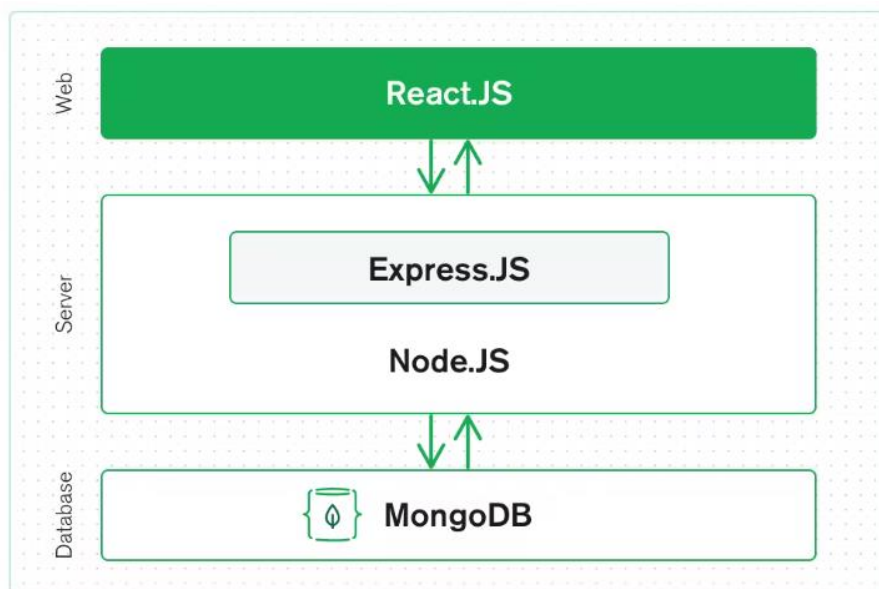


FIGURE 4 [THE MERN STACK ARCHITECTURE VISUALIZED]. (MONGODB, N.D.-A)

2.2.9.1 FRONTEND, REACT.JS

The MERN stack's frontend is developed using React.js, which allows you to create individual components that can be combined to form various parts of your application. These components can be rendered as HTML and linked to your backend server. React.js' strength lies in its ability to handle data-driven, stateful interfaces with relatively little code and minimal complexity. Additionally, it offers features such as error handling, event support, and form support to aid your development efforts (MongoDB, n.d.-a).

2.2.9.2 BACKEND, EXPRESS AND NODE.JS

On the backend of the architecture, we have Express.js and Node.js. Express.js is a lightweight, high-speed web framework that runs within a Node.js server. It provides capabilities for handling HTTP requests and responses, as well as URL routing. To integrate your React.js frontend with your backend built on Express.js and Node.js, you can use both XML HTTP Requests and conventional GET/POST requests to interact with the Express.js functions that drive your application (MongoDB, n.d.-a)

2.2.9.3 DATABASE, MONGODB

Lastly, we have the database part of the architecture. In the MERN stack, MongoDB is the database being utilized. MongoDB is a popular database system that can store and retrieve data. It uses a flexible schema that enables storing data in a document format instead of the traditional tables and rows format that other databases like MySQL use. The documents are stored and can be accessed as JSON-like objects. This approach makes MongoDB databases scalable and makes it easier to work with intricate data structures (MongoDB, n.d.-b)

2.2.10 CSS FRAMEWORK

2.2.10.1 TAILWIND CSS

Tailwind CSS is a utility-first CSS framework that streamlines web development by providing a comprehensive set of pre-built CSS classes that can be easily combined to create custom user interfaces. Unlike traditional CSS frameworks, Tailwind CSS is designed to be highly customizable and flexible, enabling developers to quickly create modern, responsive designs that meet the needs of their specific project. With Tailwind CSS, developers can avoid writing custom CSS code by hand, reducing the amount of time and effort required to create visually stunning websites and applications. This framework has gained popularity among developers for its intuitive naming conventions, its focus on rapid prototyping, and its ability to facilitate a component-driven development approach (Tailwind Labs Inc., u.d.).

While there are numerous benefits of utilizing the Tailwind framework, it is not without its drawbacks. One such disadvantage is that the application of pre-defined classes to different HTML elements, can lead to an increase in the length and size of the HTML code.

Additionally, some developers may find it challenging, restricting, or clumsy utilizing already pre-defined classes and may prefer the flexibility of writing own custom CSS classes. Despite this, the benefits outweigh the drawbacks, especially in larger projects, making the use of Tailwind CSS highly worthwhile (Tailwind Labs Inc., u.d.).

2.2.11 GOOGLE MAPS JAVASCRIPT API

The Google Maps JavaScript API enables the creation of personalized web app maps, location services, and geospatial interactions with dynamic and interactive features (Google, 2023).

2.2.12 DEPENDENCIES – NODE PACKAGES

This section will explain some of the dependencies we used in our project. NPM was utilized to install these dependencies.

NPM is responsible for managing Node.js packages, and it is included in the installation of Node.js. The packages managed by NPM contain all the necessary files for running a particular module. In this context, module refers to different JavaScript libraries that provide different types of functionalities based on a project's needs (W3schools, u.d.).

2.2.12.1 *REACT FEATHER*

React Feather is an open-source collection of icons. The icons are customizable. You can change the colour of the icons, stroke width and size in pixels.

2.2.12.2 *ANIMATE.CSS*

Animate.css is a pre-defined library of CSS animations.

2.2.12.3 *MATERIAL UI*

The open-source Material UI is a React component library that follows Google's Material design and offers prebuilt components for use in production. The components are customizable, and this facilitates personalization of the already beautiful design (Material UI , 2023).

2.2.12.4 AXIOS

According to Sophie Bates (2020), Axios is a promise-based HTTP library that enables developers to send requests to either their own server or a third-party server to retrieve data. In simpler terms, Axios is helpful when creating GET, POST, and DELETE requests.

2.2.12.5 REACT GOOGLE MAPS API

The react-google-maps/api node package offers easy-to-use bindings to the Google Maps API ([2.2.11](#)), allowing you to incorporate it as React components in your application (React Google Maps Api, u.d.).

2.2.12.6 REACT H5 MEDIA PLAYER

The react-h5-audio-player is a node package that allows you to create consistent audio player components for your website. It facilitates making high-quality audio player that seamlessly integrates with your website and enhances the user experience (Luo, 2022).

2.2.12.7 REACT ROUTER

The React Router package enables client-side routing, which allows for faster and more dynamic user experiences by updating the URL and rendering new UI without requesting another document from the server. This results in faster load times and enables features such as animation (React Router, u.d.).

2.2.12.8 MONGOOSE

Mongoose is a library that provides an Object Data Modelling solution for MongoDB in Node.js. Mongoose's main objective is to enforce a predefined schema at the application level. This is to ensure consistency in the data structure. In addition to the main objective, Mongoose offers other helpful features like hooks and model validation as well as others to make it easier to work with MongoDB databases (Kukic & Vlaeva, 2023).

2.2.12.9 BODY PARSER

Express Body-Parser is used to process data sent in an HTTP request body. It provides four express middlewares for parsing JSON, Text, URL-encoded, and raw data sets over an HTTP request body. Before the target controller receives an incoming request, these middleware routines handle it (Sharma, 2023).

2.2.12.10 FOCUS-TRAP-REACT

Focus-Trap-React is a React component designed to capture and restrict focus. This component is beneficial for making accessible modals as it simplifies navigation for users who rely on screen readers (Karakaya, 2023).

2.2.12.11 MULTER

Multer is a node.js middleware which is primarily used for uploading files (ExpressJS, u.d.). It can for example be used to handle the files uploaded in an HTML form via the “file” input type.

2.2.13 ACCESSIBILITY

Explained by Web Accessibility Initiative (2023), web accessibility refers to the practice of designing and developing websites, tools, and technologies in a way that allows people with disabilities to use them effectively. This involves enabling individuals to perceive, understand, navigate, and interact with the content on the web, as well as to contribute to it. There are different helping tools that a developer can use, such as wave and lighthouse.

2.2.13.1 DISABILITIES

Designing for accessibility means developing content that can be accessed and used by individuals with a range of disabilities as identified by the Web Accessibility Initiative (2023) including auditory, cognitive, neurological, physical, speech, and visual impairments. It's important to note, however, that web accessibility can also have advantages for people who don't have disabilities.

2.2.13.2 WCAG - WEB CONTENT ACCESSIBILITY GUIDELINES

WCAG is a reputable technical standard guideline. The W3C make the guidelines to make web content more accessible to all people, with an extra focus on the people who rely on assistive technologies to interact with websites. WCAG has 12 guidelines grouped into 4 principles: Perceivable, Operable, Understandable, and Robust. Each principle has 3 levels (A, AA, AAA), with A being the minimum and AAA being the maximum level of compliance to web accessibility standards. (Ritter & Winterbottom, 2017, p. 263)

2.2.13.3 WAVE - WEB ACCESSIBILITY EVALUATION TOOLS

WAVE is a collection of evaluation tools that help developers in making their content on the web more accessible to people with disabilities. WAVE helps to detect accessibility and WCAG errors, while also enabling manual evaluation of web content (WAVE, u.d.).

2.2.13.4 GOOGLE LIGHTHOUSE

Google Lighthouse is an open-source tool that automatically improves web page quality. It can be used on any web page, whether public or requiring authentication. Lighthouse offers audits for various aspects, such as performance, accessibility, progressive web apps, SEO, and more (Google Chrome, 2022).

2.2.13.5 TAB INDEX

Tab index is an attribute that can be given to html elements to allow for keyboard users to navigate to them or disallow them in certain circumstances. The former is done by given an element the index of 0, 1 (or any other positive number), and the latter by giving the element a negative tab index. Tab index is important for people with visual or motor control disabilities that are reliant on the use of a screen reader to understand/navigate the contents of internet pages (Bailey, 2021).

2.2.13.6 ARIA

ARIA, which stands for Accessible Rich Internet Applications, is a collection of roles and attributes that outline techniques for improving web applications and other online content to be more accessible for individuals with disabilities. It expands the functionality of HTML by enabling assistive technologies to access interactions and widgets where no other mechanism exists. ARIA allows for the creation of accessible JavaScript widgets, form hints and error messages, live content updates, and additional features (MDN, 2023a).

2.2.14 SUSTAINABILITY

“Sustainable web design is an approach to designing web services that put people and the planet first. It delivers digital products and services that respect the principles of the Sustainable Web Manifesto: clean, efficient, open, honest, regenerative, and resilient.” (Vassilyovska , 2022).

Developing sustainably is a measure adopted by developers to decrease the carbon footprint of their web applications.

2.2.14.1 CODE SPLITTING

Code splitting is the process of splitting larger chunks of code, into smaller segments which can be loaded on demand by using lazy load, according to the Binge on code team (2021).

2.2.14.2 LAZY LOADING

As per the article authored by the Bing On code team (2021), lazy loading is an extension of code splitting that enables loading of different segments only when they are needed. This leads to reduced page load times and improved website performance.

2.2.14.3 FILE TYPES AND FORMATS

When discussing files, it is important to understand the distinction between file types and file formats. File types refer to categories of files such as audio files, image files, and others. On the other hand, file formats such as MP3, WAV, and JPG specify the way in which data is encoded and stored within a file.

There are many file formats that are optimized for different use cases. Therefore, it is important to consider what you want to achieve when choosing a format. Several image formats are available, such as APNG, AVIF, GIF, JPEG, PNG, SVG, and WebP. The most relevant for this project are JPEG, PNG, SVG, and WebP formats, which are further elaborated below.

Joint Photographic Experts Group image (JPEG) is mostly used as a good choice for lossy compression of still images and is according to MDN (2023c), the most popular image format.

Portable Network Graphics (PNG) is according to MDN (2023c) a preference over JPEG when you want a more precise reproduction of source images, or when transparency is needed. PNG also has a noticeably larger file size compared to something like JPEG and WebP.

Scalable Vector Graphics (SVG) are images displayed in a vector format. The use of SVG's is ideal for user interface elements such as icons, diagrams and other elements that needs to be drawn accurately according to MDN (2023c).

Web Picture format (WebP) is an image file format developed by Google. WebP is an excellent option for both static and animated images. It provides significantly superior compression compared to PNG or JPEG, while also supporting higher colour depths, animated frames, and transparency MDN (2023c).

The WebP format is more efficient for uncompressed photos compared to other formats, such as JPEG, PNG, and AVIF. Specifically, it can achieve 1.42x compression compared to JPEG, 1.70x compared to PNG, and 1.75x compared to AVIF (Broz, 2022).

Regarding audio formats, there are three main categories: lossy, lossless, and uncompressed. Examples of each format include MP3 for lossy, WAV for uncompressed, and FLAC for lossless.

MP3 is the most popular of the lossy formats, and work on most devices. According to Adobe (u.d.), the files can be as small as one-tenth the size of a lossless file such as FLAC. Adobe also mentions that the MP3 format usually is fine for the consumer since the drop in quality is inaudible.

Waveform Audio File (WAV) retains all its original data and is ideal for sound engineers according to Adobe (u.d.).

Free Lossless Audio Codec (FLAC) is as its name implies, a lossless file format.

2.2.14.4 REDUNDANCY IN CODE

Redundant code refers to code that has been written but is not being utilized. This results in more code being loaded on a website, which in turn negatively impacts the website's performance.

3 METHODS

In this chapter we will explain how we utilized the different methods described in [chapter 2](#). We will explain how we used them to create Yggdrasil. The first part will describe how we created the design, and the second will talk about how we developed the application.

3.1 DESIGNING YGGDRASIL

In this section you can expect to learn about how the design came to be and how we conducted our testing of the design with the use of prototypes.

3.1.1 TARGET AUDIENCE

For this project we identified two groups within our target audience (*see [2.2.1](#)*) that we want to aim our solution towards. The first group we wanted to target was people who want to, or are thinking about traveling a pilgrim path, but lack sufficient experience, and the proper knowledge, to feel comfortable doing so. The second group are people who want to learn more about their local communities and share their knowledge of their homestead with others.

The project also aims for these target groups in the Gjøvik area at this time but will conceptually work the same for other areas as our solution expands to other areas.

The target audience is crucial for us to create insightful personas which we will talk about in the next section.

3.1.2 PERSONAS

We created 4 different role-based personas (*see [section 2.2.2](#) for definition*), that would each personify a user role for our website. We divided them based on the different capabilities of our site. One would be interested in stories, one interested in learning more about pilgrimage, one wanting to share their knowledge with their community and one wanting to share great spots to visit.

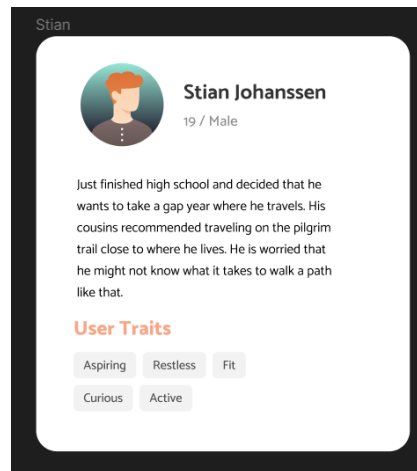


FIGURE 5 EXAMPLE OF HOW OUR PERSONAS LOOK

See appendix [9.1](#) for the rest of the personas.

3.1.3 USABILITY COMPETITIVE ANALYSIS

During the design phase, it was necessary to explore how other companies operating in the same marketplace had designed their products. The aim was to identify the positive aspects that attracted users to their products, and the features that negatively impacted their performance. This was done to gain inspiration for our own project, and to understand what might not be effective. The research provided valuable insights into successful design strategies that were adopted in the market, which helped guide the development of our solution.

See results in [4.1.1.2](#)

3.1.3.1 FINDING COMPETITORS

The first hurdle was to find what other companies existed that worked with pilgrimage and that were developing products that could compete with our own for the target audience. To achieve this, we established a list of prerequisites that potential competitors would need to satisfy to be considered. We conducted Google searches to identify companies that operate in the same domain as us and then evaluated them against our criteria. Eventually, we identified three potential competitors - rurAllure, Visit Norway, and Pilegrimsleden.no. In reality, these are collaborators that work together and share resources to reach common goals, but for the

sake of the competitive analysis we call them competitors. To compare them, we used Figma to create a table with the criteria and possible competitors.

Possible Competitors			
Criteria	rurAllure	Visit Norway	Pilegrimsleden.no
Map with POIs	YES	YES	YES
Information about pilgrimage in Norway	YES	YES	YES
Focuses mainly on pilgrimage	YES	NO	YES
Chosen as competitor?	YES	NO	YES

FIGURE 6 TABLE SHOWING THE COMPARISONS BETWEEN POSSIBLE COMPETITORS

The comparison showed that all three potential competitors met most of the criteria, except for Visit Norway which was excluded due to their broad focus instead of being specific to pilgrimage. Consequently, rurAllure and Pilegrimsleden.no were selected as our competitors.

3.1.3.2 SWOT

We took the competitors mentioned prior and analysed their websites to gather insights into what we could do for our solution to be able to compete with what they have. We then filled in a SWOT diagram to visualize their strengths and weaknesses to see if there were anything that we could learn from them and improve upon.

See results in [4.1.1.1](#)

3.1.4 LOW FIDELITY PROTOTYPING (LO-FI)

3.1.4.1 WIREFRAME

For the first iteration of our lo-fi prototype, we created a wireframe to outline the contents of each page for our solution. We designed three distinct pages for our website, each with different content. The landing page was intended to introduce our website and give users an idea of what they could expect from it. The second page contained a map with points of interest and accompanying sound clips. Lastly, we designed a page that would provide useful information for new pilgrims interested in walking a path in the future.

By creating a wireframe that established the structure of our pages, we were able to clearly convey our intentions without becoming too preoccupied with design details. We utilized Figma, a tool that simplifies collaboration on prototypes, to create basic squares representing each page of our solution. As a team, we brainstormed ideas for each page and kept them as straightforward as possible for clarity. This stage ensured that our vision and other design details were aligned, and differences in ideas were not critical until later stages of development.

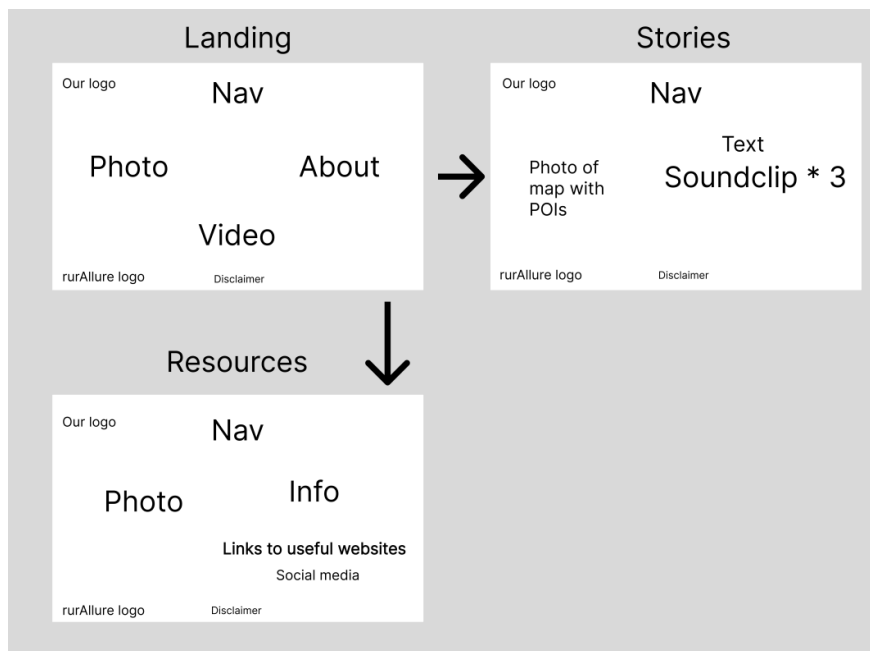


FIGURE 7 FIRST ITERATION "WIREFRAME" LOW FIDELITY PROTOTYPE

3.1.4.2 CREATING A LOW FIDELITY OF THE MAP IDEA

We continued with a second iteration of creating low fidelity. This iteration was also done digitally in Figma and included only the map page with story POIs.

As you can see in figure 8 and 9 below our design encompasses the visual appearance of the map, including its markers and list of stories as well as the modals behaviour when opened.

