

# **DEVELOPING A WEB PLATFORM FOR A SMALL BUSINESS**

Case: CompanyX

## Abstract

Author(s) Bravo, Juan	Type of publication Bachelor's thesis	Published Spring 2020
	Number of pages 55	
Title of publication <b>Developing a Web Platform for a Small Business</b> Case: CompanyX		
Name of Degree Bachelor of Business Administration		
Abstract <p>Service digitalization and an organization's ability to adapt to the changing pace of digital marketing can have a profound impact on a business's performance and capacity to create new relationships with their customers.</p> <p>This study describes the creation process of a web platform for a small company operating in the field of fitness and wellbeing. The web platform developed allows the company to create new value in the digital marketplace and build foundations to a continuous momentum in digital marketing and eCommerce.</p> <p>Primary data for this study was collected through interviews, workshops, and observations, while secondary data was gathered from reliable literature electronic sources and electronic documentation.</p> <p>The study contained a theoretical part focused on the defining terms of eBusiness. Technical part focused on choosing a set of technologies. As a result of the research a tangible web application was created based on the company's requirements. The application fulfilled company's requirements as well as modern web usability guidelines.</p>		
Keywords eBusiness, eCommerce, web, JavaScript, Drupal		

## CONTENTS

1	INTRODUCTION .....	4
2	RESEARCH STUDY STRUCTURE .....	6
2.1	Thesis objectives .....	6
2.2	Research questions .....	7
2.3	Limitations .....	7
3	DIGITAL SERVICES AND CUSTOMER INTERACTIONS .....	9
3.1	A word on contemporary digital business and the impact of the internet in organizations.....	9
3.2	Using a digital service to find new customers.....	11
3.3	Service evolution through the web .....	12
4	RESEARCH AND DATA ANALYSIS.....	13
4.1	Main sources of data collection .....	13
4.2	Data analysis .....	13
5	DESIGN.....	15
5.1	Choosing the appropriate architecture for the main website.....	15
5.1.1	Traditional website structure .....	15
5.1.2	Simplified structure .....	16
5.2	Website requirements .....	17
5.3	Design mock-ups .....	18
5.3.1	Frontpage and navigation bar .....	19
5.3.2	Company information.....	21
5.3.3	Opening hours and reservations .....	23
5.3.4	Services.....	24
5.3.5	Additional service page.....	24
5.3.6	Price list.....	25
5.3.7	Contact information & footer .....	26
6	ENVIRONMENTS.....	28
6.1	Version control and development environment .....	28
6.1.1	Git version control.....	28
6.1.2	Development in a virtualized environment using Windows Subsystem for Linux (WSL).....	28
6.2	Environment setup and configuration for development.....	30
7	TECHNOLOGY CHOICES FOR WEB FRONT .....	33
7.1	Choosing a web framework.....	33
7.2	Typescript.....	36
7.2.1	Styled components .....	36

8	CONTENT MANAGEMENT SYSTEM .....	38
8.1	Drupal and decoupled Drupal .....	38
8.2	Drupal JSON API.....	39
9	DEVELOPMENT OF A SINGLE PAGE APPLICATION .....	41
9.1	Project Setup .....	41
9.2	Enabling HTTP interactions .....	42
9.3	Implementing the Bootstrap framework.....	43
9.4	Code editor .....	43
10	TESTING AND FINDINGS.....	44
10.1	Testing the web application and CMS usability .....	44
10.2	Test findings .....	45
11	CONCLUSIONS .....	47
11.1	The current state of eBusiness and digital user interactions.....	47
11.2	Suggestions for further research.....	48
	REFERENCES .....	50
	APPENDICES.....	55

## FIGURES

Figure 1 Example of a content web page .....	15
Figure 2 Example of a front-page.....	16
Figure 3 Frontpage mock-up.....	21
Figure 4 Company information section mock-up .....	22
Figure 5 Opening hours and time reservation positioning mock-up.....	23
Figure 6 Services section mock-up .....	24
Figure 7 Additional service section mock-up .....	25
Figure 8 Pricelist section mock-up .....	26
Figure 9 Contact information and footer mock-up .....	27
Figure 10 Drupal page and React page loading time comparison .....	45

## 1 INTRODUCTION

The internet has brought new opportunities for small and large businesses to reach their customers and engage in new types of interactions (Cebi 2013, 1). A website can contribute to a company's marketing and business strategies by increasing visibility and giving a place where customers can find information without spending much time or effort (Cebi 2013, 2). According to Tao et al. (2013, 2), a good website has the potential to increase customer's trust in the company, and it can be substantially more cost-effective than more traditional marketing campaigns. Additionally, a website can provide a company with a gateway to engage their customers with more active methods such as service subscriptions and purchasing of products (Cebi 2013, 3).

Doing business can become faster and easier for both companies and customers thanks to the development of web commerce and websites in general. Some customers make purchasing decisions quicker when they can skim through available products and select desired items from nearly any place they wanted, without having to visit a store or making phone calls. From a company's perspective, web commerce can reduce operational costs and personnel (Weinman 2015, 95). This cost reduction can happen when a digital web store provides continuous service to customers regardless of opening times, or availability of a seller, also allowing companies operating in more niche and lower scale businesses to have a digital web store for their customers, when the sales volume would not justify renting physical premises and hiring additional personnel to accommodate transactions with customers.

These days, digital marketing platforms, web stores, and websites have come a long way from being exclusive to desktop computers and limited to a certain number of people, even though it remains limited by people with access to the internet. According to Chaffey & Smith (2017, 4), there were already over three billion customers using online platforms in 2017. Businesses have a great opportunity to make use of this scale and its benefits to grow businesses and enter new markets, not only using the internet to sell their products and services to end customers but also to create new alliances and deals with traders and suppliers from remote places that would have been unimagined without the internet as a mean of communication (Chander 2013, 1).

When designing new digital marketing platforms, we must consider the current context of digital usage and customers' interests, as well as the upcoming trends and ways of utilizing digital services. As of now, mobile devices have surpassed traditional computers as a mean to access the internet, and therefore more people are likely to reach a website, or a

web-shop, through a device other than a desktop computer or a laptop (Chaffey & Smith 2017, 5).

However, some studies reveal that most eCommerce customers are still more likely to make purchasing decisions on a desktop computer or laptop (Coleman 2017). Users make use of mobile devices to skim and browse through available products and services. Understanding this behaviour is vital to design and deliver a modern and effective digital platform. In a few words, a successful digital marketing and eCommerce platform must be easy to use on mobile devices, especially made easy to navigate and find products, services, opening hours, directions, and other information relevant to a company or organization that a customer might require on-demand, while also being able to deliver good usability on a larger device, such as a laptop or tablet, creating a deeper connection with the customers and leaving a positive image of the company or organization that entrusts customers to make purchase decisions.

This study focuses on the development of new digital commerce possibilities for CompanyX, a small business working in the field of fitness and wellbeing.

## 2 RESEARCH STUDY STRUCTURE

### 2.1 Thesis objectives

This study aims to explore the business and technical requirements of bringing the products and services of CompanyX to a digitalized platform which will enable the company to improve their service and product offering, quality, and reliability. The case looks at the implications of choosing the right tools and technologies for the company to position themselves in the digital marketplace by establishing its branding and meeting customer requirements in a customized digital platform.

The first part of this study focuses on defining the modern context of web interactions between companies and their customers. The objective is to provide a clear view of the context of modern digital marketing and eBusiness, as well as set the main foundations for the requirements to be set in the following process of design and implementation of the online solution.

The second part of the research focuses on determining the main requirements of a digital solution together with the client. The idea is to set up objectives and functionalities that we agree should be present when the solution is ready. These objectives include things such as:

- Main functionalities: What can the user do with the website and what kind of information is available to customers.
- Look and feel: How do we present the information to customers, what kind of impression the company wants to give, such as company logo, colours, and images.
- Maintenance: How do system administrators interact with the solution to keep content up to date.

The third part focuses on the project set-up from a software development standpoint. This includes finding the most suitable development environments, version control, tools to be utilized, and other commonly used paradigms related to web development, also briefly looking at the technical implementation and architectural design of the solution, observing the technologies used and design choices suitable for answering the main requirements set on the first stages of this study.

The final section consists of the validation and assertion of previously set requirements. The objective is to ensure the solution contains the main functionalities and requisites discussed with the customer in the earlier stages of this study.



## 2.2 Research questions

Within the context of contemporary eCommerce and digital marketing, and considering the use case of the proposed solution, as well as the requirements and usability features discussed with the customer, the main research question of this study is the following:

- How to enable CompanyX to create new value within the digital market?

To answer the main research question, the following questions supply additional insights towards this research:

- What new services does CompanyX want to offer to their customers?
- What type of digital solution can carry the features required by CompanyX to enable the desired digital transformation?
- Which technologies are necessary to build the digital solution for CompanyX and what does it take to build the solution?

To answer these questions the author will research through the available academic sources of the relevant topics, leverage the know-how learnt through university studies, and make use of their knowledge and professional expertise gained through real-life work experience.

This study is based on empirical findings, documentation of practical and tangible results, and validation using both pieces of literature as well as modern software development validation schemas.

## 2.3 Limitations

The main limitation of this work is that the solution is designed and tailored to a specific client and use case, which makes it unusable in other cases. However, the methodology used in this study can still be applied to other use cases as well as even to other industries.

This study is also driven by the contemporary trends in digital services and software development, and since these trends are likely to change and evolve at a considerable pace, the methodology used in this study is limited to this point of time, and it is not meant to be used as a future-proof method for these types of cases.

In the technical research of technologies and frameworks, the referenced materials and books should be considered outdated and only as a historical mention of how a number of these concepts used to function. Some of the web frameworks used in this study, such as React, change at such a fast pace that even a few years old publications may contain information that is nowadays considered deprecated or proven wrong after a short time. The recommendation for matters of real-life work is to seek information from official documentation and community postings, to get the most up-to-date guidelines and have a better understanding of where are contemporary technologies heading towards.

### 3 DIGITAL SERVICES AND CUSTOMER INTERACTIONS

#### 3.1 A word on contemporary digital business and the impact of the internet in organizations

Some studies in the early 2000s show the concern and the uncertainty of the potential impact of the internet on customer relationships. As cited by Daekwan et al. (2005, 1), the internet can potentially have a not only positive impact in the marketplace but also negative and disruptive consequences (Kettinger et al. 1994 ). Commonly used methods of customer interaction, understanding customer needs and engaging customers in service interactions and service offering improvement would need to be reinvented to become effective in the digital world and on top of that not only offer an alternative to previous methods but also offer new value in a digitalized context (Day & Bens 2005, 2).

Despite the uncertainties, it was believed that the internet would rather create new opportunities and enable organizations and companies to expand their lead and use digital services as an addition to their existing customer service and marketing networks (Day & Bens 2005, 2). Similar reinventions had appeared in the past introducing new ways of reaching people over the distance such as the radio, telephones, and television, and the internet would become one of them as personal computers with internet access became more and more common in households.

Nonetheless, as told by Day & Bens (2005, 8), internet and its capabilities are only tools and do not by themselves bring any competitive advantage or marketing benefit to a company or organization without first understanding the business context, the customers and the resources that the organization has available to invest in their digital platforming. One of the challenges that organizations had to face when entering the digital marketplace is the ability to synchronize all the different channels used for customer interactions and avoid having each channel exist as an independent department or service unaware of the other channels. The customer does not distinguish between these different channels but sees a company or a brand as one entity reachable using multiple methods (Day & Bens 2005, 5).

Most of the users of a digital service in the modern world are people who have been used to interacting with digital services and consider digital channels to be part of regular life (Perkin & Abraham 2017, 1). Digital services have been present long enough to have proven their benefits to both consumers and companies, and thus businesses not only from the technology sector are expected to engage in digital interactions to some degree.

The idea of digital transformation and digital adaptation is a concept that companies research and develop on today more than ever before. However, digital transformation is not only about a company or organization investing in their technology infrastructure, cloud computing, modern website and so forth, but it is just as important to understand the business processes, strategy and customer behaviour related to the digital marketplace and its users (Perkin & Abraham 2017, 3).