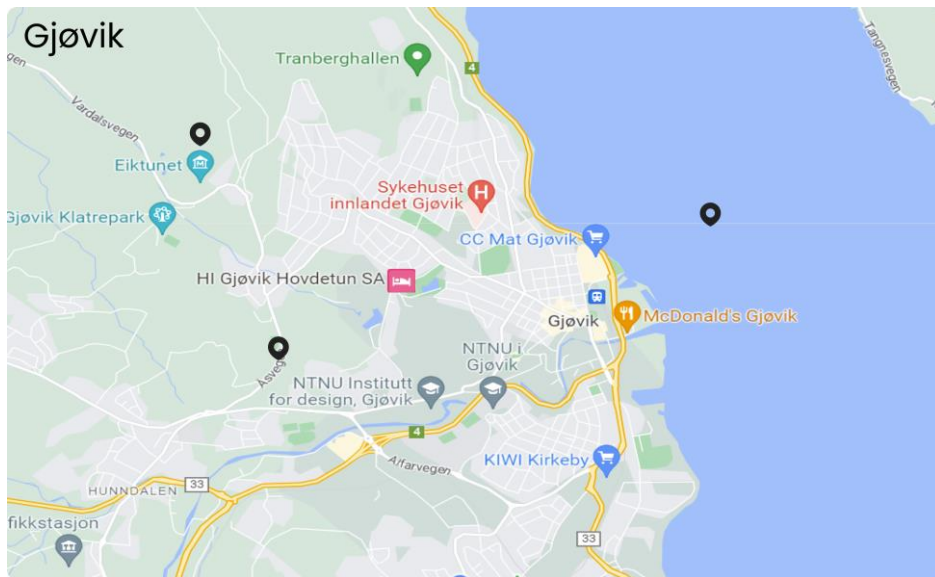


FIGURE 8 LOW FIDELITY VERSION OF OUR MAP PAGE

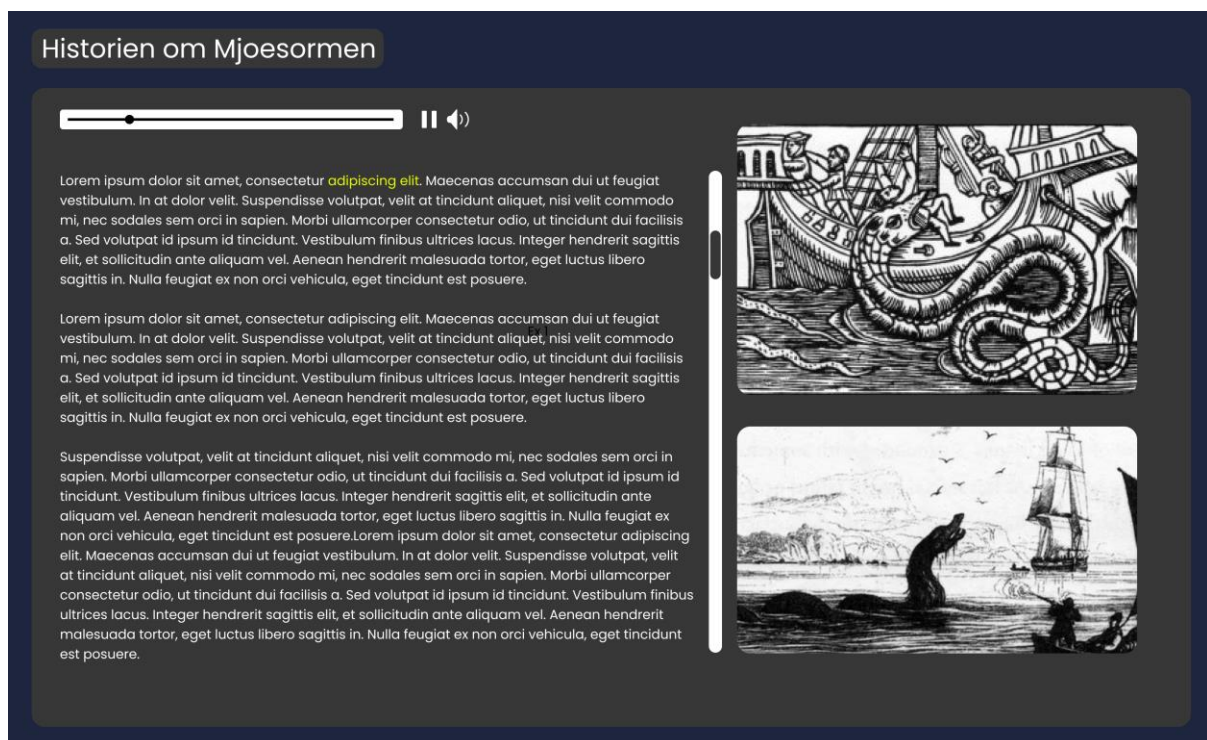


FIGURE 9 LOW FIDELITY PROTOTYPE OF MAP MODAL

3.1.4.3 LO-FI PROTOTYPE USER TESTING

After creating a low fidelity prototype of the map page format, we chose to test it with NTNU students. We believed that feedback on the map page was crucial, so we conducted user testing with only the map page as a low fidelity prototype. To recruit participants, we focused

mainly on our fellow students as they were the easiest group to contact and the most efficient to work with. We posted an announcement about testing our prototype in the "BWU-forum" team on Microsoft Teams, which is a group for web development students from first to third year.

The following day, we reviewed the responses to our announcement and discovered that seven students were willing to participate in the user testing. Our next step was to create a schedule for each user test and prepare a document with specific questions about the design that we wanted to receive feedback on.

We presented the users with various concepts for a solution, which were created and evaluated in Figma. Afterwards, we gathered feedback on their preferred elements and requested suggestions for additional features.

The valuable insights we gathered from the low fidelity user testing led us to make changes in our design. These changes became the hi-fi which we will talk about in the next section.

See [4.1.2.1](#) for results.

3.1.5 HIGH FIDELITY PROTOTYPING (HI-FI)

After successfully testing the low-fidelity prototype with users, our next step was to create a high-fidelity prototype for the entire website. Since we only designed the desktop map page in the low-fidelity prototype, we needed to create designs for both the landing page and the information page in the high-fidelity prototype, for both desktop and mobile versions.

3.1.5.1 CREATING THE HI-FI

We used the feedback from the low fidelity and improved the solution. Some of the feedback was conflicting (See [appendix 9.3](#)), which led us to try to explore them as different variations of a solution.

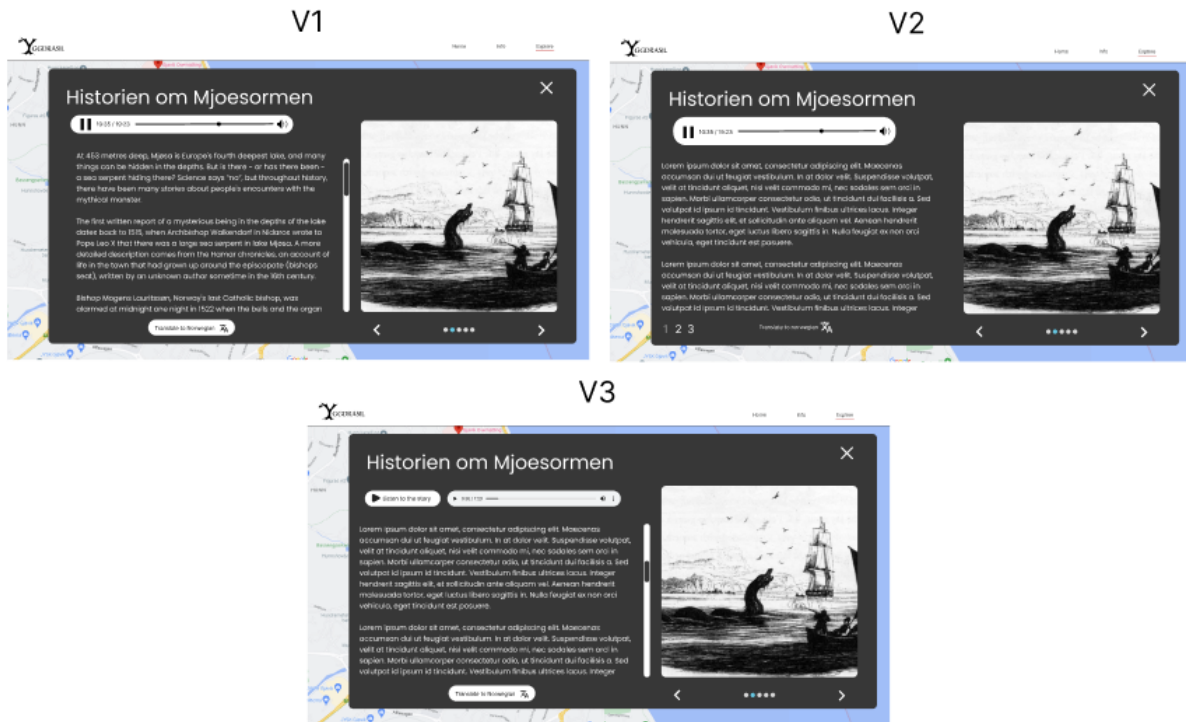


FIGURE 10 THE DIFFERENT VERSIONS OF THE MODAL "RURALLURE STORY" IN THE HI-FI PROTOTYPE

The navbar was revised with new names for our information page and map page. To determine alternative names for "info" and "explore", we held a brainstorming session to generate a list of potential options. Please refer to the appendix [9.13](#) for a complete list of suggestions.

Following the brainstorming session, we conducted a vote to select the best suggestions, ultimately resulting in the new names "A Guide to Pilgrimage" to replace the "Info" page, and "Story Map" to replace the "Explore" page. See figure 11 for the iteration.

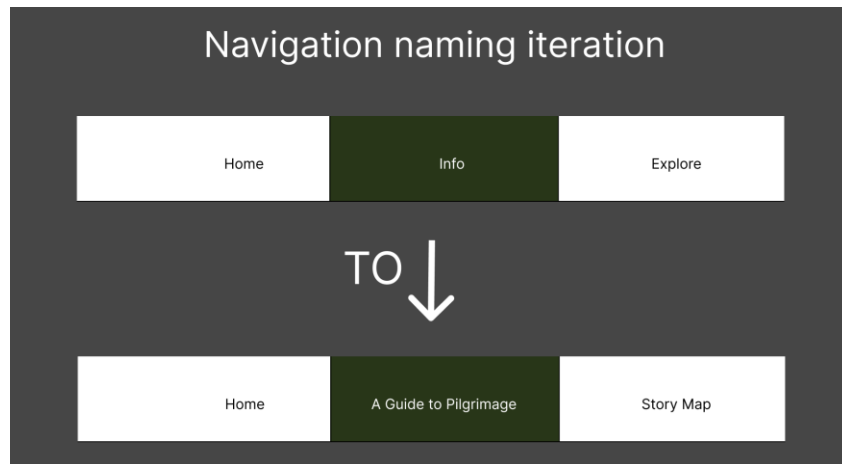


FIGURE 11 THE ITERATION ON OUR NAVIGATION BAR

The hi-fi evolved on the design aspects and smoothed out the interactions. We moved the stories on the map to a modal that could be opened in the middle of the screen vs on the side which we had in the lo-fi. We also fleshed out some of the actual contents and design aspects of the other pages at this stage, both on the information page and on the homepage, also including the map page itself.

3.1.5.2 HI-FI USER TESTING

The hi-fi testing began with an observation-based user test (*See section 2.2.6.1 for definition*). The observation of the tasks given allows us to study how the users navigate the architecture. If we observe many users struggling to complete a certain task we give them, it shows us that we have to change the design and interactions to make it more intuitive.

Following the observation phase, we asked the participants to share their thoughts on each task. The testing for the high-fidelity prototype ended similarly to the lo-fi. We asked the participants some questions regarding the design elements to get a feel on what design elements needed more tweaking in both the mobile and desktop designs.

See [4.1.3.1](#) for results.

3.1.5.3 LOGO AND BRAND NAME

When the other changes were complete, we needed a brand for the users to connect with our application. This meant coming up with a name for our application and a simple logo that would fit with the name and be recognizable.

A session of brainstorming was conducted, and we came up with several suggestions for the brand name. The group resonated with some suggestions more than others.

These are some of our brand name suggestions we liked best (*see appendix [9.4](#) for all the names*):

- ***Fjodr***
- ***Journey***
- ***DiscoverRurality***
- ***NorseRoutes***
- ***Sojourn***

We wanted the name to have a symbolic meaning and be tied to Norwegian culture. A wish was that the name also could be a symbol of the many pilgrimage paths Norway has. The name that we found the most fitting with these ideas in mind was “Yggdrasil”.

“What is Yggdrasil?” you might ask. “According to Norse mythology, all Nine Realms of the cosmos either hung from its branches, or else grew from its massive roots.” (Apel, 2022). In other words, Yggdrasil is the world tree in Norse mythology.

The reasoning behind our name choice was because we found that the tree had interesting cultural significance and we wished to utilize its branches as a metaphor for the diverse routes one can take when exploring the pilgrim routes and local areas.

Once we had settled on the name, the next task was to create Yggdrasil’s logo. As mentioned earlier, we wanted to utilize its branches, and that is precisely what we did. We had multiple designs, each with its unique features, however we quickly came to an agreement on the most fitting one. The logo was created in Adobe Illustrator.



FIGURE 12 FINAL DESIGN OF YGGDRASIL'S LOGO

The logo features a simplistic drawing of the Yggdrasil tree as the “Y” letter in our logo, with branches dividing to enhance the metaphor. The goal was to create a simple, yet effective logo that would blend in with our website’s design without being overly distracting.

3.2 DEVELOPING YGGDRASIL

Here you can find how we built the application, and how it was divided into our frontend, backend and database. We will describe the methods we used to make our app and paint the picture of why it was done this way.



FIGURE 13 THE PROJECT FOLDER STRUCTURE OF MY-APP

3.2.1 FRONTEND

We developed our application in React, which means that it runs as a single page application. A single page application means that the browser only has to load one page and then updates the body content when the user clicks around (MDN, 2023b).

We created the starting point for our application by writing “`npx create-react-app my-app`” in the terminal. This creates a folder called `my-app` and sets up the basis for our frontend structure within it. Anything we add after will be on top of that structure.

3.2.1.1 FRONTEND STRUCTURE

Our frontend has a two-folder file structure. It is divided into the “public” folder and the “src” folder. To help visualize the structure we made figure 14 below.

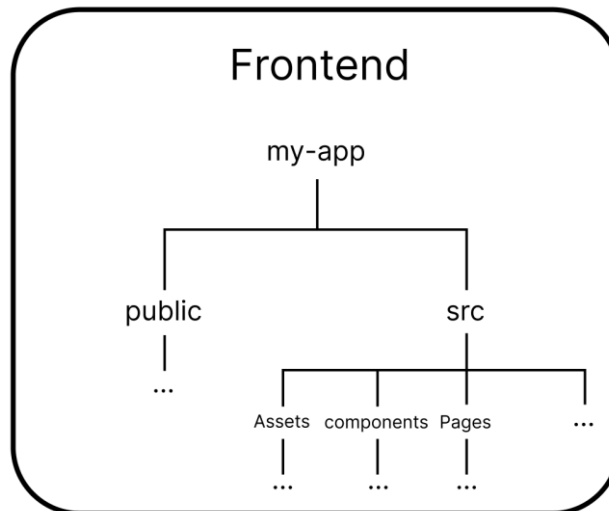


FIGURE 14 STRUCTURE OF THE FRONTEND

The “public” folder contains static assets like images, that do not require any processing. The folder also contains the “index.html” file, which is the entry point for the entire application. Additionally, it includes other compiled files that is read by the browser.

The “src” folder holds all the visible code that the user can access, with various subfolders like "assets" for images, audio, and SVG files, "components" for our React application, and pages that the user can browse through.

The “src” subfolder called “Assets” contains all the different assets we have used on our website, such as audio files, images, transcript files and SVGs.

The next “src” subfolder “components” is as its name suggests, the folder that contains various components used throughout the website's pages. These components serve as the fundamental building blocks of the website.

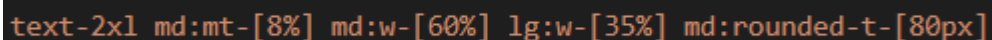
The last “src” subfolder “pages” contains files for each page that the user can navigate to. Each page file incorporates multiple components that together create the layout for that specific page.

For a complete overview of the project structure, see figure 13.

3.2.1.2 DEVELOPING MOBILE FIRST

As we mentioned in [2.2.5](#), mobile first design is an important part of the design and development of applications in the present day, where the majority of people own a smart phone. We utilized this design thinking in the development of the website. Every page was styled for the mobile design first and then the styling was adjusted when the device resolution got larger.

We utilized the Tailwind CSS library (*see [2.2.10.1](#)*) for our styling, which uses custom class names to style elements. You can then give each class name a prefix which determines when it will be used such as “hover:” and “active:”. These prefixes act similarly to selectors in CSS. In the library they also include prefixes that act as media queries. You can add “md:”, “lg:” or “xl:” in front to make the styling only relevant for medium, large, or extra-large screen sizes respectfully. That means that styling without these prefixes will be default and for mobile designs, and that we had to style the other resolutions differently later. This ensured that the mobile design was developed first and thoroughly.



```
text-2xl md:mt-[8%] md:w-[60%] lg:w-[35%] md:rounded-t-[80px]
```

FIGURE 15 EXAMPLE OF THE USE OF MEDIA QUERY PREFIXES

3.2.2 BACKEND

We utilized Express.js and node.js to establish the backend (*see section [2.2.9.2](#)*). The primary responsibility of the backend was to retrieve data from the database for use in the frontend of the project, as well as to store submitted user data in the appropriate collection. For the backend to accomplish this it utilizes POST and GET requests in combination with Mongoose, Axios, body-parser and Multer.

(See the dependencies section [2.2.12](#) for more info)

3.2.2.1 BACKEND STRUCTURE

The backend of the Yggdrasil project consists of “server.js”, the “models” folder and the “userImages” folder. Figure 16 below helps illustrate the structure of the backend.

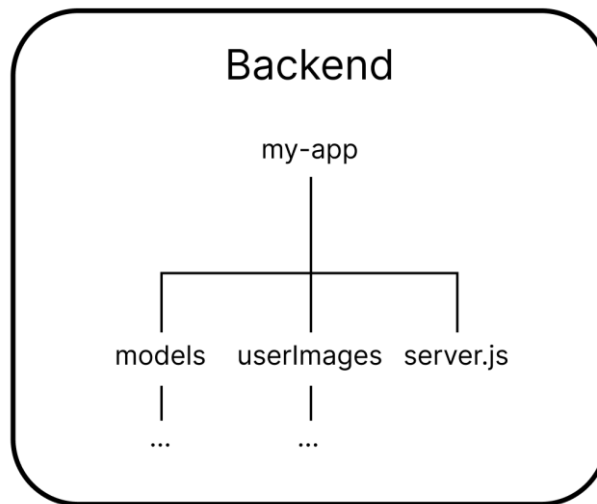


FIGURE 16 STRUCTURE OF THE BACKEND

“server.js” contains our web server as well as our POST and GET requests.

The “models” folder contains our Mongoose schemas wrapped in models.

And lastly the “userImages” folder serves as a storage for images that are uploaded by users. The images are stored here and then uploaded to MongoDB via Multer.

3.2.2.2 DATABASE

Since we decided to use the MERN stack for the development of Yggdrasil, we used MongoDB as our database (*see section 2.2.9.3*). To manage it, we utilized MongoDB's own database manager, Atlas. Our database contained the stories provided by our employer and user submissions. Each POI's content was stored in separate collections, and their key-value pairs were defined by our Mongoose schemas wrapped in models. The models were used to add new documents to our database through POST requests and retrieve them with GET requests to populate our map component with POIs.