Companies and organizations must face multiple challenges to find their spot in the digital marketplace and be able to achieve positive results. Some of these challenges also reflect parts of the organization's cultural values and ideas and can lead to potential loss in the journey towards a more digitalized business model. Some of these challenges are:

- Need to update older IT systems and infrastructure that can cause difficulty in adapting newer systems and technologies (Perkin & Abraham 2017, 34). It is a common theme with medium to large companies that have adopted data digitalization earlier in the century that the growth of this data has kept the tools and platforms stuck on their original legacy systems, making their update and change to a newer infrastructure costly and time-consuming, thus keeping some organizations from updating them.
- Finding suitable processes to create a digital strategy that takes into consideration different levels of benefits and caveats, judging all dimensions that affect the business (Weinman 2015, 68) .
- Avoid focusing on the needs of digitalization in small or short-term objectives. In most cases, it would be more beneficial for the organization to plan their digitalization looking at future possible scenarios and think of the situation after the new digital offering has reached their target market (Perkin & Abraham 2017, 34). An alternative to this is for an organization to employ an agile and adaptive approach to digitalization, understanding that short-term investments may only have a small yet positive impact, and that continuous delivery of digital services may be needed to maintain a digital status (Perkin & Abraham 2017, 35).
- Cultural and behavioural structures that get in the way of digitalization, especially in some older companies or organizations founded before the rise of digital business and marketing. Some of these organizations have achieved stability and growth with their original business models and strategies and may fail to see the potential or the need to invest in digitalization (Perkin & Abraham 2017, 34).
- Limited resources to utilize in digitalization, and especially if the planned digitalization roadmap includes a larger investment than what the organization can afford.

## **Defining the start point for digitalization**

According to Perkin & Abraham (2017, 139), digitalization should first focus on the people and customers first and foremost, to understand the needs and enable the organization to react and answer with a solution that fits in the market context and answers the desires of the customers. However, McKeown & Durkin (2016, XIV) argue that the first step in digitalization is to look at the organization itself and understand its business ambition and strategies before exploring the context outside the inner structures of the organization. Both approaches can be used to determine the first steps in taking an organization into a digitalized stance, yet their differences can lead to distinct results after the implementation of digital solutions. Depending on the use case and the state of the organization's business in the planning phase, either approach can lead to a successful outcome.

In some cases, an organization is looking at improving its digital offering and updating services that customers consume using digital platforms. For these cases, a people-first approach is often used as a primary method for the planning of new digital services. With this approach, an organization can make use of their previously built digital services and produce improvements upon them by making user research and studying the feedback and needs of users. This strategy is often more cost-effective compared to rebuilding all digital services at once, making it the most used method in most organizations that seek to minimize risk (Perkin & Abraham 2017, 81).

CompanyX has had minor incursions in the digital segment, which include a website built in the early 2000s and occasional marketing campaigns as well as informational posts through social networks. The current website, however, is so outdated that I would take a larger effort to fix it rather than building a new one from scratch. Additionally, the company itself, business strategies, and company branding have changed in a way that the current website would not be able to connect to their customers.

For these reasons, the desired approach focuses on the business and company branding, with the purpose of not only improving the previous digital offering but creating new services and digital relationships with their customers.

### **3.2 Using a digital service to find new customers**

CompanyX works in the business of fitness and wellbeing. Their services are provided in-place offering gym machines, multi-sport courts, massage, solarium, refreshments, and gear reselling. This area of business can be considered to have a large user volume,

since most people engage in fitness activities and seek for these types of services, however, the geographical factor lowers the maximum number of possible customers considerably because most people would not travel long distances to acquire these services, especially considering that competitors in other geographical areas offer similar services (McKeown & Durkin 2016, 42). With these aspects taken into consideration, we can conclude that the design should take into consideration local aspects of the customers living in the Lahti area of Southern Finland, use familiar terms and visuals, and utilize this knowledge to create a branding image that relates to the customer segment.

An important aspect to be considered towards establishing new customer relationships is to optimize the website for search engine performance. The main concept to understand when looking at optimizing a website for search engine performance is that when using a search engine, we are not searching directly from “the internet”, but instead we search through the engine’s index of the internet. The search engine then can decide what to show on the search results page based on how much valuable information and keywords can find from the content of the website (Ledford 2007, 60). In this project, the focus will be on optimizing the website following Google’s guidelines for optimization.

Ideally, this new digital platform would help CompanyX to shift the business focus mainly centred in their services and products, to a more customer-centred approach where parts of the business are developed based on the customers’ needs and expectations (Fader 2012, 42).

### 3.3 Service evolution through the web

Perkin & Abraham (2017, 254) discuss the change in the scene of the world’s most valuable companies, and how these have changed constantly into technology companies, arguing that businesses from all sectors should be able to adapt to the trends of digitalization. This constant evolution in the digital market comes together with challenges to companies from all sizes, as they need to reinvent strategies and processes that may have been working just fine up to this point, but that might become a new disadvantage in the future if not adjusted to the newer norm.

Changes in the availability of digital services have also changed the way businesses compete, as well as their ability to quickly react to competition. An example of this agility and adaptation has been seen after the impact of the COVID-19 pandemic, where service companies have quickly shifted their business towards the selling of products, such as hand disinfectants, to survive the situation. This adaptation is only possible thanks to the internet and its features.

## 4 RESEARCH AND DATA ANALYSIS

### 4.1 Main sources of data collection

The design process carried out in this study is based on the results of research performed together with the client. The research seeks to understand the position of CompanyX in the digital marketplace, as well as align the results of the study together with CompanyX's vision of their digital services.

The primary data collected comes from the interview questions found in the appendices of this study, answered by the company's CEO, and discussed with other members of the management. Additionally, former field knowledge from the author was used as a consultation to complete the study. Qualitative analysis is carried out to provide better insights into this company's specific case and business challenges, as well as to generate useful data for a design starting point.

### 4.2 Data analysis

The current digital brand of CompanyX is mostly active through social media channels. The current website only allows to edit the texts shown in there, which is good for showing information such as exceptional opening hours and other special announcements. The current website does not show correctly on mobile devices, creating a risk where some new potential customers might leave the page immediately if they open it first on a mobile device. Additionally, colours and logos do not follow the company's branding.

CompanyX has seen a business impact coming from competitors, leading to the conclusion that there must be investment done in the areas where CompanyX is weaker, and as noted by the company's CEO, web services are one of them. The current website was originally purchased in the early 2000s and has not seen any significant updates since its original release.