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;

const rurStorySchema = new Schema({
  title: {
    type: String,
    required: true
  },
  text: {
    type: String,
    required: true
  },
  lat: {
    type: Number,
    required: true
  },
  long: {
    type: Number,
    required: true
  }
}, { collection: 'rurStories' });

const RurStoriesModel = mongoose.model("rurStories", rurStorySchema)
module.exports = RurStoriesModel
```

FIGURE 17 MONGOOSE SCHEMA WRAPPED IN A MODEL

We ended up with 4 different schemas, these were: additionsSchema, rurStorySchema, rurTranscriptSchema and viewAdditionSchema. An overview of the different schemas can be found in appendix [9.5](#).

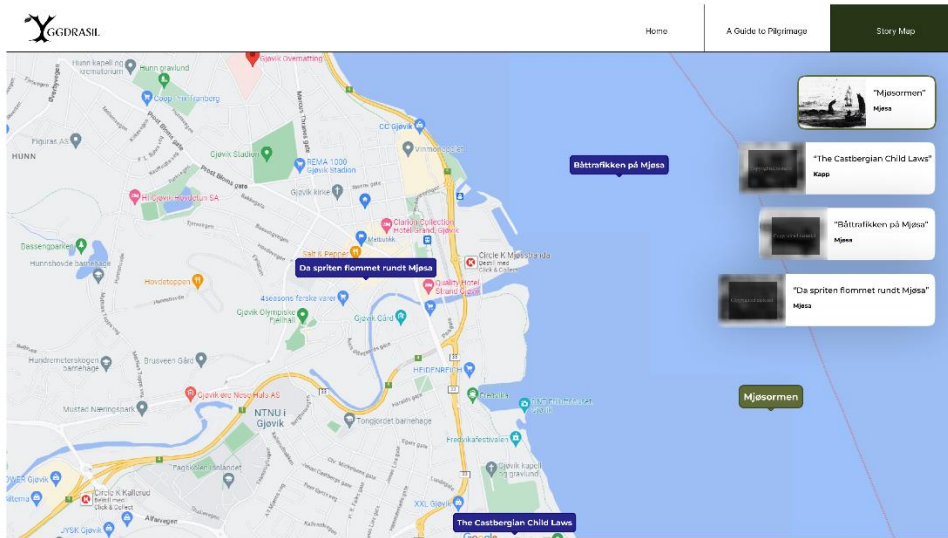
3.3 THE YGGDRASIL SOLUTION

In this section we will go over how our solution differs from the high-fidelity prototype after we had finished developing it. We will also look at how we did the final test of our solution.

3.3.1 CHANGES FROM THE HIGH-FIDELITY

When comparing the hi-fi prototype to the finished design of Yggdrasil, we found that the story map page underwent the most changes. Specifically, we made three main changes to the page. Firstly, we used the "create a style" tool in Snazzy Maps (<https://snazzymaps.com>) to style the map page to better match Yggdrasil's colour scheme, as seen in the colour palette in section [3.5.1.1](#). Secondly, we added different markers under the categories "View", "Other" and "(user) Story", which users can add themselves if desired. Finally, we added a marker to show the user's position on the map, making it easier for them to navigate.

Story map



Story map

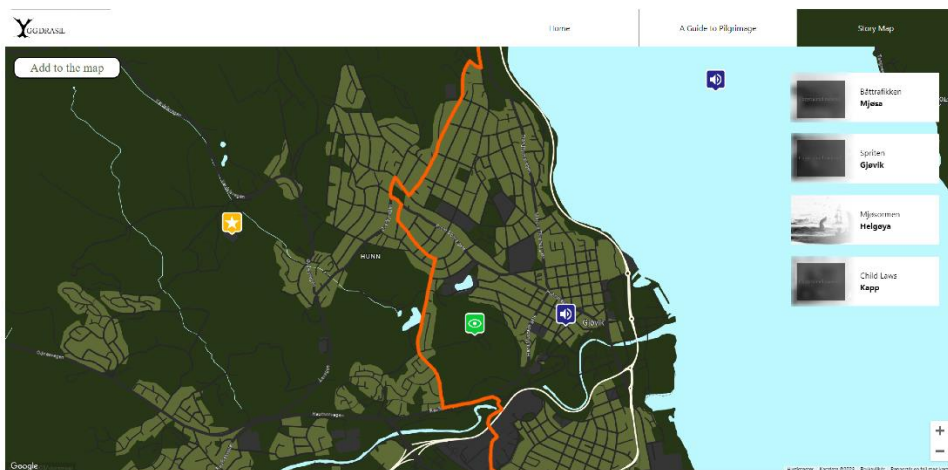


FIGURE 18 CHANGES TO THE STYLING OF THE MAP

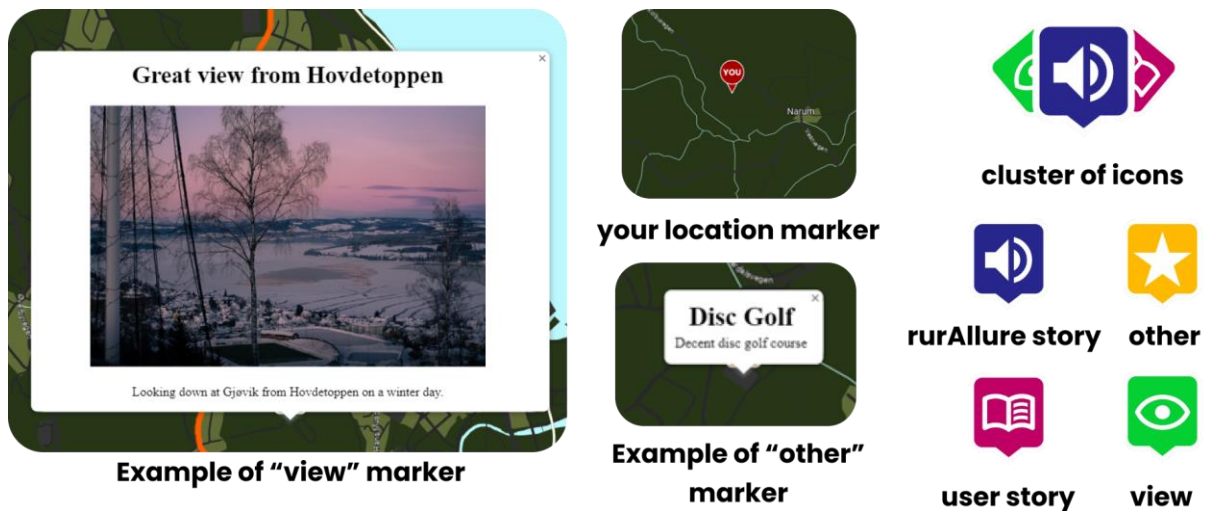


FIGURE 19 THE DIFFERENT CHANGES ON THE MAP

3.3.2 HOW WE TESTED THE YGGDRASIL APPLICATION

In this section we will explain how we tested our application.

3.3.2.1 CHOOSING A STANDARDIZED QUESTIONNAIRE

We looked at the questionnaires mentioned in [2.2.7.1](#), and to iterate they were, *QUIS*, *SUMI*, *PSSUQ*, *SUS*, *UMUX* and *UMUX-LITE*. As mentioned, we chose to utilize the *SUS* questionnaire. We will explain why we chose to do this here.

There were a few reasons as to why we did not use the others over *SUS*. *QUIS* and *SUMI* both required us to buy a license to be able to use them. We had no plans on spending any money on the project so these two were quickly discarded.

UMUX and its shorter version *UMUX-LITE* were both good candidates, but they were simply too short for us to do a proper user testing to our liking. They would work much better for a quick and simple user test, but we wanted the last user testing to be the final test to see how engaging our application was.

The last candidate *PSSUQ* was almost considered, but as we examined its question set, we noticed that a few of the questions were irrelevant to our application. Example: “The information (such as online help, on-screen messages, and other documentation) provided with this system was clear”. Our solution did not include any of these listed features and we thought it would be useless to include the question set if not all the questions were relevant

for the application. It could also negatively impact the calculations we would have to do following the testing.

SUS had all we were looking for. Its question set was a good size with ten items, and all the questions fit with the capabilities and features of our website. It was also free to use, which is good for two reasons. One, we do not need to use any of our own money and two, we did not have to wait for us to be accepted and get the license by the institute who made the questionnaire.

See appendix [9.6](#) for the questions sets of the questionnaires.

3.3.2.2 HOW WE USED SUS

We administrated a SUS questionnaire (Brooke, 1996), available on Google Forms, for 10 testers we gathered via email as well as in person. This was done after our testers had looked at the application. We wanted to measure the usability of the site now that it was finished. With the answers from the questionnaire, we would be able to calculate a SUS score for us to gauge how effective, efficient, and easy our application is to use (UIUXTrend, 2021).

3.3.2.3 REDUCING BIAS

The users answered the questionnaire in person, but the answers they gave were anonymous. We made the answers anonymous by having all the users answer the questionnaire with the same Google account on one of our computers. Since we knew all the people we tested, the anonymity and our encouragement to answer truthfully, were measures to minimize the effects of social desirability bias and selection bias. Even with these measures, the accuracy of our results may still have been affected due to bias, as mentioned in [2.2.6.2](#).

3.3.2.4 MAKING OUR OWN QUESTIONNAIRE TO MEASURE INTEREST

We also wanted to ask some questions regarding interest in the topics we focus on in our website the day we tested our application. The SUS questionnaire dictates that you only use their questions for the test. This is to ensure that you get the correct SUS score. This meant that we had to make an additional questionnaire to ask these questions. We made a small question set of 5 items where 2 questions were meant to be asked pre user test (*see appendix [9.8](#)*), and then repeated the 2 questions, with one additional, after the user test and the SUS

questionnaire were finished. This was done in order to see if our website made the testers more interested in the topics.

3.4 SUSTAINABLE MEASURES

During the initial stages of development, we aimed to incorporate some sustainability measures into the website. It is worth noting that we currently have no plans to host the website. However, the sustainable measures we implemented would benefit the project if it were to be deployed in the future. You can find details of these measures below.

See [2.2.14](#) for theory.

3.4.1 CODE SPLITTING AND LAZY LOADING

Code splitting and lazy loading are two crucial techniques used in web development to enhance website performance. The basic concept of code splitting and lazy loading involves breaking down the code into smaller segments that are only loaded when necessary. As a website grows and includes numerous pages, implementing code splitting becomes essential to maintain fast loading speeds.

In our project, we implemented code splitting and lazy loading for all our pages. This means that each page and its essential components are loaded separately. For instance, the code for the landing page is loaded when a user first visits the website. When the user navigates to another page, only then, the code for that specific page is rendered. This results in faster loading times for the website, which enhances the overall user experience.

We implemented lazy loading on our pages by using the "lazy()" function for the components of each page.



FIGURE 20 LAZY LOADING STEP BY STEP

By placing the component in a “<Suspense>” tag, they will become lazy loadable. Just as you can see on figure 20. Additionally, <Suspense> makes it possible to add a fallback function. We added the fallback function with the text “Loading...”. As a result, when the user is waiting for the components to load, they will be greeted with the given fallback text.

3.4.2 FILE TYPES AND FILE SIZE

As the website expands, it will accommodate more files such as images and audio files. These file types can occupy a significant amount of space if stored in an unfitting file format. Therefore, it is crucial to store them in appropriate file formats. The different file types we use in the project are images and audio files.

The way we decided to store these files was by using mainly WebP files for the images, and mp3 files for the audio files. The reason being that both file formats are low in file size, but still has enough quality and detail in the image and sound (*see section [2.2.14.3](#)*).

3.4.3 REMOVE REDUNDANT CODE

We ensured that we eliminated any noticeable redundant code from our project as it occupies space and causes the website to slow down. This has a positive impact on the website's sustainability and performance (*see section [2.2.14.4](#)*).

By implementing all the measures mentioned in the previous sections, the website's loading speed increased, making it a more enjoyable experience for users.

3.4.4 TESTING THE SUSTAINABILITY OF THE WEBSITE

We chose to utilize Google Lighthouse (*see section [2.2.13.4](#)*) as our tool for testing the sustainability of the website. We looked at the score it gave and what it told us to fix and then we optimized the code to achieve the highest possible score.

See [4.3.6](#) for all the Lighthouse results.

3.5 ACCESSIBILITY MEASURES

Measures to increase accessibility is an important part of designing web solutions that are usable for all types of users (*see [2.2.13](#)*). This chapter will cover the measures we implemented to make our project accessible, as well as the technologies we used to test the overall accessibility of our final product.

3.5.1 DESIGN CHOICES

3.5.1.1 COLOURS

When selecting colours for Yggdrasil, we decided to create a colour palette. We kept in mind the need for good contrast to ensure accessibility for people with colour blindness. We also wanted the colours to be consistent with the website's theme (*see [3.1.5.3](#)*), nature.

To find a suitable palette, we used the website colors.co (<https://colors.co>), which offers tools for generating and exploring popular colour schemes. After some searching, we identified a palette that met our criteria, featuring shades of green, an off-white colour, and two shades of orange/brown. Since Yggdrasil is a tree, the colours would be symbolic of that. See figure 21 for a visual representation of the palette.



FIGURE 21 PICTURE OF THE CHOSEN COLOUR PALETTE WITH HEX CODES

3.5.1.2 USING TAB INDEX

In our code, we apply tab index to all the elements that we want the user to navigate through in a certain order when using a keyboard as a navigation device. Proper use of tab index can benefit screen reader users by ensuring that they receive information about the most important content on the website (see [2.2.13.5](#)).