The interview offered a good chance to also investigate the more specific features that CompanyX expects from a new website, like the ability to sell products online and get better insights on how many people use their services.

Based on the data from the interviews and observations it was possible to draw a clearer view of the company's business situation and the impact of the current digital services and branding. CompanyX has been potentially losing new customers and sales due to the outdated website. The website is only capable of delivering minimum features but does not reflect the company's branding nor its service mission. On top of that, the current website

is slow to update and does not provide an option to sell products or services directly. The main reasons behind the ageing of their website have been the lack of knowledge and awareness of the digital marketplace as a tool for marketing and sales, together with the lack of time and suitable people to lead a renovation of these services.

One key finding from these interviews is that CompanyX does not have specific personnel dedicated to marketing and/or website administration, but instead these tasks are usually taken by whoever has the time to do it, and the company is not yet thinking of hiring people for these specific tasks. This practice can lead to inconsistent marketing and most importantly means that the new website should take into consideration that the administrators are not marketing professionals.

Limited understanding of digital marketing and its impact also makes it harder for CompanyX to see the potential and justify the investment in a new digital platform, making the available resources limited and not in high priority. This led to most of the technical choices to utilize open-source software instead of purchasing other paid services.

The information provided by these interviews was important to determine the main features and requirements of the solution as well as the scope of it. Moreover, conversations with CompanyX also helped them understand better the implications of digital services and guided them towards prioritizing digitalization in their business strategies and marketing.



## 5 DESIGN

### 5.1 Choosing the appropriate architecture for the main website

#### 5.1.1 Traditional website structure

Despite the constant evolution in web technologies and hardware, most websites keep following similar design approaches when it comes to displaying content to the user, usage of links, and structure of the content distributed to multiple sub-pages that belong to the main webpage.

This structure has been in use since the early days of web usage and we still see it as of today, it is common to see hierarchical paths in websites people regularly visit, and most users would be able to recognise in Figure 1, that the current page is a sub-page of the main web address, and that by removing “info” from the address, we would get to the website’s main page as shown in Figure 2.