```

rurAllure_bachelor - Footer.jsx
1  <img tabIndex={0}
2      src={logo}
3      alt="The logo of NTNU, Norwegian University of Science and Technology"
4      className="w-1/3 md:w-1/6"
5  />
```

FIGURE 22 EXAMPLE OF TAB INDEX

3.5.1.3 TRANSCRIPT

In addition to our user stories and text, we added a transcribe function that highlights the text being read when audio is playing.

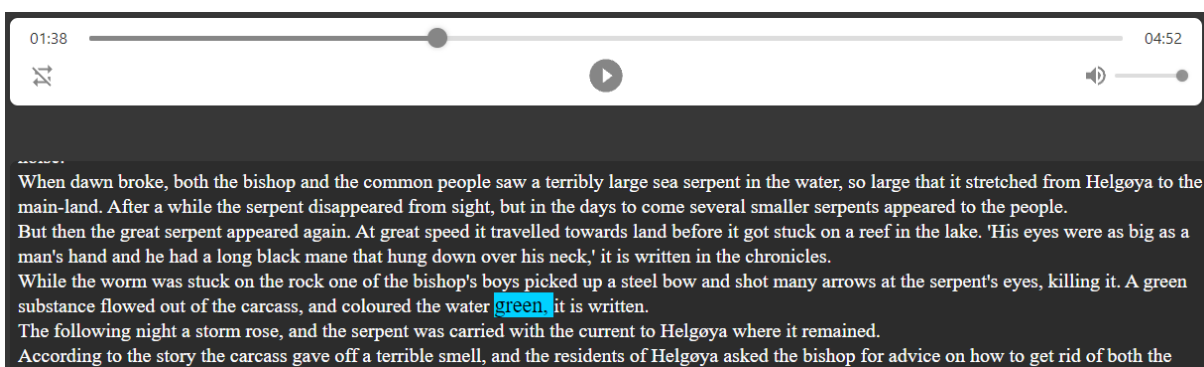


FIGURE 23 PICTURE OF OUR TRANSCRIBE FUNCTIONALITY

3.5.1.4 ELEMENT SIZING

During the development of Yggdrasil, we took measures to ensure that all the elements on the website were appropriately sized. This was done to ensure that individuals with visual impairments and other visual disabilities could view the elements clearly.

3.5.1.5 ELEMENT HOVER

According to W3C guidelines, interactive elements should be easily distinguishable to enhance accessibility. To meet this requirement, we incorporated hover effects on buttons, causing them to expand in size and change background color when hovered.

3.5.1.6 IMAGE ALT TEXT

In our code, we attempted to ensure that each image has appropriate alt text to aid screen reader users in understanding the content of the image. We did our best to create alt text that would be meaningful and helpful.

3.5.2 TESTING THE ACCESSIBILITY OF THE WEBSITE

We chose to utilize Wave (*see* [2.2.13.3](#)) and Google Lighthouse (*see* [2.2.13.4](#)) as our tools to measure accessibility.

3.5.2.1 USING WAVE

We utilized the Wave tool to evaluate our website's compliance with WCAG standards (*see* [2.2.13.2](#)). We utilized the results to enhance the website's accessibility features.

See results in [4.3.7](#).

3.5.2.1 USING GOOGLE LIGHTHOUSE FOR ACCESSIBILITY

Since Google Lighthouse evaluate multiple factors, including accessibility, we utilized it to measure the accessibility of our website using the provided score. As with the Wave tool, we used the results to improve the website's accessibility features.

See results in [4.3.6](#).

3.6 ORGANIZING THE PROJECT

When working on a bachelor project, it is important to have good ways to communicate and that the team works together well. We are a group of three students that have all studied web development at NTNU for almost three years and have worked together on several different

projects. This gives us a good understanding of each other's workflow and thought processes. Before commencing the project, we formulated a contract to ensure that each of us would contribute to the work. We value consistency in workflow and all members are aware of this.

To communicate with our tutor, we used email and Microsoft Teams. Our meetings with him were conducted either in-person at the university or digitally. We tried to keep these meetings as regular as possible to ensure that we did not progress too far without his guidance.

The report was written in Google Docs in the beginning because it allowed one of us to edit the text and the others to see the changes in real time. We however soon realized that Microsoft Word allowed for more customization of the document text. This led us to switch to it, and it was what we utilized for the rest of the report.

3.6.1 COMMUNICATION AND WORK MEDIUMS

Below are the different communication and work mediums we agreed upon to utilize during our working sessions.

3.6.1.1 DISCORD

When working remotely together, we needed a platform to communicate. Discord was our preferred option, and the platform we decided to communicate on. This platform allows us to talk together via voice chat, which means that when we cannot meet in person, we can still communicate effectively. Discord also allows their users to share images and files which made it easy for us to share material within the group.

3.6.1.2 MICROSOFT TEAMS

When physical meetings were not possible during our discussions with the project owner and supervisor, we relied on an online communication tool. As such, we opted to use Microsoft Teams for this purpose.

3.6.1.3 COLLABORATION USING GITHUB

“GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.” (GitHub, u.d.).

To collaborate remotely on the project's code, we utilized GitHub by creating a shared repository that all team members worked in. By utilizing GitHub, we also ensured that the

code would not be lost in case someone accidentally deleted parts of it since it is always possible to recover previous versions by returning to a previous snapshot.

3.6.1.4 GOOGLE DOCS

We utilized Google Docs to write and organize our notes, structure, and overall planning for our bachelor thesis. This was because we could all work on a single document remotely at any time.

3.6.1.5 CONDUCTING A QUESTIONNAIRE WITH GOOGLE FORMS

To gather user data, we used Google Forms to administer a questionnaire. Once we had collected the responses, Google Forms provided visualizations of the data, which we then used to our advantage while iterating.

3.6.1.6 FIGMA

We used Figma as the design tool for our website, where we ideated the design and interactions of the prototypes and wireframes. This is also where we had our user-testing of the prototypes. Figma lets the users click through and explore “flows” which determines how the prototype will work. It was a natural choice to use Figma because we had a lot of experience with it and because of how easy it is to create life like prototypes with it. And since it can be tested in the app it eliminated the use of extra time to export it to another platform.

4 RESULTS

This chapter will include the most important findings from our design and developing methods, as well as the accessibility and sustainability of our solution. These results will be helpful for us later in [chapter 5](#), when we will try to answer the problem statement.

4.1 DESIGN RESULTS

In this section the results from the design process will be displayed.

4.1.1 LOOKING INTO OUR COMPETITORS

4.1.1.1 SWOT RESULTS

As mentioned in [3.1.3.2 SWOT](#) we wanted to compare the strengths, weaknesses, opportunities, and threats of our competitors.

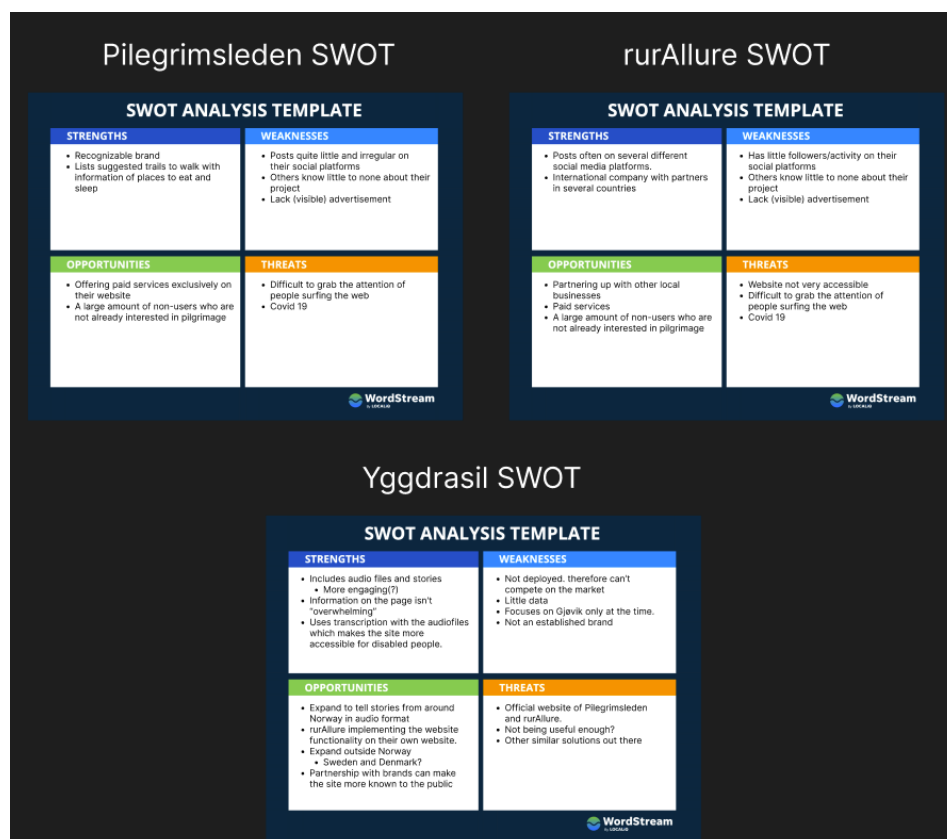


FIGURE 24 SWOT ANALYSIS OF OUR COMPETITORS AS WELL AS OURSELVES

We found that our competitors did not have the biggest online presence, lacked visible advertisement and were not best accessibility-wise.

4.1.1.2 USABILITY COMPETITIVE ANALYSIS RESULTS

We conducted a usability competitive analysis of rurAllure and Pilegrimsleden.no compared to our Yggdrasil solution. Our analysis was based on table 2.1 from “Understanding Your Users: A Practical Guide to User Research Methods (2nd ed.)” by Baxter, Courage & Caine (2015)

	Yggdrasil	rurAllure	Pilegrimsleden.no
Unique features	<ul style="list-style-type: none"> • Stories with both audio and text in POIs. 	<ul style="list-style-type: none"> • Journey planner with the ability to influence the route by selecting interests. • Apps for mobile devices. 	<ul style="list-style-type: none"> • Ability to filter POIs on the map based on interests. • Route planner with ETA and the length in km. • Ability to manually add POIs as stops in your route. • Dotted lines showing all the St. Olav Ways
Design Strengths	<ul style="list-style-type: none"> • Users are able to add POIs to the map. • Story modals contain a decent amount of content. 	<ul style="list-style-type: none"> • International focus covering several different pilgrim paths. • Has a "Pilgrims' portal" where the different maps are all together. • Has both english and norwegian version. • Account creation for saving planned routes. 	<ul style="list-style-type: none"> • Routes are customizable. • Simplistic design, easy to start using. • Has both english and norwegian version. • Account creation for saving planned routes.
Design weaknesses	<ul style="list-style-type: none"> • Lack of personalization options. • English is the only language for the page itself. • Inconsistency in language used for POIs. 	<ul style="list-style-type: none"> • Some of the maps are lacking in content. • The projects are in their early stages. 	<ul style="list-style-type: none"> • Large amount of information and POIs might be overwhelming for some users
Requirements	<ul style="list-style-type: none"> • Not available to the public 	<ul style="list-style-type: none"> • Web browser and internet connection or app and internet connection 	<ul style="list-style-type: none"> • Web browser and internet connection
Core features			
Route planner	X	✓	✓
Route Suggestions	X	✓	✓
Audio narratives	✓	X	X
Map with POIs	✓	✓	✓
Filters for POIs	X	✓	✓
Info about the St. Olav Ways	✓	✓	✓

FIGURE 25 USABILITY COMPETITIVE ANALYSIS RESULTS

4.1.2 LOW FIDELITY PROTOTYPING (LO-FI)

This chapter will cover the different results of our low-fidelity prototype, including the results of the user testing.

See appendix [9.11](#) for the entire low-fidelity prototype.

4.1.2.1 RESULTS FROM LO-FI USER TESTING

The user testing we did for the lo-fi prototype was mostly centred around gauging the users design preferences and general interest in the topics our project covered. After the tests we ended up with several suggestions for changes. Users wanted more space, no autoplay on audio files, a slideshow instead of scrollable text on mobile, a more noticeable MP3 player, and timestamps. They also wanted us to make it more obvious that the points of interest on the map were clickable, and another valuable suggestion was that we should ensure that the screen reader and text didn't interfere with each other.

When we asked users if they found the idea of audio files interesting, we received mostly positive responses. Users also showed interest in visiting the locations where the stories took place, although those who did not already have an interest in nature and local history were not very enthusiastic.

Additional recommendations we received were to seek inspiration from “Strava” or “Stolpejakt”, integrate a button in the modal to highlight text, and investigate gamification elements like a scoreboard or leaderboard to increase user engagement.

4.1.3 HIGH FIDELITY PROTOTYPING (HI-FI)

This chapter will cover the different results from our high-fidelity prototype, including user testing results, and the prototype iterations.

See appendix [9.12](#) for the entire high-fidelity prototype.

4.1.3.1 RESULTS FROM HI-FI USER TESTING

Observation-based user testing (*see section [2.2.6.1](#) for definition*) was conducted to evaluate the website's design and gain insight into participants' opinions on the mobile and desktop designs. Participants were assigned tasks, including finding information about the "Mjøsormen" story, returning to the homepage, and discovering what the "Olavs brev" is.

Feedback revealed that while the website's layout was good and easy to read, participants were uncertain about the contents of different pages based on the naming used in the navigation bar. Additional questions were asked to gain further insight into design, including opinions on colours and the necessity of a landing page. Feedback received on the questions

varied, with participants suggesting changes to the layout, colour usage, and content of the landing page. Other useful feedback included adding images when hovering over pinpoints on the map and increasing the hitbox for buttons.

4.2 THE YGGDRASIL PROJECT

4.2.1 USER TESTING THE FINAL PRODUCT

This section will highlight the results we got from the user tests of our web application.

4.2.1.1 SUS-QUESTIONNAIRE

This section will show the answers we got after we conducted a questionnaire following the SUS standard. As mentioned in [3.3.2.2](#), the users would answer it in Google Forms. The data was linked to a excel sheet where we could turn the data into informative graphs.

After inputting all the data into an excel document, we calculated the raw SUS score by using the approach described in [2.2.7.2](#). Figure 26 shows how we did it.

$$= (B2-1) + (5-C2) + (D2-1) + (5-E2) + (F2-1) + (5-G2) + (H2-1) + (5-I2) + (J2-1) + (5-K2)$$

FIGURE 26 CALCULATING THE SUS SCORE IN EXCEL

To calculate the final SUS score for each participant, we took the raw SUS score and multiplied it with 2,5. The final SUS score we got from our testers can be seen in figure 27 below.

	Tidsmerke	would like to use this system frequently.	I found the system unnecessarily complex.	I thought the system was easy to use.	would need the support of a technical	various functions in the system were well	there was too much inconsistency in this	imagine that most people would learn to use this	I found the system very awkward to use.	I felt very confident using the system	learn a lot of things before I could get going with	RAW SUS SCORE	FINAL SUS SCORE
1													
2	03.05.2023 kl. 10.35.46	2	1	4	1	4	2	5	1	4	2	32	80
3	03.05.2023 kl. 11.11.21	3	2	4	1	3	2	4	2	3	1	29	72,5
4	03.05.2023 kl. 11.24.57	1	1	4	1	4	2	4	1	4	1	31	77,5
5	03.05.2023 kl. 11.31.28	2	2	4	2	4	3	4	2	4	3	26	65
6	03.05.2023 kl. 12.04.15	2	1	4	2	4	1	5	1	4	1	33	82,5
7	03.05.2023 kl. 12.48.54	2	2	3	1	4	3	3	2	4	1	27	67,5
8	03.05.2023 kl. 13.42.46	2	1	4	1	2	3	4	2	5	1	29	72,5
9	03.05.2023 kl. 17.37.38	4	1	5	1	5	1	5	1	5	1	39	97,5
10	03.05.2023 kl. 17.54.29	4	1	4	1	4	1	5	1	5	1	37	92,5
11	03.05.2023 kl. 19.15.31	2	1	4	1	5	2	5	1	4	1	34	85
12													
13													
14												Average sus score:	79,25

FIGURE 27 EXCEL TABLE OF SUS SCORE

As seen on figure 27, the average sus score is 79,25. Compared to table 8.5 in the book “Quantifying the User Experience : Practical Statistics for User Research” by Sauro & Lewis (2016, p. 204) , 79,25 equals to an A- grade.

SUS Score Range	Grade	Percentile Range
84.1–100	A+	96–100
80.8–84.0	A	90–95
78.9–80.7	A–	85–89
77.2–78.8	B+	80–84
74.1–77.1	B	70–79
72.6–74.0	B–	65–69
71.1–72.5	C+	60–64
65.0–71.0	C	41–59
62.7–64.9	C–	35–40
51.7–62.6	D	15–34
0.0–51.6	F	0–14

FIGURE 28 GRADING SCALE FOR SUS (SAURO & LEWIS, 2016, P. 204)

See appendix [9.7](#) for the full dataset.

4.2.1.2 GAUGING INTEREST

The users were additionally given two interest related questions that they could rate from 1 (not interested) to 5 (very interested), before browsing the website and the same two after, with the addition of a yes/no question after the test.

Figure 29 shows how the users felt their interest was before they tried out Yggdrasil compared to what their level of interest was after:

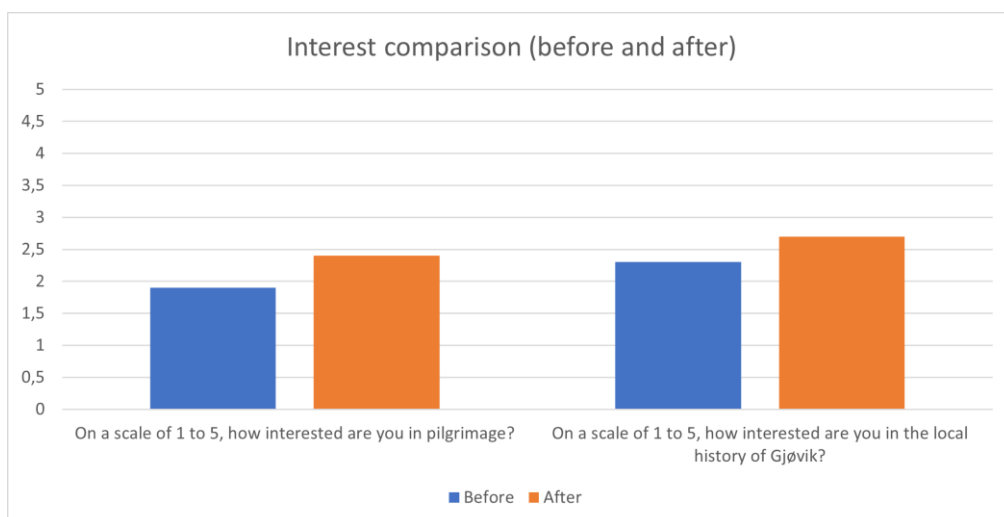


FIGURE 29 INTEREST SCORE FROM USER TESTING

After analysing and calculating the responses from our before and after interest questionnaire, we found that there was a 26.32% increase in the level of interest in pilgrimage and a 17.39% increase in the level of interest in the local history of Gjøvik.

We also analysed the results of the “Did you learn anything new while visiting the site?” question, which gave us the following results:

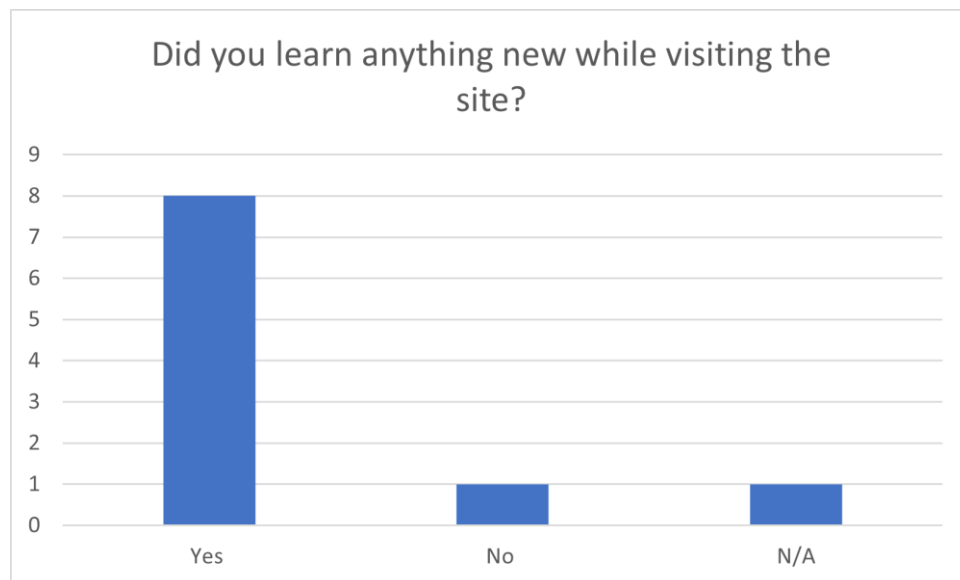


FIGURE 30 THE RESULTS OF THE QUESTION "DID YOU LEARN ANYTHING NEW WHILE VISITING THE SITE?"

Eight out of ten users responded that they learned something new, which is equivalent to 88.89% of the users. One user answered that they did not learn something new, and one user did not respond to the question and was therefore not included in the overall percentage.

See appendix [9.9](#) for the full dataset.

4.2.1.3 OBSERVING HOW THE USERS NAVIGATE THE FINAL WEBSITE

While the participants were browsing the website, we did an observation of the user habits and how they used our website. In figure 31, you can see what we were observing and what the result average was.

	Links clicked	POI's clicked	Page spent most time on	Total time spent on yggdrasil (minutes)
Result average	1,2	5,2	Map	05:52

FIGURE 31 AVERAGE RESULT OF OBSERVATION

It is worth noting that the map and information page were both popular choices for the "page spent most time on," but the map page had one more selection than the information page.

See appendix [9.10](#) for the full dataset.

4.3 SUSTAINABILITY & ACCESSIBILITY RESULTS

For our project, several measures were made to increase and improve the sustainability and accessibility of the website. In this chapter, you will read about the results of these measures.

4.3.1 APPLYING CODE SPLITTING AND LAZY LOAD

When conducting the test to measure the impact of both code splitting and lazy loading, we used the "fast 3g" throttling setting in the Chrome "network" performance test, see figure 32. We made this choice because not everyone has access to high-speed internet such as 4G or 5G. To ensure a more realistic test of the application, we opted for the "fast 3g" option.

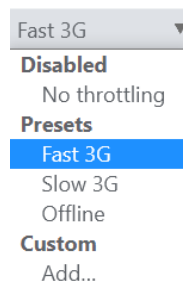


FIGURE 32 THROTTLING SETTINGS UNDER THE TEST

We are measuring two factors: "finish" time and "load" time. The "finish" time includes all the elements and code on the page, including asynchronous objects and elements. As (Prokopets, u.d.) explains, the "finish" value tells us how long it takes until all elements have finished loading, even if the page appears to be loaded to the user. On the other hand, the "load" time measures how long it takes for the page to be fully loaded for the user, excluding the asynchronous elements that might end up loading in the background. While the "load" time is more accurate for our use case, we have included both measurements to identify the differences.

To measure the difference in percentage we used the formula:

$$\text{Improvement in \%} = ((\text{original time} - \text{new time}) / \text{original time}) * 100\%$$

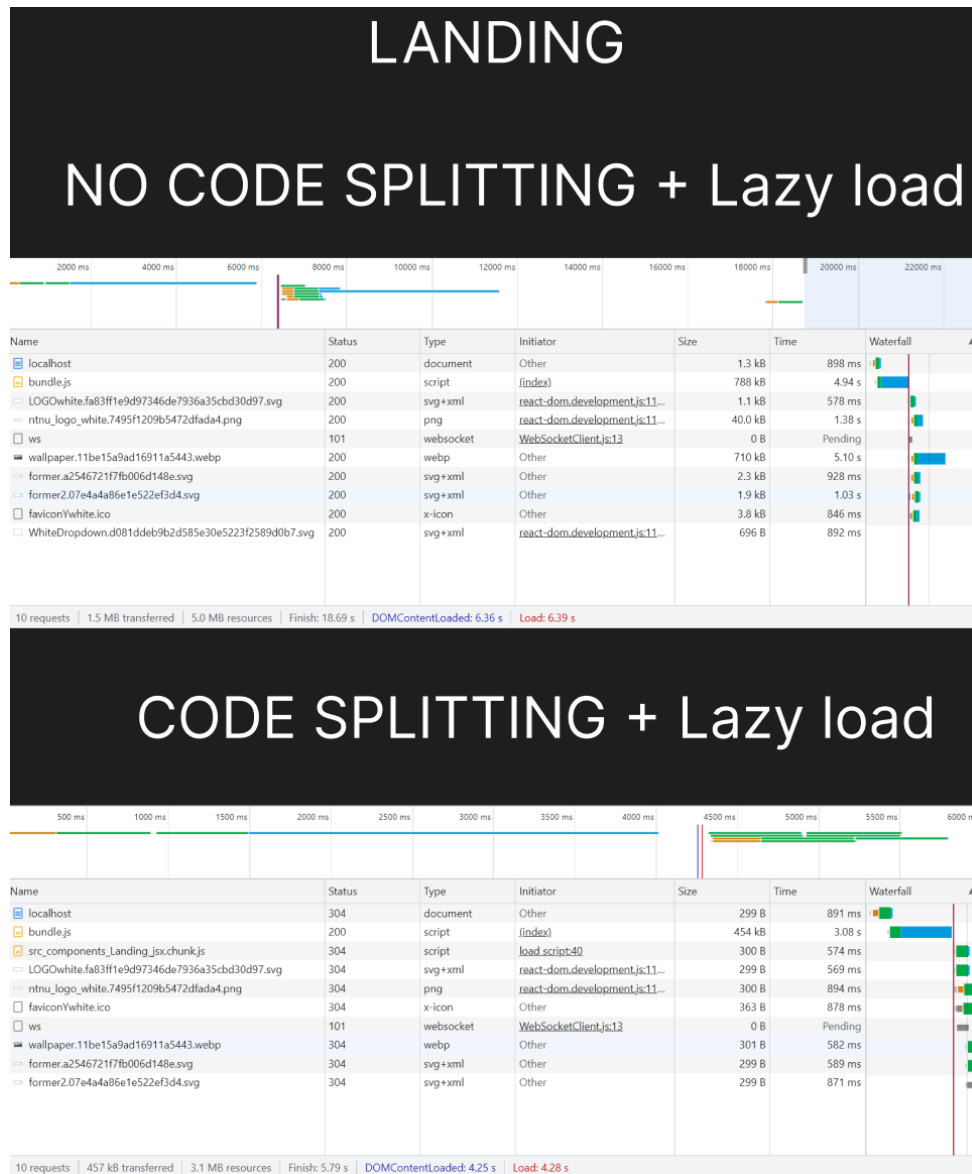


FIGURE 33 NETWORK TOOL OVERVIEW ON “LANDING” PAGE

The time difference in page load with code splitting and lazy loading is 12.81 seconds (18.6s - 5.79s). This means that lazy loading results in a 68.8% improvement in loading time ((12.81 / 18.6) * 100). Similarly, the time difference in load time is 2.11 seconds (6.39s - 4.28s), which equates to a 33% faster loading time ((2.11 / 6.39) * 100%).

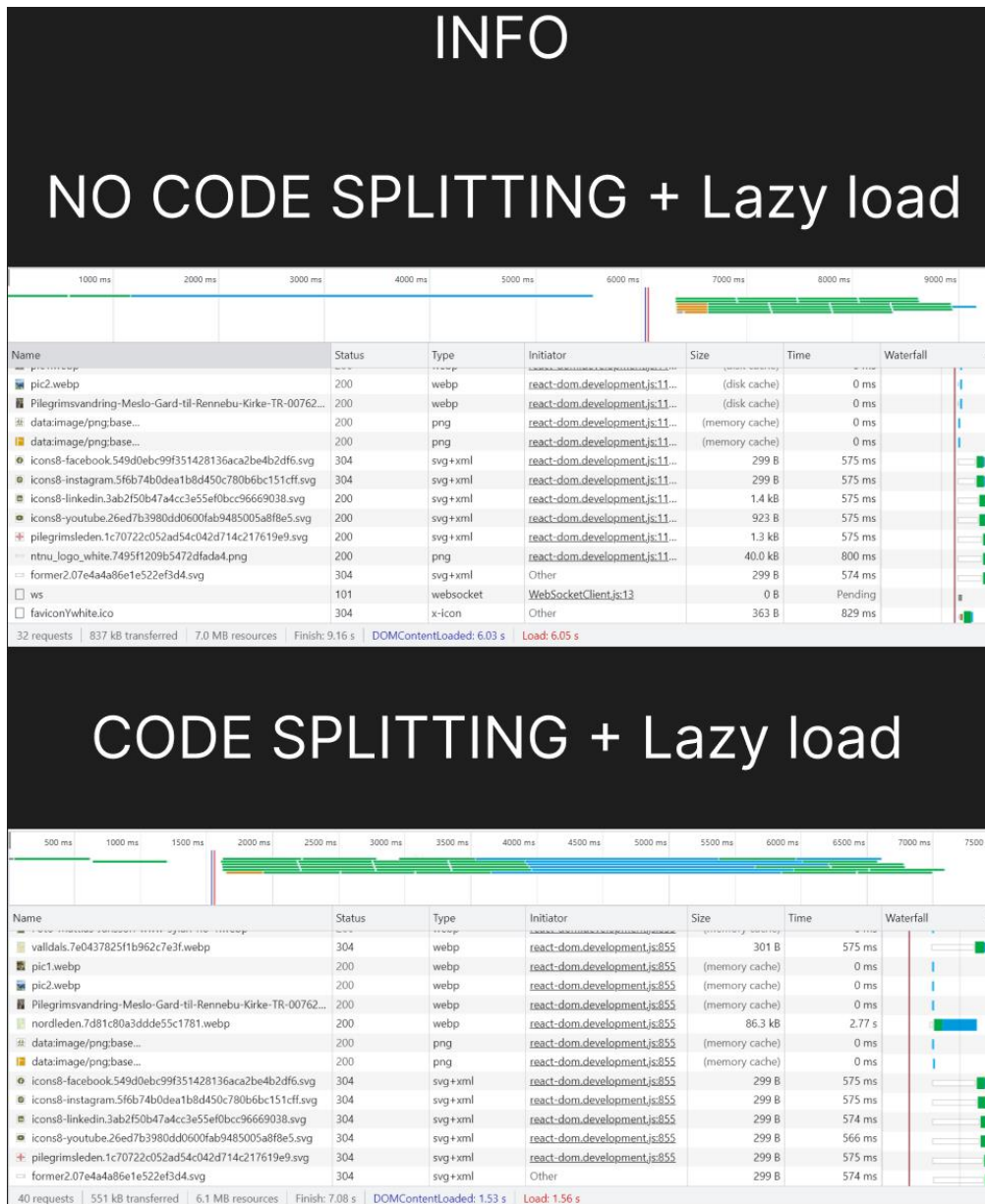


FIGURE 34 NETWORK TOOL OVERVIEW ON “INFO” PAGE

The time difference in finished load time with code splitting and lazy loading is 2.08 seconds (9.16s - 7.08s), resulting in a 22.7% faster loading time $((2.08 / 9.16) * 100)$. The time difference in load time is 4.49 seconds (6.05s - 1.56s), which equates to a 74% increase in speed $((4.49 / 6.05) * 100\%)$.

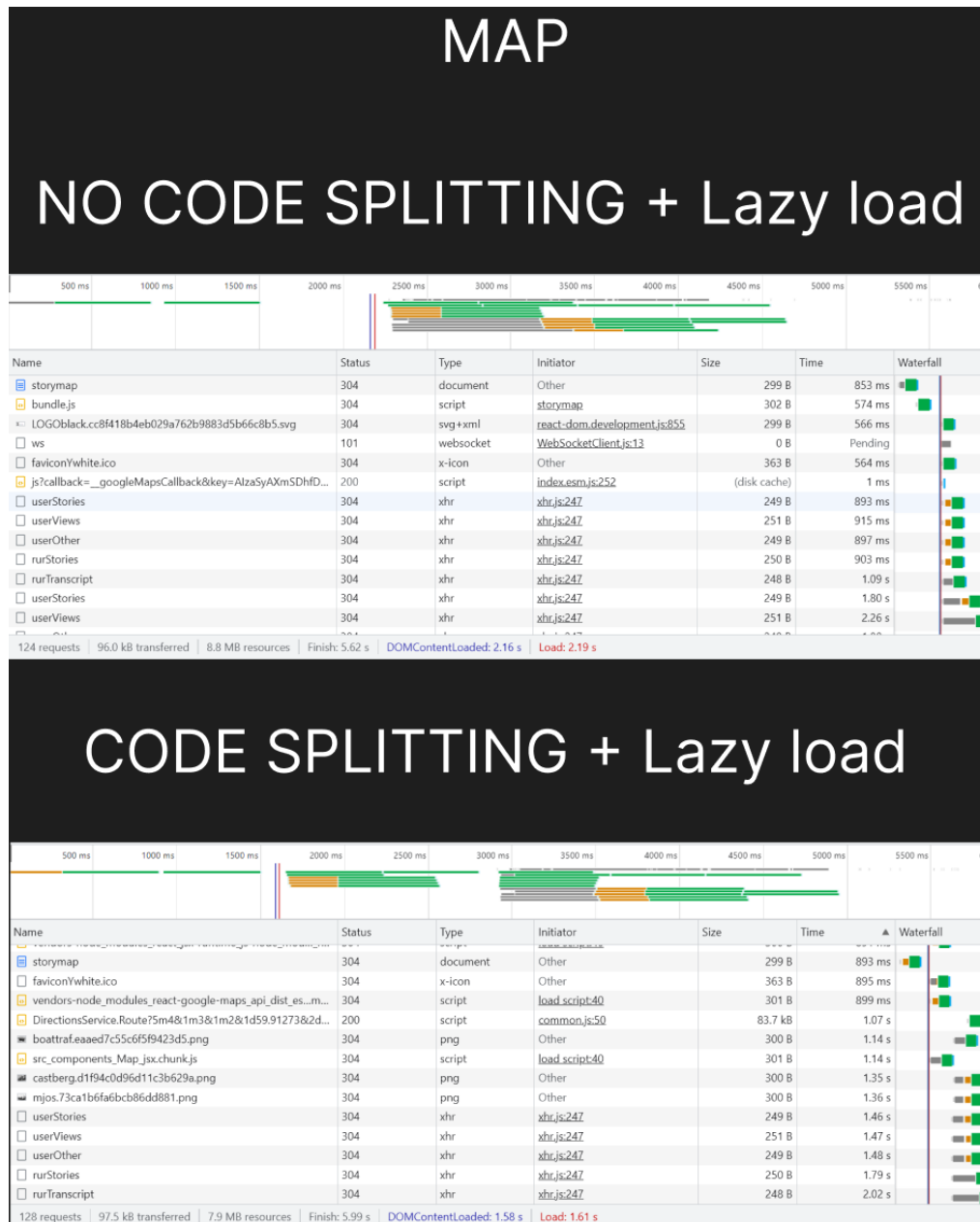


FIGURE 35 NETWORK TOOL OVERVIEW ON "MAP" PAGE

The time difference in finished load time with code splitting and lazy loading is -0.37 seconds (5.62s - 5.99s), indicating a 6.58% slower loading time with lazy loading $((-0.37 / 5.62) * 100)$. On the other hand, the time difference in load time is 0.58 seconds (2.19s - 1.61s), which equates to a 26% increase in speed $((0.58 / 2.19) * 100\%)$.

Based on the results, we can summarize that the use of code splitting and lazy loading is generally faster. On average, the speed increase we achieved by using these methods was

28.31% in "finished" and 44.34% in "load". This shows that code splitting and lazy loading are effective methods for improving website loading times and overall sustainability.

4.3.2 REMOVING REDUNDANT CODE

After completing the project's code, we identified that there was a significant amount of redundant code, which does not serve any purpose, other than slowing down the loading speed of the website. We conducted a thorough review of the code and removed all the redundant code we could find, including unused functions, unused images, unused SVG files, and unused components, to mention a few.

As a reference, our starting project folder size was 11,9MB. This size will be reduced as we progress further in this chapter.

11,9 MB (12 500 072 byte)

FIGURE 36 PROJECT FOLDER SIZE (EXCLUDING NODE_MODULES AND USERIMAGES) BEFORE REMOVING REDUNDANT CODE

By utilizing the integrated "coverage" tool in Google Chrome, we can generate a visual representation of which files in our project has the most unused bytes. Figure 37 displays a list of files with the most unused bytes, represented in the "usage visualization", where unused code is marked in red and used code in blue. After analysing, we can determine that "bundle.js" and "react-google-maps_api" are the two largest files with the most unused code, however there are still more files with a lot of unused code.

URL	Type	Total Byt...	Unused Bytes ▼	Usage Visualization
http://localhost:3000/static/js/bundle.js	JS (per f...	2 333 136	661 924 28.4%	
/vendors-node_modules_react-google-maps_api	JS (per f...	717 065	447 621 62.4%	
https://maps.googleapis.com/maps-a.../webgl.js	JS (per f...	727 343	257 024 35.3%	
/vendors-node_modules_mui_icons-material_Arr	JS (per f...	570 341	119 457 20.9%	
https://maps.googleapis.com/maps-api-.../util.js	JS (per f...	168 194	93 226 55.4%	
/js?callback=__googleMapsCallback&key=AlzaSy	JS (per f...	192 901	68 975 35.8%	
http:.../src_components_InfoContent_jsx.chunk.js	JS (per f...	79 214	66 138 83.5%	
https://maps.googleapis.com/map.../common.js	JS (per f...	279 004	53 724 19.3%	
https://maps.googleapis.com/maps-.../marker.js	JS (per f...	74 401	49 093 66%	
https://maps.googleapis.com/maps.../controls.js	JS (per f...	93 009	44 409 47.7%	
http://localh.../src_components_Map_jsx.chunk.js	JS (per f...	214 580	37 093 17.3%	
https://maps.googleapis.com/maps-api-.../poly.js	JS (per f...	55 309	35 070 63.4%	
https://maps.googleapis.com/maps-api-.../map.js	JS (per f...	75 376	30 344 40.3%	
https://maps.googleapis.com/ma.../directions.js	JS (per f...	85 845	28 689 33.4%	
/css?family=Roboto:300,400,500,700 Google+San	CSS	26 145	26 145 100%	
https://maps.googleapis.com/maps-a.../onion.js	JS (per f...	26 986	20 297 75.2%	
/vendors-node_modules_react_jsx-runtime_js-no	JS (per f...	49 546	17 989 36.3%	

FIGURE 37 VISUAL REPRESENTATION OF UNUSED BYTES IN OUR PROJECT CODE

4.3.3 WEBP AS IMAGE FORMAT

Choosing a fitting file format can have a huge impact on the overall performance of your website (see [2.2.14.3](#)). On our website, we decided to utilize WebP as much as possible to keep the image file size as low as possible.

For a smaller image, like the images on the list on the “user stories” page, we can see a decrease in file size when converting it from PNG to WebP. Going from 50,3 kb to 9,24 by using WebP as image type.

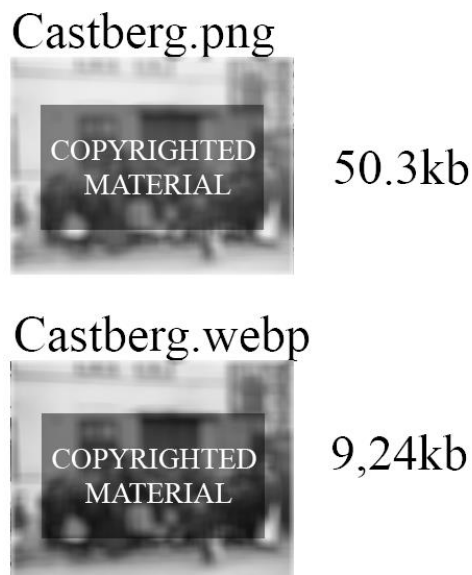


FIGURE 38 EXAMPLE OF FILE COMPRESSION BY USING WEBP

4.3.3.1 REDUCING FILE SIZE FURTHER BY COMPRESSING WEBP FILES.

Despite the smaller size of WebP images compared to other types like JPEG and PNG, they can still have a significant file size. To maintain the website's speed, we chose to further compress the largest WebP files by using the compression tool available on “Compress-Or-Die” (<https://compress-or-die.com>).

As an example, the first image displayed on the "Information about Pilgrimage" page, the original file size was 1996 kb. After compressing it on the website “Compress-Or-Die” (<https://compress-or-die.com>), the new size was reduced to 1036kb, resulting in a 48.11% decrease in size, as displayed in figure 39.

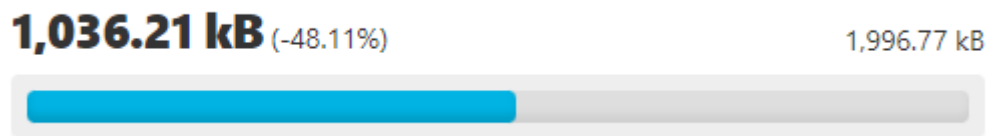


FIGURE 39 THE SIZE REDUCTION OF WEBP IMAGE ON INFO IMAGE

The folder size of our project has now been reduced to 11,6MB, after converting images to WebP format, and compressing the largest ones.

11,6 MB (12 211 180 byte)

FIGURE 40 THE PROJECT FOLDER SIZE AFTER CONVERTING AND COMPRESSING IMAGES.

4.3.4 MP3 AS AUDIO FORMAT

The original file format we recorded the audio files in, were WAV. We decided to convert these files to mp3, as this has a smaller file size and the drop in quality is inaudible to most users (see [2.2.14.3](#)).

As an example, “Mjosormen.wav” has a file size of 24,5MB, and by converting this to MP3, we are reducing the file size to 3,37MB. That equals to an 86.2% reduction in size, by just changing a file format.

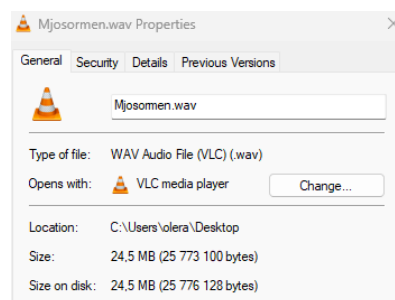


FIGURE 41 "MJOSORMEN.WAV" FILE DETAILS

4.3.4.1 REDUCING MP3 FILES WITH COMPRESSION

Like WebP files, MP3 files can be compressed to reduce their size. We used the compression tool “XConvert” (<https://www.xconvert.com/compress-mp3>) to compress our files, lowering the bitrate from 97kbps to 64kbps which results in a file size reduction. As an example, we compressed our audio file "Mjosormen.mp3", which originally had a file size of 3.37MB.

After compressing it with XConvert, the file size reduced to 2.22MB, resulting in a 34.1% decrease in size. Similar or better results were accomplished for the other audio files as well.

Before compressing the audio files, the size of the “audio” folder was 8.30MB. After compression, the folder size reduced to 5.51MB, resulting in a 33.5% decrease in size.

4.3.5 THE FINAL FOLDER SIZE

As shown in chapter [4.3.2](#), the original project folder size was 11,9MB. After doing the different measures explained above, the new achieved folder size is, as seen in figure 42, 8,92MB. This results in a total folder size reduction of 2,98MB or 25,2%.



FIGURE 42 REDUCTION OF PROJECT FOLDER SIZE

4.3.6 LIGHTHOUSE TESTING THE ACCESSIBILITY AND SUSTAINABILITY OF THE WEBSITE

After the measures were made, we decided to measure our project’s sustainability and accessibility with a Lighthouse test. The results are displayed in figure 43.

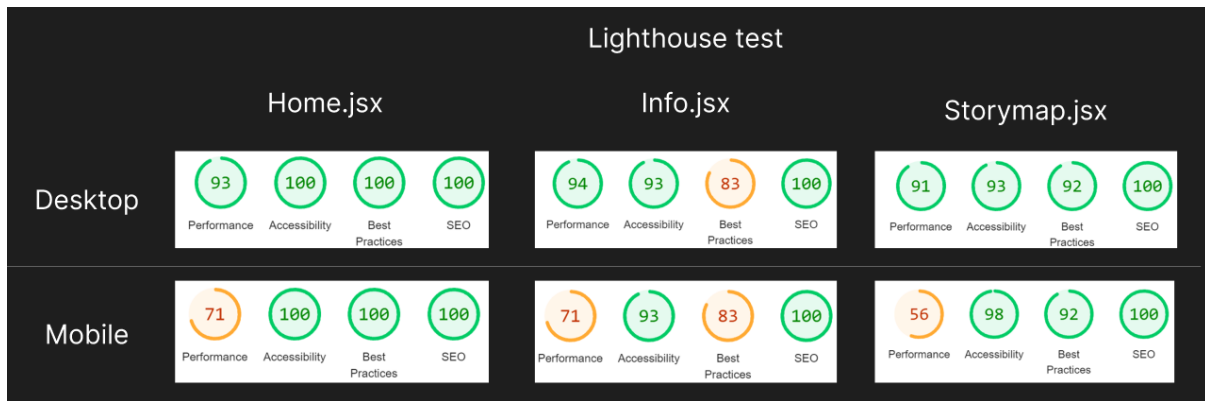


FIGURE 43 LIGHTHOUSE TEST RESULTS

As seen in the figure, the overall scores for quality, accessibility and performance in the desktop version are quite high, however on mobile it lacks some points on performance, especially in “Storymap.jsx”.

4.3.7 WAVE TEST

This is what Wave scored our website in terms of accessibility:

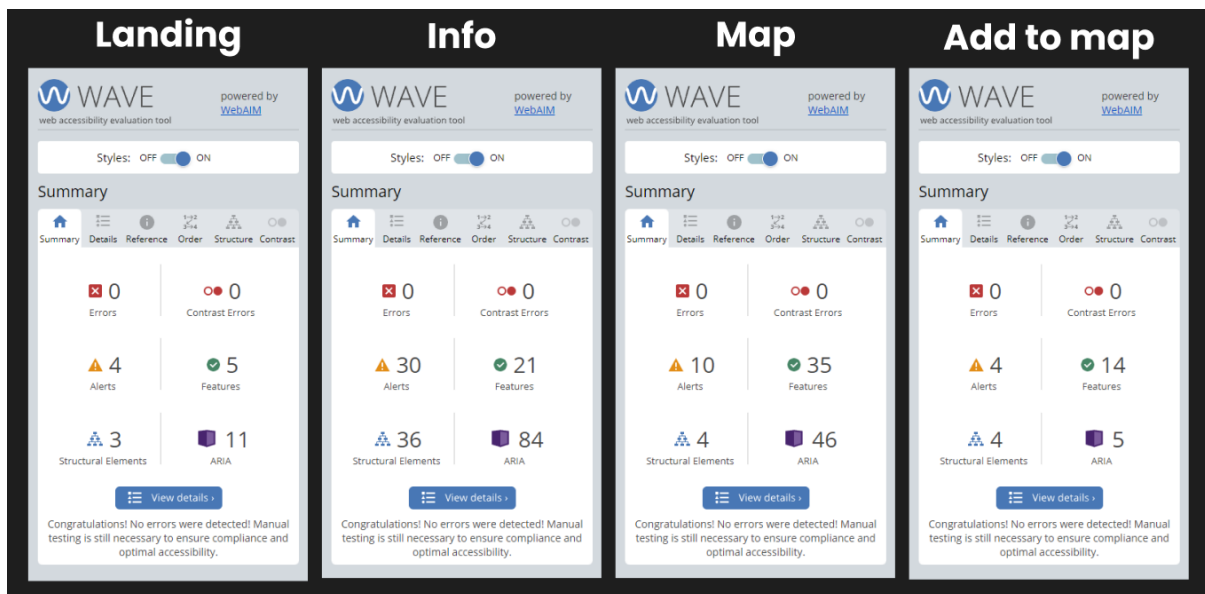


FIGURE 44 SUMMARY OF WAVE TEST

As you can see in figure 44, our application had no errors and no problems regarding contrast.

Now that we have looked at all the results we will move on to the “Discussion” chapter where we will discuss these findings and how they might answer the problem statement.

5 DISCUSSION

So far, we have talked about some underlying theory, how we used the theory with our methods, and the results we gathered from the methods. Now we will tie it together to the best of our abilities.

The following problem statement was defined in [1.2](#) as such:

How could one design a captivating and sustainable web application that integrates a map feature, gathers helpful information for pilgrimages, and incorporates stories from Gjøvik? Would developing such an application increase interest in the topic of pilgrimage and the local history of Gjøvik?

The discussion chapter will be divided into 3 sections that will help us answer this problem statement: *Design and development, Sustainability and accessibility, and Creating interest.* Following this, we will discuss what can be done for further development.

5.1 DESIGN AND DEVELOPMENT

In this section we will discuss the methods that are tied to the design and development part of the bachelor process.

5.1.1 STRENGTHS AND WEAKNESSES OF OUR DESIGN AND DEVELOPMENT

5.1.1.1 COMMUNICATION WITH OUR TUTOR

Although we found our tutor's feedback and guidance during meetings valuable, we feel that scheduling more evenly spaced meetings throughout the process of creating our bachelor project could have resulted in further improvement.

5.1.1.2 TARGET AUDIENCE

Our target audience was determined to be individuals residing in Gjøvik (*see [3.1.1](#)*). To cater to this audience, we linked the stories on our map to Gjøvik. Although we recognize that we could have taken additional steps to better engage our target audience, our aim was to create a scalable solution that wasn't limited to a single group of individuals.

5.1.1.3 COMPETITOR ANALYSIS

Our competitive analysis relies on our personal experiences with the various competitors' solutions that we gained specifically for this project. This imposes a limit on the accuracy of our analysis. It would be preferable to obtain input from users who have first-hand experience with these solutions.

5.1.1.4 THEME

Our current theme is loosely characterized as “nature”, and we attempted to adhere to a specific colour scheme. The project would have benefited from a robust graphic profile accompanied by a design manual. This would have improved the project's overall visual consistency.

5.1.1.5 SCALABILITY AND QUALITY MODERATION

The project has the potential to incorporate user-generated content into the map, making it scalable to some degree. However, the current version of the project is not optimized for a larger market and struggles to render multiple large images simultaneously, resulting in significant lag and long loading times. To enhance the user experience and enable smooth navigation of the map page, optimization of the project's rendering capabilities is necessary.

Yggdrasil does not currently have a login system in place. To maintain quality control, user additions must be approved in MongoDB Atlas before they appear on the map. However, this method will not be effective when the number of additions increases significantly. To address this issue, it would be preferable to implement a login system with separate admin privileges, allowing moderators to approve additions through an admin interface instead of manipulating the data directly in the MongoDB database.

Without the quality control measures in place through MongoDB Atlas, there would be no moderation of user-generated content. This could result in a significant amount of unfiltered and potentially low-quality content on the site, ultimately leading to a decline in the overall quality of the platform. Poor quality content would hinder the site's ability to effectively convey culturally significant stories and highlight genuinely interesting places that users would want to explore.