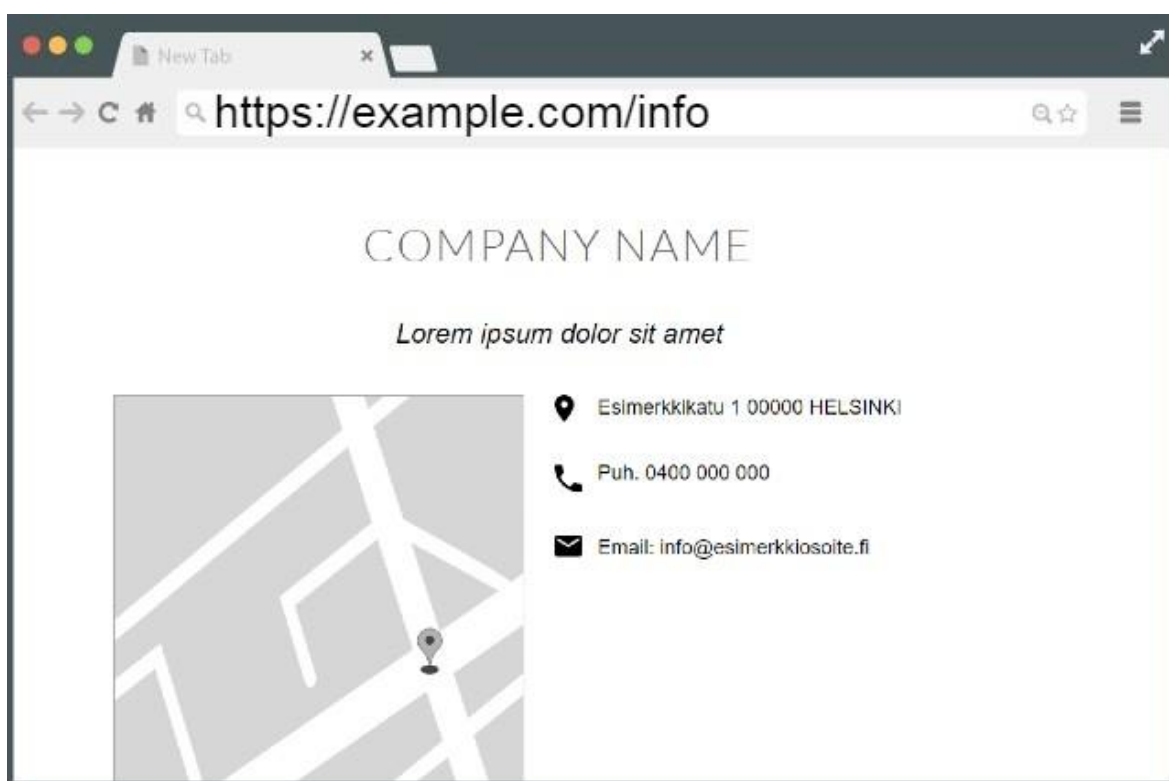


Figure 1 Example of a content web page

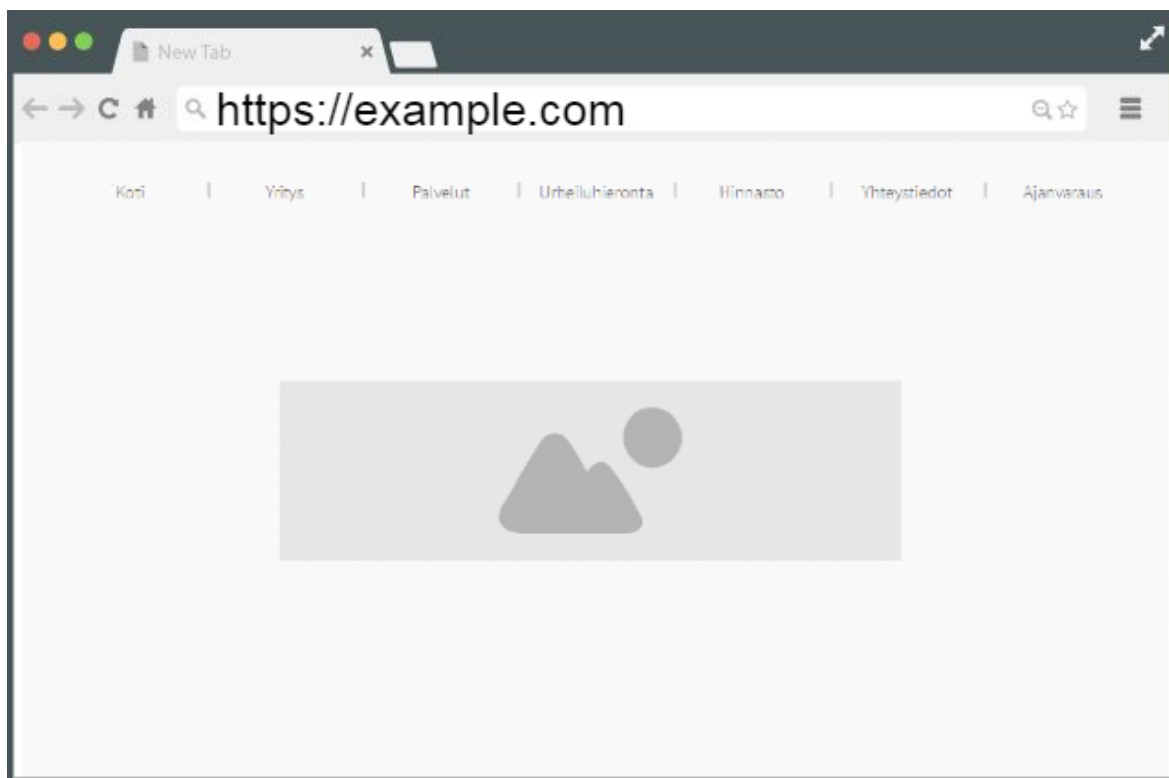


Figure 2 Example of a front-page

This design principle works well in cases where there are large amounts of content that need to be categorized. For example, on a university website, there may be multiple areas of study, each with multiple study programmes that belong to the main area. In this case, it would be most suitable to have separate pages for each study programme and build paths according to this structure having first the main address of the website, followed by the area of study and finally the study programme.

### 5.1.2 Simplified structure

A different approach to the previously discussed Multi-page application is to implement a website using a methodology called a Single Page Application (SPA). These applications allow the content of a web application to be fetched and displayed dynamically on a browser by leveraging JavaScript's functionalities.

JavaScript is capable of not only interacting with a webpage's HTML document but can also utilize the web browser's capabilities to send HTTP requests and fetch data after a page has already been loaded on the screen. Fetched data can have different types of attributes that JavaScript can use to dynamically change the content of the page event after the initial load. The JavaScript application running in the client's browser can contain more

visual elements and components than what is shown at once, and thus it can quickly react to the user's interactions with the website and modify the elements on the screen.

According to Melnik (2020), SPAs have worse search engine optimization (SEO) when compared to the more traditional multi-page application due to their dynamic nature, that is, content is not directly rendered on the request to the server, but instead the page's content is rendered by the JavaScript engine in the client's browser (Melnik 2020). Search engines' web crawlers usually index pages by saving their HTML documents directly obtained by requesting pages to their servers, in case of SPAs, server returns an empty HTML document and a JavaScript file, the latter is capable of independently requesting data from a server and building the HTML document afterwards with the collected data.

Nonetheless, as seen in recent years and in line with Gore (2018), modern web crawlers can run the JavaScript requests and execute JavaScript code, allowing them to wait for the content to be available and properly indexed. This means that a SPA can be indexed and can appear in the search results page when using a search engine (Gore 2018). However, SPAs perform worse than their multi-page application counterpart, due to the search engine having to wait for the content to become available thus making the process slower and more inefficient as opposed to the web requests delivering the complete HTML document with the webpage's content (Gore 2018).

## 5.2 Website requirements

The main requirements for the website were discussed with the client, as the main idea was to agree on specs and what things are technically possible. The main priorities for the design were:

- **Simplicity in design:** Keeping the on-screen elements to a minimum to avoid giving too much information at once, making it easier to navigate through the website, and find relevant information such as price lists, opening hours, and others.
- **Consistency in design:** Ensure that there all the colours used in the website are configured to all elements alike, making elements like texts and borders have the same colours in the whole website. Similarly, ensure that margins, paddings, font size, and other aesthetical elements have been configured consistently, for example, titles and paragraphs should have the same separation.
- **Mobile compatibility:** The design should take into consideration that a considerable number of users are going to visit the website using a mobile device (Chaffey & Smith 2017, 5). In practice, it means the design will allow elements of the screen to

shrink and/or stack on top of each other to preserve readability in a vertical orientation.

- Quick navigation: The website is designed as a Single Page Application (SPA) which means that the user only needs to wait once for the page to load its contents. The application can fetch the rest of the elements in the background and without the need for the browser to load a page, making the user experience more fluent.

In a few words, the website is a single, scrollable site with a navigation bar on top. This menu bar contains links that allow the user to navigate quickly between sections within the site.

### 5.3 Design mock-ups

The start-point for the design of the website is to look at usability, its main characteristics, and how to incorporate them into the design of the website according to the expectations of the customers and what is technically possible to build.

The main concept behind the usability of a web service is simple: As Nielsen describes (2012) when an online service is hard to use, users will likely not come back after their first visit. This on its own tells how important usability is for the survival of a web service in the digital marketplace.

This design aims to create a website that has good usability, by showing the user the right information at the right time and avoiding issues where a user might not find what they need when visiting the website. An important aspect of this matter is to identify the features that the service wants to provide to customers. Main features are:

- Display relevant information regarding the company, their services, pricing and opening hours
- Allow users to make time reservations through the website
- Allow users to use the website on different devices, especially smartphones and tablets
- Inform users of exceptional offers and opening hours

Usability will become visible when users get to use these features and evaluate how easy it was to find what they were looking for. Only after the users interact with these features, we will be able to determine if the website is useful (Nielsen 2012).

To work on improving usability, one of the main actions would be to study the users and their behaviour with the actual platform in use. However, in this research, this method is out of scope since the final solution is not yet being used by end-users. Regardless, there are other methods through which it is possible to get a better idea of a usable website, like looking at other websites from competitors and assessing their usability, taking the good and the bad as examples of what should and should not be done. Besides, sticking to a well-established frontend style framework can set the basis for a responsive and accessible website layout.