5.1.1.6 WHY MERN?

After evaluating different technology stacks, we concluded that the other options available did not differ significantly enough from MERN to warrant investing time in learning them.

Hence, we decided to go with MERN, which we were already familiar with, to save time that would have been spent getting acquainted with a new stack.

5.1.1.7 TRANSCRIPTION OF THE NARRATIVES

Currently, there is only one transcribed story. The reason being that it was both time-consuming and tedious to implement. We could not find a transcription tool that met our desired quality standards, so we had to manually verify the start and end times of each word in the audio file we wanted to transcribe. The one narrative we transcribed serves as a proof of concept of the feature's capabilities.

Incorporating a transcribe feature into every modal could make the application feel more cohesive, but we concluded that our time would be better spent enhancing other aspects of our application.

5.1.1.8 COMPONENTS

Our application is designed with components that segregate crucial features, enabling code reuse when necessary. However, not all features are divided into simple components. We ought to have broken down some of the more complex components into smaller, separate ones to enhance Yggdrasil's performance and scalability.

5.1.1.9 EXPANSION OF MAP FEATURE

To make our application more scalable and interactive, we decided to let users add content to our website. However, during the application development process, we determined that our initial plan of creating a separate page to list user additions was uninspiring. To make the impact of user additions more engaging, we decided to display them as markers on the map instead. To differentiate these markers from the rurAllure narratives, we devised distinct appearances for each marker type on the map, encompassing all categories of user additions and rurAllure narratives.

5.2 SUSTAINABILITY AND ACCESSIBILITY

In this section we will discuss the measures we took to increase sustainability and accessibility during the development phase of the project.

5.2.1 IS THERE ROOM FOR IMPROVEMENTS?

We might question whether we have incorporated sufficient sustainability and accessibility measures into Yggdrasil. Although we have included a considerable number of such measures (as outlined in chapters [3.4](#) and [3.5](#)), there is still room for improvement across the board. As indicated in section [4.3.6](#), our lighthouse tests did not yield a perfect score of 100 in performance. Therefore, improvements could be made, to increase the lighthouse score.

Despite having implemented lazy loading and code splitting to some degree, it's possible that we could have enhanced it further by breaking it down into even smaller segments.

To further optimize Yggdrasil's efficiency, we could implement techniques like lazy loading and code splitting, that we explain in [2.2.14.1](#) and [2.2.14.2](#), for unused JavaScript code, such as “bundle.js”. This may result in a reduction in the amount of loaded, and unused JavaScript code that negatively affect the performance of Yggdrasil. It is worth mentioning that we tried to investigate how we could remove some of the unused code in “bundle.js” and other large folders, but due to our lack of knowledge and experience with this particular issue, we were unable to complete this task.

The application currently saves user-uploaded images in the “userImages” folder prior to sending them to MongoDB. Although these images are in “jpg” format, optimizing their file size and making them more suitable for the web by converting them to “WebP”, could enhance the application's performance as well as its sustainability.

Our website's accessibility was evaluated using a WAVE test (*see* [4.3.7](#)). The results indicated that there were no errors, but several alerts were present. These alerts mainly originated from the alt-text of images in arrays. While we were unable to include meaningful alt-text for these images, it should be noted that most of them were used for decorative purposes and did not contribute to the text on the pages.

In order to enhance the project's accessibility, we implemented tab-index and "aria label", but further testing is required to determine if our implementation is satisfactory.

5.3 CREATING INTEREST

5.3.1 GENERAL THOUGHTS ON USER TESTING

The number of participants for our tests was not optimal, and in order to accurately represent our solution's userbase, we should have included diverse age groups and people from different professional backgrounds. The userbase consisted solely of young student developers from NTNU, and the lack of diversity in the test subjects might have affected the precision of our findings. To obtain more meaningful responses, it would have been ideal to pose more effective questions.

5.3.1.1 LO-FI TESTING

There are several areas where the low-fidelity testing could have been improved. The primary issue with the low-fidelity prototype was that, aside from the wireframe, it only focused on the map page. We made this decision because we had a general idea of what would be on the landing and info pages, but we had not yet determined the specific content to display on those pages. Additionally, we were awaiting clearance on what we were permitted to showcase in our prototype at that time.

We concluded that it would be more efficient to conduct the user test solely on the map page since it was the most critical page. Our reasoning was that the map page would consume the most development time, so instead of delaying testing until everything was ready, it was crucial to start testing it right away.

5.3.1.2 HI-FI TESTING

The hi-fi prototype was split into two versions - one for mobile and the other for desktop. During testing, the former group of test subjects evaluated the desktop version, while the latter group tested the mobile version. The testing would have benefitted from having the test subjects focus on one version. As a result of testing two separate versions, we were unable to conduct a thorough evaluation of either one.

We hold the belief that the outcomes of the testing would not be significantly impacted by having a greater number of users. This is due to the fact that the feedback provided by the users was predominantly consistent with one another. Hence, it is logical to infer that enlarging the testing pool would yield more comparable responses and fewer fresh insights.

5.3.1.3 APPLICATION TESTING

There are reasons to assume that the first impressions of the final application might not have been at its fullest potential. The users had to test the narratives, which had inconsistent

languages. Moreover, we recorded the audio files for the narratives ourselves, causing some to recognize the voice, which might have made the experience less formal.

While testing the application, we noticed that some users had a poor grasp of the term "pilgrim" and in turn the nature of our application. This could potentially impact the accuracy of their responses as they may not fully comprehend the questions and their intended meaning.

5.3.2 WAS INTEREST INCREASED?

In chapter [4.2.1.1](#), when we administered the SUS questionnaire, we primarily received positive responses and ended up with an average sus score of 79,25 and an A- grade in the grading scale. These results can be used to infer that our final application is quite usable. We would argue that an application that is usable makes it easier for the users to engage with the content it displays. The less usable the application is, the more likely it is that the user's level of interest might be affected negatively.

The response data we gathered throughout the project revealed a rise of 26.32% in the level of interest for "pilgrimage" and a rise of 17.39% in the level of interest for "local history of Gjøvik" after users experienced our application (*as seen in [4.2.1.2](#)*). Based on this we can infer that our application had a positive impact on the topics we aimed to enhance. Nevertheless, the reliability of this data is open to debate.

It may be considered whether the responses were significantly influenced by the geographical area of interest. However, we contend that none of the material is tailored exclusively for individuals residing in Gjøvik to enjoy. If we had altered the location of the narratives and posed identical inquiries to respondents from that area, we would anticipate observing comparable findings.

As we observed users navigating the website (*as seen in [4.2.1.3](#)*), we monitored their interactions with it. Our analysis revealed that on average, they stayed on the page for 5 minutes and 52 seconds, suggesting that they were sufficiently engaged, to avoid exiting the page. Moreover, the data indicated that users frequently clicked on both the links available on the information page and the points of interest (POIs) marked on the map page, indicating their interest in, and the utility of these features. While the map page appeared to be the most

popular, the difference in engagement levels between the map and information pages was minimal, making it difficult to draw definitive conclusions.

Upon examining the results of the interest questionnaire, it was observed that nearly all responses began at 2 and concluded at 3. Because we were present during the application testing, the participants of the testing might not have answered truthfully because they felt that their answers were not fully anonymous, despite the fact that we could not discern their specific responses. These factors raise concerns about the potential influence of social desirability bias, as explained in section [2.2.6.2](#).

Social desirability, which means to give an answer that is perceived as desirable, could have occurred during both the interest questionnaire and the SUS questionnaire. Since the testing participants knew us, the creators of the application, they may have deliberately responded in a manner that would please us and meet our expectations.

Another question to consider is the extent to which our results may have been influenced by convenience sampling, which is a form of selection bias. Due to time constraints, we solely reached out to and polled individuals from our university who were easily accessible. It is worth exploring whether students are the appropriate population to survey or if a larger, more diverse sample should be recruited to enhance the generalizability of the findings.

The people we ended up testing were ultimately those who showed up. Nonresponse bias suggests that people who are willing to answer a survey might be fundamentally different than those who are not (*see* [2.2.6.2](#)). This is a challenge that is difficult to combat, and it is important to acknowledge when collecting data from individuals.

Although we received a satisfactory number of responses during the final user test, a larger sample size would be helpful to accurately measure the increase in interest. Moreover, making Yggdrasil accessible to a wider audience by deploying it to something like Netlify, would likely provide us with a definitive answer regarding the extent of the interest increase.

The discussion chapter's findings indicate that our solution has sparked some level of interest. However, it is possible that the perceived magnitude of the increase could be overstated due to potential biases. Furthermore, the limitations of our solution are evident, indicating that it is not without flaws.

Yggdrasil has adopted various strategies to improve website accessibility and sustainability. Nevertheless, there is still potential for enhancement, and given more time, we could introduce additional measures to achieve these goals.

Our application features an interactive map with intriguing markers that users can enhance by adding their own. Furthermore, we have incorporated a page that consolidates relevant information in a single location. The insights we have gathered through multiple user tests indicate that interest was increased in pilgrimage and the local history of Gjøvik, but the interest might appear larger than what it is in actuality, due to bias.

5.4 FURTHER DEVELOPMENT

There are some matters that we would like to address in the event that other researchers pursue further development.

As this was a task we did while in university, we did not allocate any funds to our research. Things like the Google Maps API, questionnaire licences and hosting services require financial resources and as a result, we may have been limited in our choices and opted for free alternatives. For further research, it may be advantageous to invest in some of these resources to generate more reliable results.

If one were to conduct similar user tests it would be recommended to increase the testing pool, minimizing selection bias where possible, and ensure that none of the testers have relations with the administrators of the tests, to prevent social desirability bias.

6 CONCLUSION

We started our thesis mentioning the problem statement as such:

How could one design a captivating and sustainable web application that integrates a map feature, gathers helpful information for pilgrimages, and incorporates stories from Gjøvik? Would developing such an application increase interest in the topic of pilgrimage and the local history of Gjøvik?

We divided the problem statement into three distinct areas of concern to guide us in addressing the problem. Our thesis delved into these areas by utilizing a structured approach that included a theory chapter (2), methods chapter (3), results chapter (4), and a discussion chapter (5).

The first area of concern which is “***Designing a web application that will feature a map, and a place to store helpful information for people that want to start their pilgrimage journey.***” was done by using competitive analysis, SWOT, personas, wireframing, low-fidelity prototypes and high-fidelity prototypes. This was written in section 3.1 and the results displayed in 4.1.

Then for the second area of concern which is “***Develop features that increases interest in pilgrimage and the local history of Gjøvik.***” we developed an application that has an information page with useful insights and a map page that showcased Gjøvik, with multiple markers that would display different things such as views and stories, written in 3.2 and 3.3, with the results in 4.2. It was done with the design methodologies as well as the addition of the MERN stack (see 3.2). We gauged the interest by conducting questionnaires (see 3.3.2) then displayed those answers in 4.2.