Once the main design concepts have been discussed with the client, the next step is to create mock-ups that will visually represent the layout of the website. A mock-up will contain the positioning of buttons, text, images, and other elements present in the website. The idea is to give the client a hint on how the final website will look and feel and have the opportunity to make adjustments before larger work is done.

### 5.3.1 Frontpage and navigation bar

The front page of a website works as the main point of entry for most users (Nielsen 2002). Unlike a traditional website, where every internal link goes to a page within the website, a SPA brings the whole site into one scrollable piece of content.

#### **Navigation bar**

One of the first concerns with this approach was about how to avoid the user from getting lost within the content of the page while scrolling down and needing to go back to another section. To solve this query, a study was executed to choose the most suitable solution for this use case.

The Bootstrap framework (see chapter 9) has a set of examples of different visual components and UI elements that can be used when building an application with that framework. It also contains a set of navigation bar examples that were used to measure suitability with this use case (some of the examples were edited and more content was added, to allow scrolling)

- Navbars example: It contains several examples of a navigation bar, each with different responsiveness settings. These demonstrate on which screen size the navigation bar changes from desktop to mobile styling. Bootstrap's default responsive navigation bar properties were chosen since there are no specific reasons to choose a smaller or a larger pixel count for the responsive breakpoints

- **Navbar static:** In this example, the navigation bar stays at the very top of the application and it disappears as the user scrolls down the window. This example got criticism because it would create situations where a user would need to scroll back up whenever wanting to use the navigation bar. The suggestion also considered the usage of a separate “scroll back to top” button, which would allow the user to get back to the navigation quicker by only clicking on that button. The idea was still dismissed as it seemed like a solution to a problem that should not be there in the first place.
- **Navbar fixed:** This example shows a navigation bar that is fixed to the top of the viewport, as opposed to the top of the HTML document. This property allows the navigation bar to stay visible even when the user scrolls down through the website. This alternative was selected finally because it allows the page to grow larger with content in a vertical orientation, but it makes it possible to navigate quickly to other parts of the site no matter where the user is at.

### **Design philosophy**

A Flat Design consists mostly of minimizing text and images down to only the most important elements while keeping colours simple and consistent. This means avoiding cluttering the website with excessive use of shadows, gradients, animations, and other distracting elements, keeping the user’s attention on the relevant pieces of information found on the website (Pratas 2014, 20).

One of the most requested features was to have a large background picture immediately on the front page of the site together with the company logo. Large pictures can have sometimes a negative impact on websites: They can have large file sizes, slowing down the performance of the website overall, and they can be too invasive when trying to have a flatter design overall.

The design mock-up shown in Figure 3 aims to keep the design of the front page simple and flat while bringing the company logo to the front and allowing a background picture to be placed and cover the whole background. The background is set to have a grey overlay, which flattens the colours in the picture itself and allows the text areas to preserve contrast and remain visible and readable (Pratas 2014, 21).

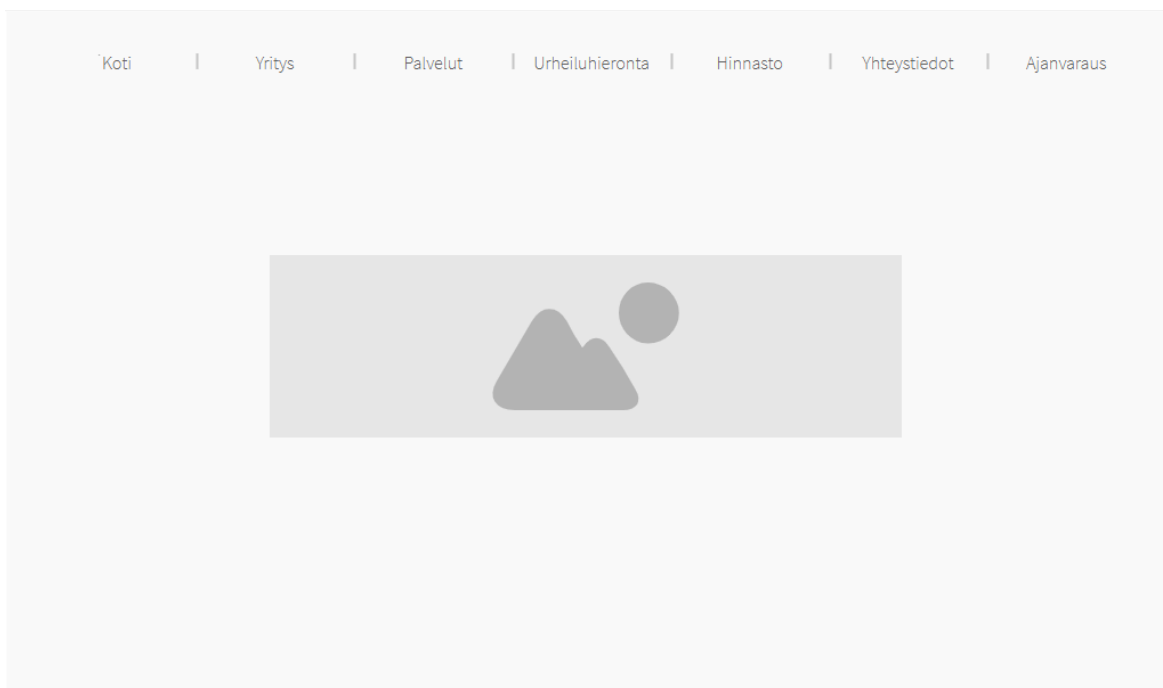


Figure 3 Frontpage mock-up

### 5.3.2 Company information

This section contains the most content-rich part of the website. It consists of a large heading, a subheading, a paragraph, and a split image-paragraph section. Additionally, the design also includes a social media widget, placed on the right side below the second paragraph (see Figure 4).

This section is shown immediately after scrolling down from the front page, and its purpose is to give information on the company and its services cleanly and concisely, trying to keep text to a minimum, so that all of the section's contents fit in one large or medium screen, and keeps scrolling to a minimum in smaller screens.

The main heading of this section brings the attention of the user to the most relevant keywords related to the company and its services, while also being the main heading for search engines.

The subheading works as a place for short company's branding, such as a slogan or similar phrase meant to give a more personal touch to the reader.

The following paragraph is meant to carry a motivational and invitational message to the reader, encouraging them to engage with the company and its services. Hence the positioning between the main heading portion and the rest of the section.

The split view section (see Figure 4), which contains pictures, text paragraphs, and a widget. On the left side, the images are small and show the company's premises from the outside. The first paragraph on the right contains more specific information on the company, who are they, what they do, and together with the picture on the side, the objective is to deliver a more intimate idea of the company, showing its core values and a few mentions on their services.

Another requested feature was to have live social media updates directly on the website. This way special promotions and announcements can be published through the social media service Facebook, and they would become immediately visible on the website. The Facebook widget is placed directly underneath the company information text.



Figure 4 Company information section mock-up



### 5.3.3 Opening hours and reservations

This section uses a more traditional table for organising information displayed on the screen. The objective is to make opening hours easy to find and readable, having minimum clutter on screen and ensuring texts and contrasts are optimal.

The bottom part after the table (see Figure 5) contains an embedded document referring to an external service. The company uses a third-party application to allow the customer to make time reservations online. As discussed with the client, it would be ideal to allow customers to purchase time slots directly in the main website without the need of following a link to an external website. The solution to this problem was to embed the time reservation application in a container inside the main webpage.

#### AUKIOLOAJAT

Vastaanotto ja kahvio:	Kulkukortilla salille (Sisään viimeistään 1h ennen sulkemisaikaa)
<b>Ma - To</b> 16.00 - 20.00	<b>Ma - Pe</b> 05.00 - 22.00
<b>Pe ja Su</b> 14.00 - 19.00	<b>La - Su</b> 08.00 - 20.00
La suljettu	

#### AJANVARAUS

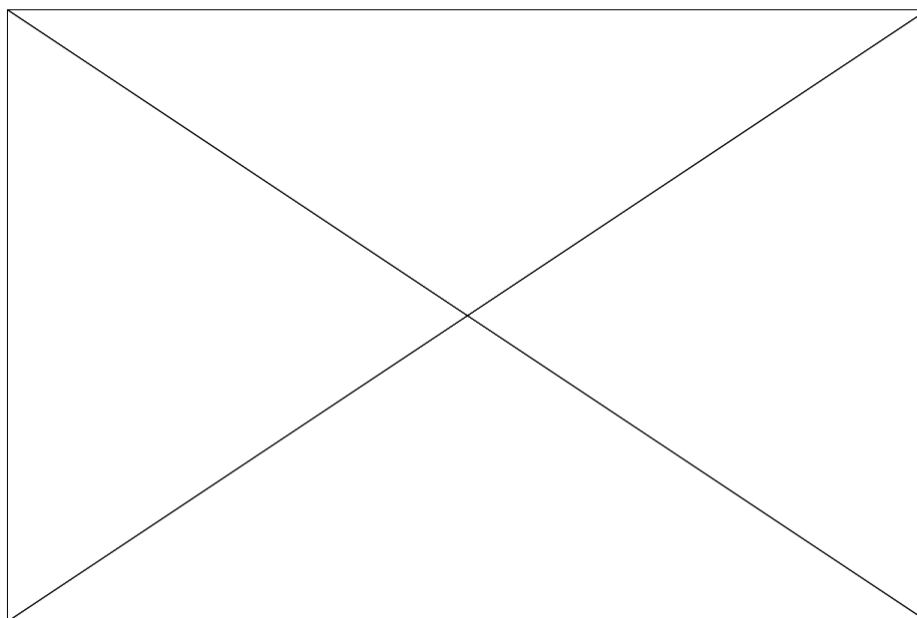
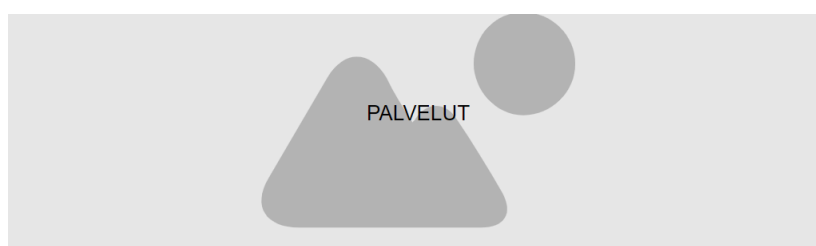


Figure 5 Opening hours and time reservation positioning mock-up

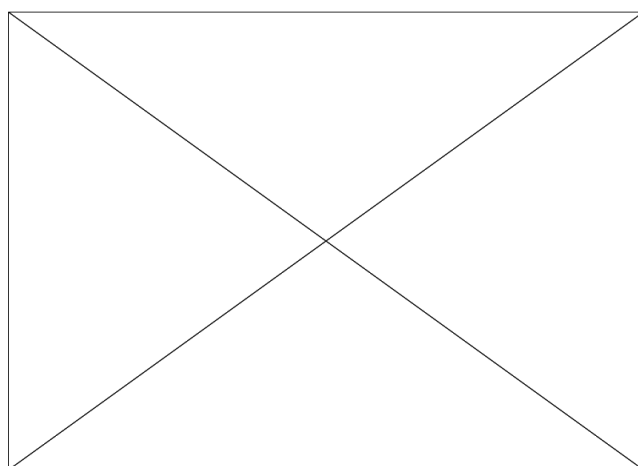
### 5.3.4 Services

This section has a simple layout containing a large title with a background picture, and a set of a title, paragraph, and picture for each service needed to be displayed. The main idea of this section is to keep texts to a minimum and choose the pictures carefully to show the premises available for use. This design choice is to allow users to quickly scroll through the website and get most of the necessary information visually from the pictures, so users do not need to spend much time reading. Paragraphs should contain only the most critical information and keywords used mostly for accessibility and search engine optimization.



KUNTOSALIT

*Lorem ipsum*



MAILAPELIT

Figure 6 Services section mock-up

### 5.3.5 Additional service page

This section re-utilizes the layout shown in Figure 4 Company information section mock-up, apart from adding a title with a background picture on the upper part. This additional service needs a larger information block since it is directed to a more specific group of

customers, and some users might come to this page for the only purpose of reading this section.