Lastly, our final area of concern “***Ensure that the application is accessible and sustainable***” was tackled by implementing various measures. These measures included reducing file sizes through compression and converting image and audio files to a different and better format, implementing tab index, ensuring good overall contrast, eliminating redundant code, utilizing code splitting and lazy loading. At the end, we measured Yggdrasil’s accessibility and

sustainability with the WAVE and Lighthouse tools. All the following measures are mentioned in section [3.4](#) and [3.5](#) and the impact of these is presented in [4.3](#).

During our discussion, we reviewed our work and findings to determine whether we could address the questions presented in the problem statement. We found that we had developed a sustainable application that made the users more interested in pilgrimage and stories from Gjøvik. We acknowledge that the results could have been affected by bias and found it reasonable to assume that those effects could have caused our results to be more positive, but that it ultimately made a minimal impact in the outcome.

Our recommendation is for further work to be conducted in order to obtain more conclusive results. Additionally, we suggest deploying a comparable solution to the market to assess its viability in other regions.

7 BIBLIOGRAPHY

- Adobe. (n.d.). *The best audio format types for audiophiles*. Retrieved from <https://www.adobe.com/creativecloud/video/discover/best-audio-format.html>
- Amundsen, S., Randen, O. R., & Stenbråten, M. (2022). *IDG3101 In-depth-project, Project report*. Undergraduate Dissertation.
- Apel, T. (2022, December 7). *Yggdrasil*. Retrieved from Mythopedia: <https://mythopedia.com/topics/yggdrasil>
- Babich, N. (2017, November 29). *Prototyping 101: The difference between low-fidelity and high-fidelity prototypes and when to use each*. Retrieved from Adobe Blog: <https://blog.adobe.com/en/publish/2017/11/29/prototyping-difference-low-fidelity-high-fidelity-prototypes-use>
- Bailey, E. (2021, January 28). *Use the tabindex attribute*. Retrieved from The a11y project: <https://www.a11yproject.com/posts/how-to-use-the-tabindex-attribute/>
- Bates, S. (2020, December 17). *What is axios*. Retrieved from Codebots: <https://codebots.com/docs/what-is-axios>
- Baxter, K., Courage, C., & Caine, K. (2015). *Understanding Your Users: A Practical Guide to User Research Methods* (2nd ed.). Amsterdam: Morgan Kaufmann.
- Binge On Code. (2021, September 15). *What Are Code Splitting And Lazy Loading?* Retrieved from Medium: <https://javascript.plainenglish.io/code-splitting-and-lazy-loading-b9cbcd05494d>
- Brooke, J. (1996). SUS: A 'quick and dirty' usability scale. In P. W. Jordan, B. A. Thomas, B. Weerdmeester, & A. L. McClelland (Eds.), *Usability evaluation* (pp. 189-194). London: Taylor & Francis.
- Broz, M. (2022, October 21). *Image Format Comparison (JPEG, PNG, WEBP, & AVIF) – 2023 Statistics*. Retrieved from Photutorial: <https://photutorial.com/image-format-comparison-statistics/>
- Dam, R. F., & Siang, T. Y. (n.d.). *Personas - A Simple Introduction*. Retrieved April 17, 2023, from Interaction Design Foundation: <https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them>
- ExpressJS. (n.d.). *ReadME : Multer Build Status*. Retrieved May 13, 2023, from Github: <https://github.com/expressjs/multer>
- GitHub. (n.d.). *Hello World*. Retrieved May 13, 2023, from GitHub Docs: <https://docs.github.com/en/get-started/quickstart/hello-world>
- Google. (2023, March 1). *Maps JavaScript API*. Retrieved from Google Maps Platform: <https://developers.google.com/maps/documentation/javascript#features>
- Google Chrome. (2022, May 24). *Overview*. Retrieved from Chrome Developers: <https://developer.chrome.com/docs/lighthouse/overview/>
- Hoque, S. (2020). *Full-stack react projects : learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node.js*. Birmingham: Packt Publishing.

- Ibragimova, E. (2016, December 28). *High-fidelity prototyping: What, When, Why and How?* Retrieved from Medium: <https://blog.prototypr.io/high-fidelity-prototyping-what-when-why-and-how-f5bbde6a7fd4>
- Karakaya, H. (2023, March 2). *Build an accessible modal with focus-trap-react*. Retrieved from LogRocket: <https://blog.logrocket.com/build-accessible-modal-focus-trap-react/>
- Kenton, W. (2023, April 21). *SWOT Analysis: How To With Table and Example*. Retrieved April 17, 2023, from Investopedia: <https://www.investopedia.com/terms/s/swot.asp>
- Kucic, A., & Vlaeva, S. (2023, April 13). *MongoDB & Mongoose: Compatibility and Comparison*. Retrieved from MongoDB: <https://www.mongodb.com/developer/languages/javascript/mongoose-versus-nodejs-driver/>
- Luo, H. (2022, September). *react-h5-audio-player*. Retrieved from Github: <https://github.com/lhz516/react-h5-audio-player>
- Marketing Evolution. (2022, July 20). *Steps to Find Your Target Audience*. Retrieved from <https://www.marketingevolution.com/marketing-essentials/target-audience>
- Material UI . (2023, April 20). *Material UI - Overview*. Retrieved from Material UI : <https://mui.com/material-ui/getting-started/overview/>
- MDN. (2023a, April 12). *ARIA*. Retrieved from Mdn Web docs: <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA>
- MDN. (2023b, February 21). *SPA (Single-page application)*. Retrieved from MDN web docs: <https://developer.mozilla.org/en-US/docs/Glossary/SPA>
- MDN. (2023c, May 10). *Image file type and format guide*. Retrieved from Mdn web docs: https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Image_types
- Merriam-Webster. (n.d.). *pilgrim*. Retrieved May 13, 2023, from merriam-webster.com: <https://www.merriam-webster.com/dictionary/pilgrim>
- MongoDB. (n.d.-a). *MongoDB MERN Stack Explained*. Retrieved April 15, 2023, from MongoDB Web site: <https://www.mongodb.com/mern-stack>
- MongoDB. (n.d.-b). *Why Use MongoDB and When to Use It?* Retrieved April 17, 2023, from MongoDB Web site: <https://www.mongodb.com/why-use-mongodb>
- Oberlo. (n.d.). *WHAT PERCENTAGE OF INTERNET TRAFFIC IS MOBILE?* Retrieved April 17, 2023, from <https://www.oberlo.com/statistics/mobile-internet-traffic>
- Omniconvert. (2023, January 17). *What is... User testing*. Retrieved April 18, 2023, from <https://www.omniconvert.com/what-is/user-testing/>
- PageFly Academy. (n.d.). *Mobile First Vs Responsive Design: What Is Right For Your Business?* Retrieved May 14, 2023, from <https://academy.pagefly.io/guide/mobile-first-vs-responsive-design/>
- Panagiotidi, M. (2021, December 23). *Standardized usability questionnaires: post-task measures*. Retrieved from UX Psychology: <https://uxpsychology.substack.com/p/standardized-usability-questionnaires-ccb>

- Pesot, J., & Plantenberg, S. (n.d.). *Define Personas*. Retrieved April 17, 2023, from IBM:
https://www.ibm.com/garage/method/practices/think/practice_personas/
- Prokopets, M. (n.d.). *The Beginner's Guide to Chrome Developer Tools*. Retrieved May 13, 2023, from nira Blog:
<https://nira.com/chrome-developer-tools/>
- React Google Maps Api. (n.d.). *React Google Maps Api Style Guide*. Retrieved May 13, 2023, from <https://react-google-maps-api-docs.netlify.app/>
- React Router. (n.d.). *Feature Overview*. Retrieved May 13, 2023, from <https://reactrouter.com/en/main/start/overview>
- Ritter, M., & Winterbottom, C. (2017). *UX for the Web : Learn How UX and Design Thinking Can Make Your Site Stand Out From the Rest of the Internet*. Birmingham: Packt.
- rurAllure. (2023, April 14). *rurAllure About*. Retrieved April 14, 2023, from rurAllure Web site:
<https://rurallure.eu/project/about/>
- Sandmæl, C. O., Hansen, G. E., & Schrier, T. (2022). *uniLeague – Measuring student engagement in IDG2100 at NTNU in Gjøvik*. (Bachelor Thesis, NTNU). Retrieved from <https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/3004841>
- Sauro, J., & Lewis, J. R. (2016). *Quantifying the User Experience : Practical Statistics for User Research* (2nd ed.). San Francisco: Morgan Kaufmann.
- Sharma, A. (2023, February 7). *A Complete And Comprehensive Guide To Understand Body Parser In Express JS*. Retrieved from Simplilearn: <https://www.simplilearn.com/tutorials/nodejs-tutorial/body-parser-in-express-js>
- Tailwind Labs Inc. (n.d.). *TailwindCSS*. Retrieved May 13, 2023, from TailwindCSS: <https://tailwindcss.com/>
- UIUXTrend. (2021, February 9). *Measuring and Interpreting System Usability Scale (SUS)*. Retrieved from <https://uiuxtrend.com/measuring-system-usability-scale-sus/>
- Unadkat, J. (2022, June 1). *Mobile First Design: What It Is and How to Implement It*. Retrieved April 17, 2023, from BrowserStack: <https://www.browserstack.com/guide/how-to-implement-mobile-first-design>
- Usability.gov. (n.d.). *Prototyping*. Retrieved April 18, 2023, from <https://www.usability.gov/how-to-and-tools/methods/prototyping.html>
- Vassilyovska , M. (2022, December 1). *What is Sustainable Web Design and How to Achieve It*. Retrieved from Kualo: <https://www.kualo.com/blog/sustainable-web-design>
- W3C Web Accessibility Initiative . (2023, March 31). *Introduction to Web Accessibility*. Retrieved from Web Accessibility Initiative: <https://www.w3.org/WAI/fundamentals/accessibility-intro/>
- W3schools. (n.d.). *Node.js NPM*. Retrieved May 13, 2023, from https://www.w3schools.com/nodejs/nodejs_npm.asp
- WAVE. (n.d.). *WAVE Web Accessibility Evaluation Tools*. Retrieved May 13, 2023, from Web Accessibility Evaluation Tool: <https://wave.webaim.org/>

8 FIGURE LIST

Figure 1 Mockups for our 3 main pages	11
Figure 2 Add to map page.....	13
Figure 3 [Overview of the percentage of internet traffic that comes from mobile] (Oberlo, n.d.)	21
Figure 4 [THE MERN STACK ARCHITECTURE VISUALIZED]. (MongoDB, n.d.-a)	25
Figure 5 Example of how our personas look.....	34
Figure 6 Table showing the comparisons between possible competitors.....	35
Figure 7 First iteration "wireframe" low fidelity prototype.....	36
Figure 8 Low fidelity version of our map page	37
Figure 9 Low fidelity prototype of map modal	37
Figure 10 The different versions of the modal "rurAllure story" in the hi-fi prototype	39
Figure 11 The iteration on our navigation bar.....	40
Figure 12 Final desing of Yggdrasil's logo	41
Figure 13 The project folder structure of my-app	42
Figure 14 Structure of the frontend	43
Figure 15 Example of the use of media query prefixes	44
Figure 16 Structure of the backend	45
Figure 17 Mongoose schema wrapped in a model.....	46
Figure 18 Changes to the styling of the map	47
Figure 19 The different changes on the map.....	48
Figure 20 Lazy loading step by step	51
Figure 21 Picture of the Chosen colour palette with HEX codes	53
Figure 22 Example of tab Index	53
Figure 23 Picture of our transcribe functionality.....	53
Figure 24 Swot analysis of our competitors as well as ourselves	57
Figure 25 USABILITY COMPETITIVE ANALYSIS RESULTS.....	58
Figure 26 Calculating the Sus Score in excel	60
Figure 27 Excel table of sus score.....	60
Figure 28 Grading scale for SUS (Sauro & Lewis, 2016, p. 204).....	61
Figure 29 Interest score from user testing	61
Figure 30 the results of the question "did you learn anything new while visiting the site?"	62
Figure 31 Average result of observation	62
Figure 32 THROTTLING SETTINGS UNDER THE TEST.....	63
Figure 33 Network tool overview on "landing" page	64
Figure 34 Network tool overview on "info" page.....	65
Figure 35 Network tool overview on "map" page	66
Figure 36 project folder size (excluding node_modules and userImages) Before removing redundant code.....	67
Figure 37 VISUAL REPRESENTATION OF UNUSED BYTES IN OUR PROJECT CODE	67
Figure 38 EXAMPLE OF FILE COMPRESSION BY USING WEBP	68
Figure 39 The size reduction of webp image on info image.....	69
Figure 40 The project Folder size after converting and compressing images.	69
Figure 41 "Mjosormen.wav" file details	69
Figure 42 Reduction of project folder size.....	70
Figure 43 Lighthouse test results.....	71
Figure 44 Summary of Wave test	71

9 APPENDIX

This chapter will contain all the extra info that we could not fit into the rest of the thesis.

9.1 OUR PERSONAS

This is a complete overview of the personas we made.

Personas

Lars

Lars Konrad Hovde
73 / Male

Has a lot of local stories to tell that finds place in Gjøvik. He feels that the youth today are missing out on many wonderful tales that were told when he was young. He's a librarian for Gjøvik bibliotek, and therefore has access to knowledge that is not commonplace.

User Traits

Forgetful Calm Generous
Cool Passionate

Eva

Eva Marie Solhaug
32 / Female

Nurse working in Gjøvik with a passion for nature photography. She is always on the lookout for more views to capture on film. Although she enjoys scouting for new locations, she has difficulty finding a place for it in her busy schedule.

User Traits

Kind Explorative Generous
Artistic Nosy

Stian

Stian Johanssen
19 / Male

Just finished high school and decided that he wants to take a gap year where he travels. His cousins recommended traveling on the pilgrim trail close to where he lives. He is worried that he might not know what it takes to walk a path like that.

User Traits

Aspiring Restless Fit
Curious Active

Heidi

Heidi Mortensen
42 / Female

As a history teacher, Heidi has always had a passion for learning about the past. Lately, she has developed a growing interest in delving deeper into the history of her local area, and is now actively seeking out intriguing local stories and histories to explore.

User Traits

Curious Calm Happy
Passionate

9.2 THE QUESTIONS AND ANSWERS FROM THE LO-FI PROTOTYPE TESTING

These are the questions and answers we received when we user tested the low-fidelity prototype.

Does the idea behind the audio files sound interesting?

1. Yes
2. Yes, but a poster on site would be just as good.
3. Yes, if you are interested in local history. (Geofencing, what is nearest your position)

Do you become interested in learning more about/visiting the locations the stories take place?

1. Yes
2. Yes, especially if you are rewarded by doing so.



What modal design do you like the best (Added above for reference)?

Note: None of the users answered this question directly, but they talked us through what they liked and disliked about the different versions. The general consensus pointed to V2 of the modal being the preferred option.

Anything else you would like to add?

1. Get inspiration from Strava/Stolpejakt
2. Should be possible to toggle between highlighted text and non-highlighted text.
3. Some form of gameification? Scoreboard/leaderboard etc.

9.3 THE Q AND A, TASKS AND ADDITIONAL NOTE FROM THE HI-FI PROTOTYPE TESTING

These are the questions we asked, the answers we received, the tasks we gave the participants, and other notes we took when we user tested the high-fidelity prototype.

Note from questions

Did you understand which page you were on?

Desktop:

- Yes x5

Mobile:

- Meh, No. Maybe show an active link
- Yes

Is there anything on the prototype that needs changes? Is the design messy, if yes what?

Desktop:

- What is the difference between "Info" and "Explore"?
- What is the main focus of the website? (We need to clarify this)
- We should change the title of the website.
- There is a lot of white space between the sections on the "Info" page. We need to fill it in with something.
- The lines under the navbar should have a stronger color and be a bit closer to the "Home" button.

Mobile:

- We need to display an active link and the header.
- We should reconsider the wording for "Listen to the stories" button as users were confused by the map.
- We can try different names such as "Journey Map", "Story Map", or "Map Journey".
- We need to make the website more organized and less confusing.

What is your opinion on the design colors?

Desktop:

- It's okay.
- Yes, it's nice.
- It fits the theme, but the color usage on the landing page is low contrast.
- It suited the theme well.
- The quote is very eye-catching, maybe a bit too much color.

Mobile:

- I like the colors, good text contrast and layout. Works well on mobile.
- Looks clear, but I wouldn't have used those colors myself.

Is a landing page necessary?

Desktop:

- No.
- No, not as it is now. Change the information on the landing page. Write more about us.
- Yes, it's good to have something to land on.
- Yes or no, it depends on what's on it.
- Yes, it's good for those who haven't been here before.

Mobile:

- Yes, it is useful for the people who has not been on the page before

What design iteration did you like the best?

Note:

- The play button is there, but when you click it, the mp3 play bar pops up.
- Scrolling is preferable, as it would be awkward to split the text into sections.
- I appreciated the segmented format and the ability to have a button that plays audio.
- Scrolling is likely the best option. The play button is useful, as it lets users know that there is a story to be listened to, rather than background music, for example.
- The swipe function works best on mobile.
- I liked the play button and suggest making it a different color to stand out. It could change to a pause button after being clicked and be styled like a typical media player with a "Resume Story" option.

Other notes

Note:

- We need to modify the landing page, as it doesn't quite make sense.
- The shape of the bottom div looks a bit odd and doesn't fit in well.
- It's not immediately clear what is under "Info" and "Explore".
- It would be helpful to have a toggle on the map with the story boxes.
- When hovering over the pinpoint on the map, a picture should be displayed.
- The mp3 chrome feature is not accessible to everyone, so we need to make some changes.
- We require CMS functionality if users want to upload their own stories.
- We should save data in JSON format.
- We also need to highlight the text and switch the segments automatically.
- Can we combine the "Home" and "A Guide to Pilgrimage" sections on the same page?
- We should aim for a minimum button size of 40 or 44 pixels on mobile.
- It's essential that the hitbox of the button is large.

Note from user task

Find the story of the "Mjøsoorm"

Desktop Note:

- The user clicked "Info" in the navbar first, but navigated easily to the map afterwards.
- User stated they were confused by the titles in the navbar. "Consider changing the title of the navbar"
- Overall, the user had a good experience.
- The user was able to find what they were looking for easily.
- Found the story easily.

Mobile Note:

- Tried to use the CTA button, not properly implemented, had to use the navbar.
- Used navbar, it went well.

Go back to the landing page.

Desktop Note:

- Went well, used navbar.
- Went well, used logo.
- Went well, used navbar.
- Went well, used logo.

Mobile Note:

- Clicked the logo. Went well.
- Clicked the logo. Went well.

Find information about the "Olavsveiene"

Desktop Note:

- Went well.
- Went well.
- Went well.
- Went well, but user was unsure which button to use. ("Info" or "Learn more about Olavsveiene").
- Didn't start scrolling immediately. "It would be helpful to animate the arrow downwards to encourage scrolling."

Mobile Note:

- Used navbar, went to the map page first.
- Went to the map page first.

9.4 APPLICATION NAME SUGGESTIONS

These were the names we considered for our application.

- Trippstur/trippstr
- Sojourn
- UtpåTur
- Journey
- NorthQuest
- Norsewalker
- Tripster/trip
- findOlav
- Skattejakten
- find.no
- OlavsFølge
- NidarosQuest
- DiscoverRurality
- Yggdrasil

9.5 SCHEMA OVERVIEW

This is an overview of the fields and type of the Mongoose schemas we used.

Schema	Field	Type
additionsSchema	email	String
	additionType	String
	latitude	Number
	longitude	Number
	title	String
	additionType	String
	approved	String

rurStorySchema	title	String
	text	String
	lat	Number
	long	Number

rurTranscriptSchema	title	String
	lat	Number
	long	Number
	src	String

viewAdditionSchema	email	String
	additionType	String
	latitude	Number
	longitude	Number
	additionImg	Buffer
	title	String
	addition	String
	approved	String

9.6 STANDARD QUESTIONNAIRES QUESTION SETS

Here you can find the entire questionnaire question sets from each of the following (except QUIS and SUMI because they required a license).

9.6.1 PSSUQ

The Post-Study System Usability Questionnaire Version 3		Strongly agree							Strongly disagree	NA
		1	2	3	4	5	6	7		
1	Overall, I am satisfied with how easy it is to use this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	It was simple to use this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	I was able to complete the tasks and scenarios quickly using this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	I felt comfortable using this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	It was easy to learn to use this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	I believe I could become productive quickly using this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	The system gave error messages that clearly told me how to fix problems.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	Whenever I made a mistake using the system, I could recover easily and quickly.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	The information (such as online help, on-screen messages, and other documentation) provided with this system was clear.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	It was easy to find the information I needed.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11	The information was effective in helping me complete the tasks and scenarios.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
12	The organization of information on the system screens was clear.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
13	The interface* of this system was pleasant.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
14	I liked using the interface of this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
15	This system has all the functions and capabilities I expect it to have.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
16	Overall, I am satisfied with this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

*The "interface" includes those items that you use to interact with the system. For example, some components of the interface are the keyboard, the mouse, the microphone, and the screens (including their graphics and language).

(Sauro & Lewis, 2016, p. 193)

9.6.2 SUS

The System Usability Scale Standard Version		Strongly disagree		Strongly agree		
		1	2	3	4	5
1	I think that I would like to use this system frequently.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	I found the system unnecessarily complex.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	I thought the system was easy to use.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	I think that I would need the support of a technical person to be able to use this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	I found the various functions in the system were well integrated.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	I thought there was too much inconsistency in this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	I would imagine that most people would learn to use this system very quickly.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	I found the system very awkward to use.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	I felt very confident using the system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	I needed to learn a lot of things before I could get going with this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(Sauro & Lewis, 2016, p. 198)

The System Usability Scale Positive Version		Strongly disagree		Strongly agree		
		1	2	3	4	5
1	I think that I would like to use the website frequently.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	I found the website to be simple.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	I thought the website was easy to use.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	I think that I could use the website without the support of a technical person.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5	I found the various functions in the website were well integrated.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	I thought there was a lot of consistency in the website.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	I would imagine that most people would learn to use the website very quickly.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8	I found the website very intuitive.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9	I felt very confident using the website.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	I could use the website without having to learn anything new.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(Sauro & Lewis, 2016, p. 208)

9.6.3 UMUX

The Usability Metric for User Experience Version 1		Strongly agree				Strongly disagree		
		1	2	3	4	5	6	7
1	This system's capabilities meet my requirements.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Using this system is a frustrating experience.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	This system is easy to use.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4	I have to spend too much time correcting things with this system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(Sauro & Lewis, 2016, p. 211)

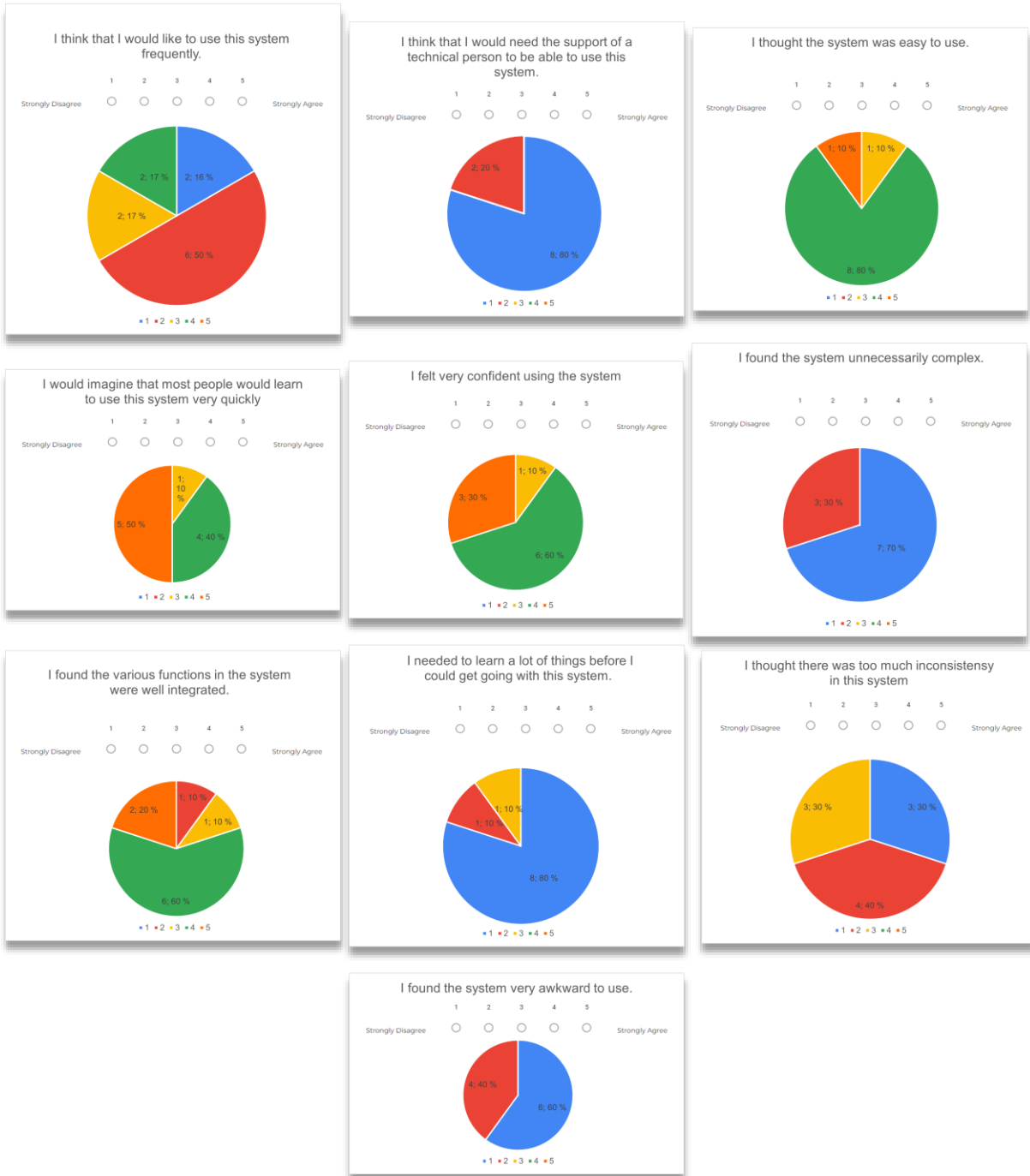
9.6.4 UMUX-LITE

The UMUX-LITE Version 1		Strongly agree				Strongly disagree		
		1	2	3	4	5	6	7
1	This system's capabilities meet my requirements.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	This system is easy to use.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(Sauro & Lewis, 2016, p. 213)

9.7 SUS DATA

This is the data we collected from the SUS questionnaire when we tested the final application.



9.8 OUR QUESTIONNAIRE QUESTION SET

These are the questions from the questionnaire we made to test our final application.

User testing Yggdrasil - 03.05

Before testing

On a scale of 1 to 5, how interested are you in pilgrimage?

1 2 3 4 5

Not interested Very interested

On a scale of 1 to 5, how interested are you in the local history of Gjøvik?

1 2 3 4 5

Not interested Very interested

After testing

On a scale from 1 to 5, how interested are you in the pilgrimage?

1 2 3 4 5

Not interested Very interested

On a scale of 1 to 5, how interested are you in the local history of Gjøvik?

1 2 3 4 5

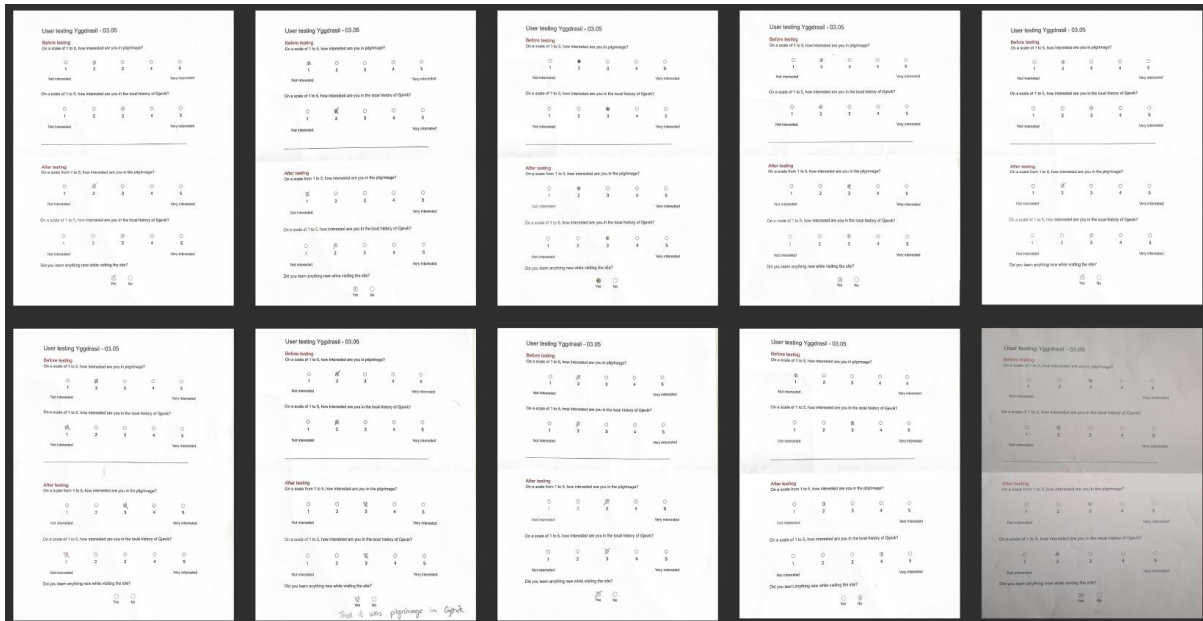
Not interested Very interested

Did you learn anything new while visiting the site?

Yes No

9.9 THE ANSWERS FROM THE SMALL QUESTIONNAIRE WE CREATED

These are the answers from the questionnaire we made to test our final application.



USERS	On a scale of 1 to 5, how interested are you in pilgrimage? BEFORE	On a scale of 1 to 5, how interested are you in the local history of Gjevik? BEFORE	On a scale from 1 to 5, how interested are you in the pilgrimage? AFTER	On a scale of 1 to 5, how interested are you in the local history of Gjevik? AFTER	Did you learn anything new while visiting the site?
User 1	2	3	2	3	3 yes
User 2	2	2	3	3	3 yes
User 3	2	2	3	3	3 yes
User 4	2	1	3	3	1 no answer
User 5	1	3	2	2	4 no
User 6	2	3	2	2	3 yes
User 7	2	3	3	3	3 yes
User 8	2	2	2	3	3 yes
User 9	1	2	1	3	2 yes
User 10	3	2	3	1	2 yes
TOTAL	19	23	24	27	88.89% learned something. (8 yes, 1 no.)
AVG	1,9	2,3	2,4	2,7	

9.10 USER HABITS WHEN EXPLORING YGGDRASIL

This is the data of what we observed when users browsed our application.

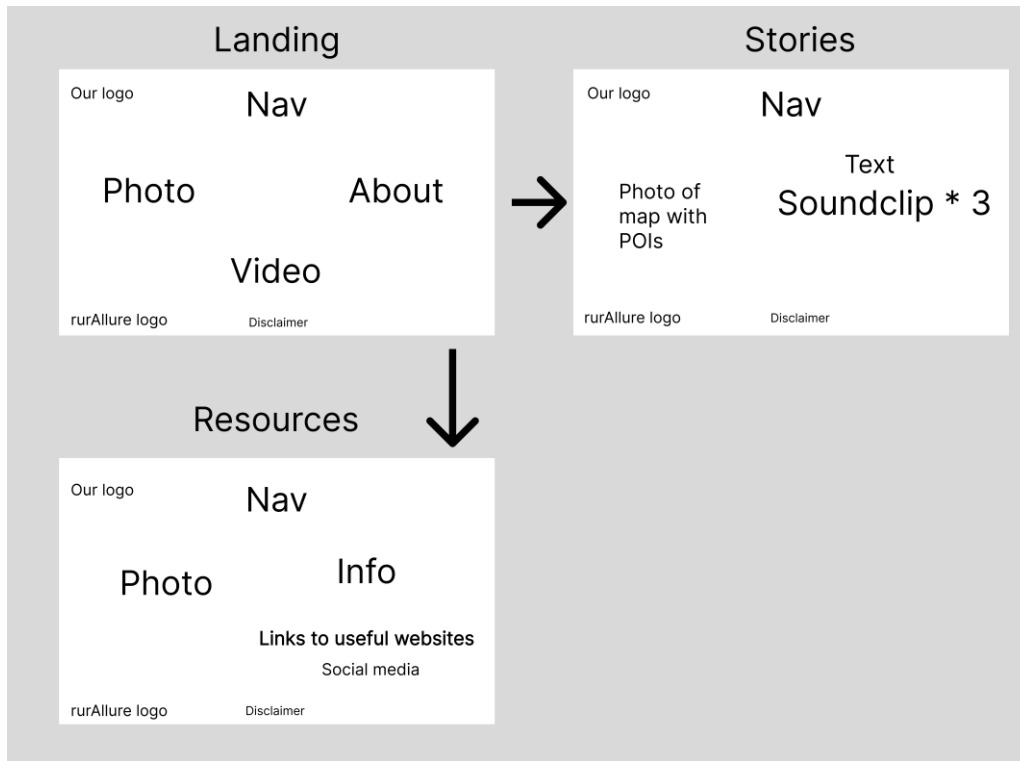
User-testing note:

User tester	Links clicked	POIs clicked	Page spent most time on	Total time spent on Yggdrasil
1	1	8	Map/Info	14:05.87
2	3	9	Map/Info	8:37.70
3	0	3	Info	3:52.31
4	0	4	Map	2:17.29
5	0	3	Map/Info	4:25.67
6	0	7	Map/Info	6:24.98
7	1	7	Map	3:15.61
8	0	5	Map/Info	5:00.09
9	1	4	Map	5:41.16
10	6	2	Info	5:05.48

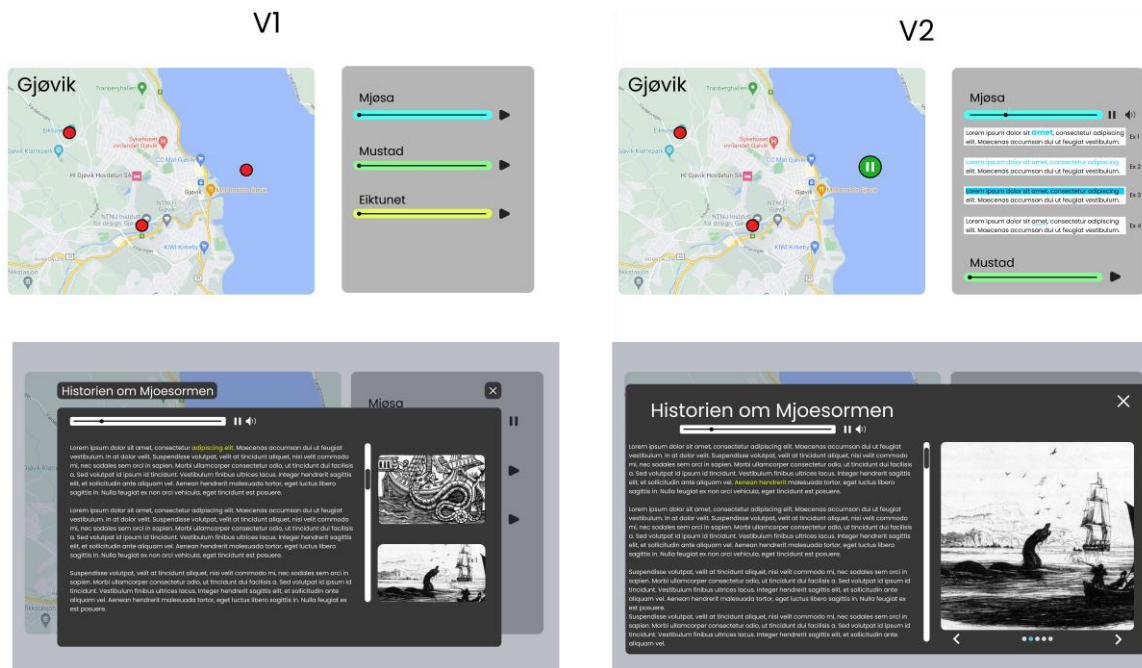
9.11 THE ENTIRE LOW-FIDELITY PROTOTYPE

This is how our entire low-fidelity prototype looks like, divided into the wireframe and the map page.

9.11.1 WIREFRAME



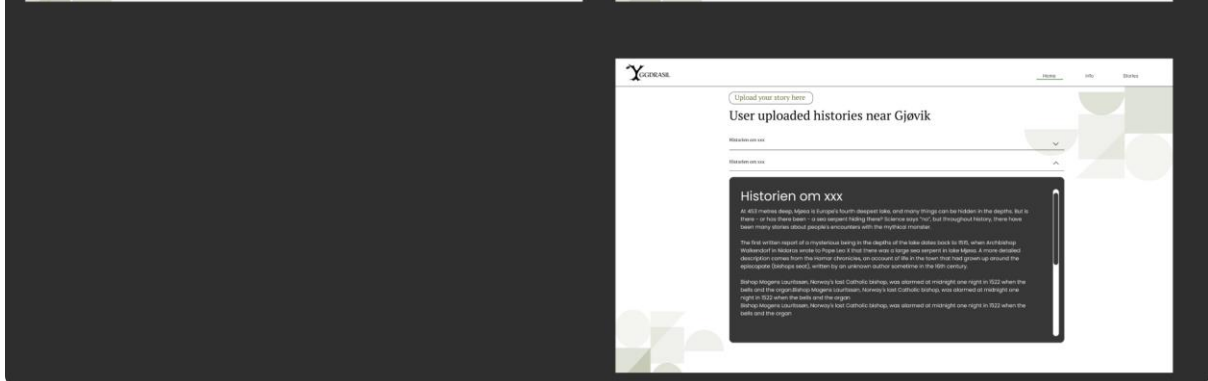
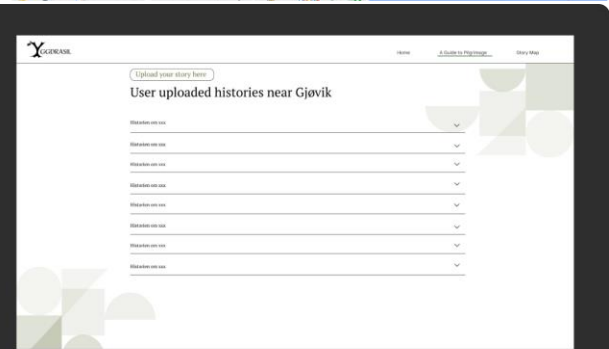
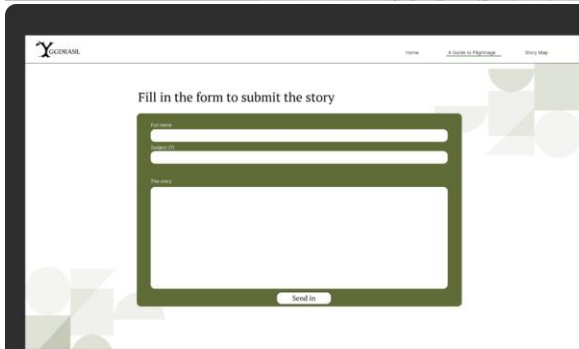
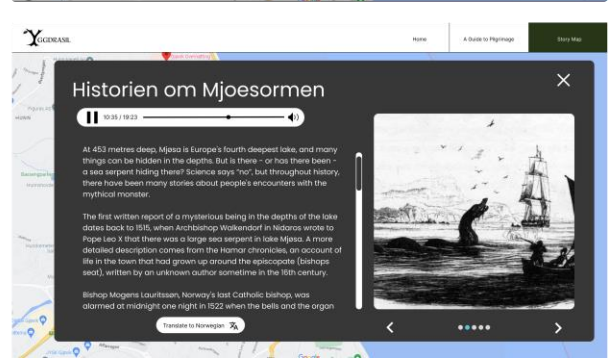
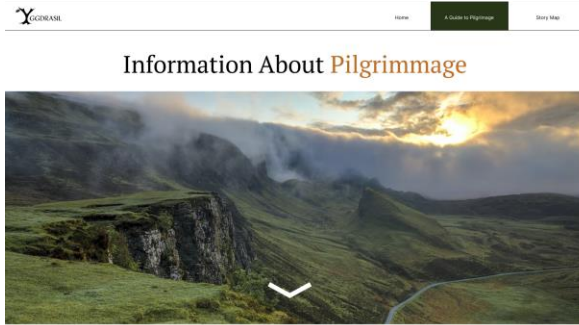
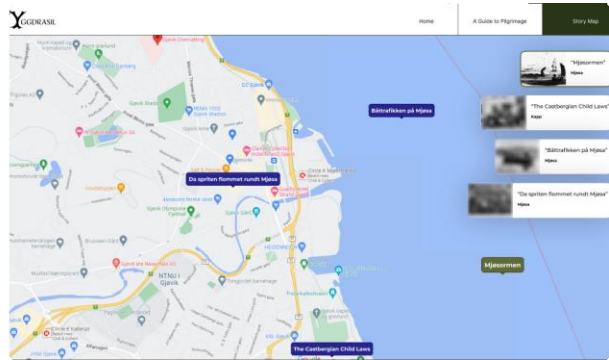
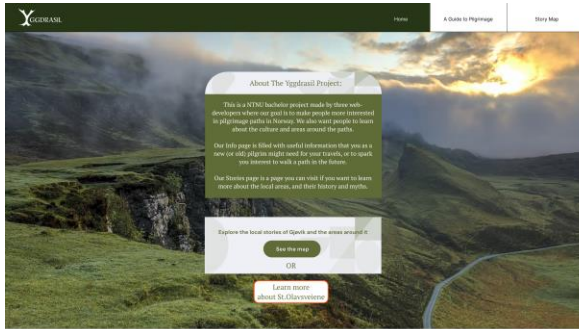
9.11.2 MAP PAGE



9.12 THE ENTIRE HIGH-FIDELITY PROTOTYPE

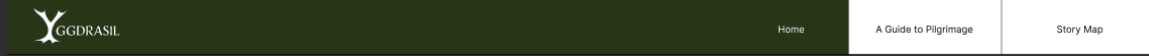
This is how our entire high-fidelity prototype looks like divided into the desktop version and the mobile version.

9.12.1 DESKTOP



Navbar desktop

navbar-45



navbar-45



navbar-45

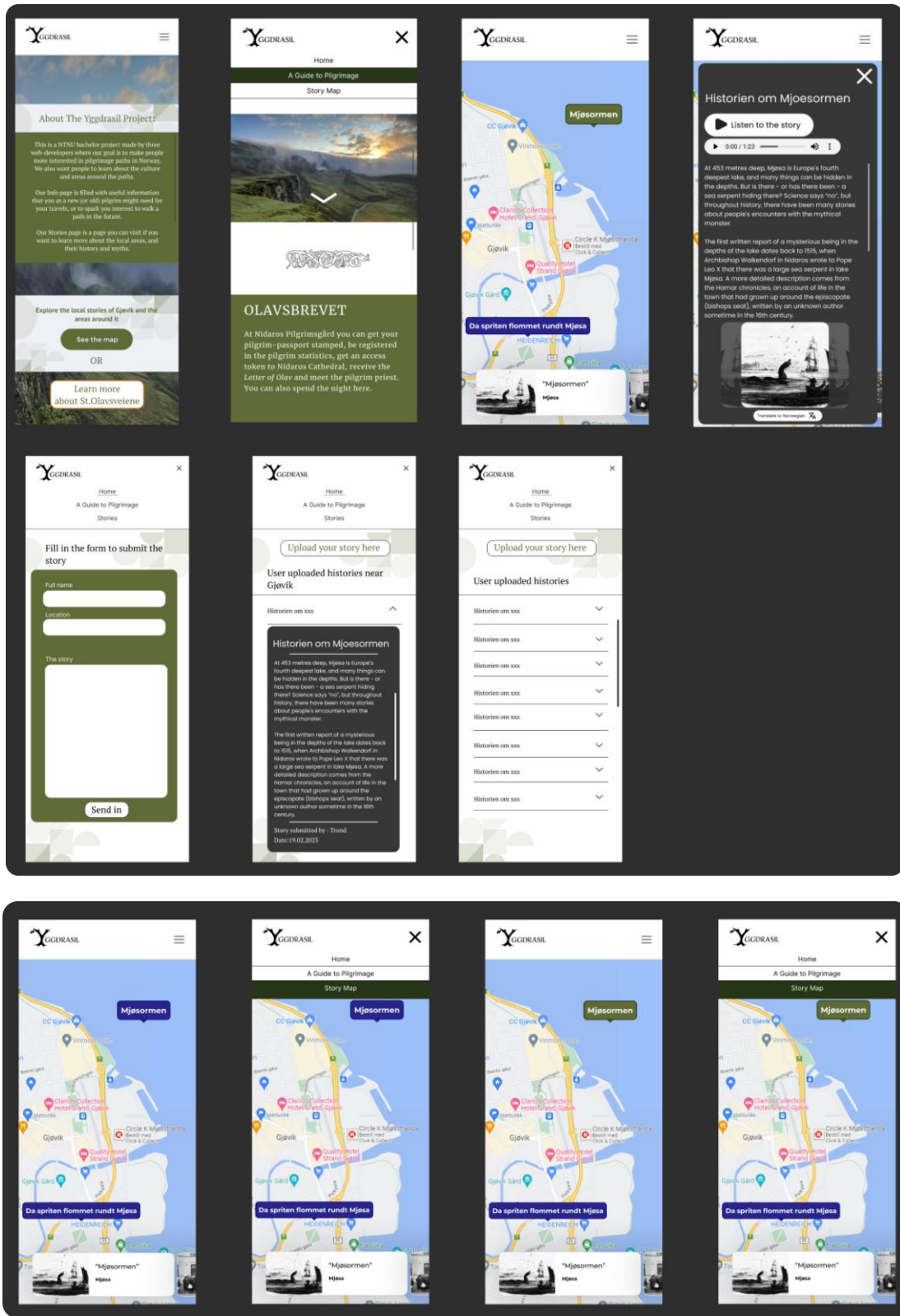


Navbar mobile

navbar-44



9.12.2 MOBILE



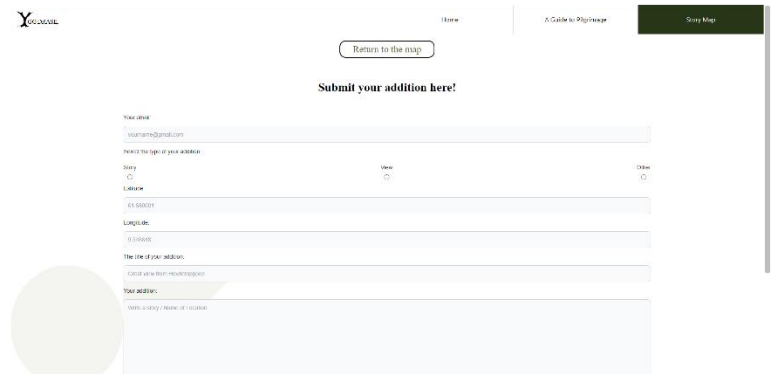
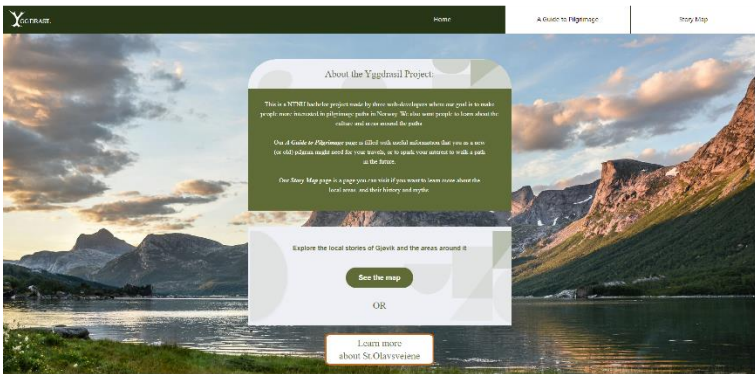
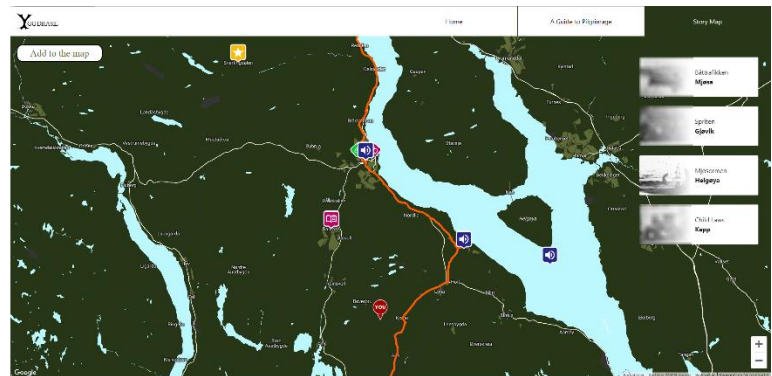
9.13 NAME SUGGESTIONS FOR THE NAV LINKS

These are names we brainstormed for our navigation links.

Suggestions "Info"	Suggestions "Explore"
Pilgrim guide	Local stories
Pilgrim Info	Explore local stories
Travel Guide	Listen to the stories
Travel Info	Learn about the stories
Pilgrim Starterkit	Stories around the ways
Good To Know	Story Map
Rookie Resources	Local tales
If You Are Interested	Tales of the area
Info about pilgrimmage	Facts and Fables
Hitchhikers guide to pilgrimmage	Wives Tales
A guide to Pilgrimmage	Urban Legends

9.14 ALL THE PAGES OF YGGDRASIL

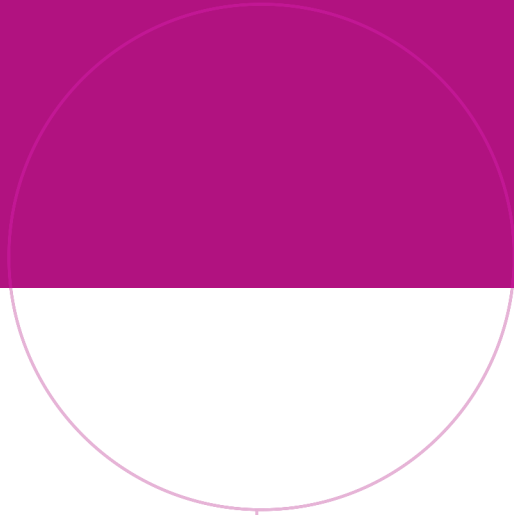
These are all our pages for our final application Yggdrasil.



9.15 LIGHTHOUSE RESULTS BEFORE AND AFTER

These are results from the Lighthouse tests before and after we did various measures to increase the values.





Norwegian University of
Science and Technology