Figure 7 Additional service section mock-up

### 5.3.6 Price list

As shown in Figure 8, this section reuses the element with a title and a background picture (used also as a separator between sections when scrolling through the website). The information in this section should be clean and easy to read while retaining its layout

within the tables. On a large screen, the tables are shown as in figure 4 side-by-side, whereas in a smaller device the tables shift places and stack on top of one another.



## HINNASTO

*Lorem ipsum dolor sit amet*

KUNTOSALI	Hieronta
Kulkukortti 5€	30 min 25€
Kertamaksu 7€ - opiskelija / eläkeläinen 5€	45 min 35€
Kuukausikortti 45€ - opiskelija / eläkeläinen 40€	60 min 40€
3 kk-kortti 120€ - opiskelija / eläkeläinen 110€	90 min 55€
6 kk-kortti 210€ - opiskelija / eläkeläinen 190€	Squash / pingis / mini Sulkis
12 kk-kortti 350€ - opiskelija / eläkeläinen 315	30min / 8 € / kenttä
10x kortti (voimassa 6kk) 50€ - opiskelija / eläkeläinen 45€	60min tai yli 15 € / h / kenttä
30x kortti (voimassa 6kk) 140€ - opiskelija / eläkeläinen 125€	SOLARIUM (K18)
50x kortti (voimassa 6kk) 225€ - opiskelija / eläkeläinen 205€	Kertamaksu 5 €
Arkiaamukortti (sisään ma-pe klo 5-14, 1kk tai 10x) 30€	

Figure 8 Pricelist section mock-up

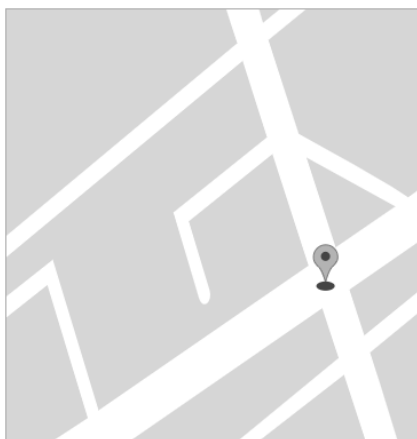
### 5.3.7 Contact information & footer


Finally, this section contains the company's contact information together with an embedded map element showing the real address. Elements are set to change form a side-by-side alignment to a stack if viewed on a small screen.



## COMPANY NAME

*Lorem ipsum dolor sit amet*



 Esimerkkikatu 1 00000 HELSINKI

 Puh. 0400 000 000

 Email: info@esimerkkiosoite.fi

---

Ylös



Figure 9 Contact information and footer mock-up

## 6 ENVIRONMENTS

### 6.1 Version control and development environment

#### 6.1.1 Git version control

A great challenge in software development is how to keep track of changes made to source code and how to pack these changes into versions, especially when a codebase is being simultaneously edited by teams with many people. To account for this problem in software development a plethora of different tools has been developed throughout the years, one of them named Git, which is used as part of this research.

Git is a version control system that allows teams to work in software development simultaneously while being able to record changes made together by the team (Narebski 2016, 2). Each developer gets to work on their isolated code base (called a branch), while Git takes care of looking for the differences between the developer's codebase and the original codebase (traditionally called a master branch). Git then takes the changes made in a developer's branch and merges them to the master branch creating what is called a commit: A set of changes made to the master branch that contains information about what was changed, when and by whom (Narebski 2016, 27).

Git will also keep a history of all the changes made to the code, making it possible to revisit and reset the codebase to earlier points in the application's history, so that it is easier to isolate changes that might have caused problems later in the application's lifecycle.

Git is used in this study for managing both the Drupal codebase and the React application.

#### 6.1.2 Development in a virtualized environment using Windows Subsystem for Linux (WSL)

Computers are built on hardware and software. Most computers programs are designed to use specific instruction sets determined by the combination of system interfaces and I/O provided by the hardware in the computer, that is, software and hardware need to be compatible with one another for a computer to function as desired (Smith & Nair 2005, 8).

This combination of hardware and software allows programs to run in an operating system by utilizing the available hardware capabilities such as memory, inputs, and outputs (Smith & Nair 2005, 9).

Traditionally, programs would run directly on the computer's hardware and operating system, however, this came with some problems: Developing new software involves testing and running software that is not yet ready, meaning that it can perform poorly, have issues where it freezes or utilizes system resources excessively and can cause malfunction not only within the program itself but also affect other software running on the same hardware.

One of the solutions created to this problem is the usage of Virtual Machines (VM): A VM is a virtualized computer that can run an operating system and any compatible programs in an isolated environment (VMware 2020). Virtual Machines allow a single set of physical hardware to run different operating systems and otherwise incompatible applications while keeping the host system clean and safe because a VM's filesystem and applications do not have access to the host system's resources directly (IBM 2019).

Virtual Machines have been used for software development for many years, as they provide a suitable environment where developers can test and try unfinished software without the risk of breaking the complete system (Smith & Nair 2005, XI). Virtual Machines often allow the host system to take "snapshots" of the state of a VM, in other words, the host machine can save a VM's state in a file that can be restored at any point in the future, allowing developers to rewind their development VM to an earlier working state if the virtualized environment fails or becomes unresponsive (IBM 2019).

Virtual Machines and their advantages can also be leveraged in production environments, where scalability and flexibility are important. Since VMs can be contained and copied as files in a host system, they can also be quickly duplicated, allowing services to be deployed multiple times to balance loads if the application running in the VM needs to handle a larger number of requests (IBM 2019).

### **Lightweight virtualization**

Some of the downsides of a traditional VM is that it can never fully take advantage of the hardware the host machine runs on, meaning that performance is affected negatively and running multiple VMs in one single host can cause instability and low performance (VMware 2020) since the host machine needs to allocate part of its physical memory and processing threads to the virtualized environment. Many of the use cases for Virtual Machines include running programs in different operating systems without the need of having a separate physical machine with a different operating system in it, however, this comes with a large overhead of installing a whole operating system in a virtualized environment.

To provide an alternative to some of these problems, different virtualization techniques have been developed to allow running, for example, Linux application in a Windows

machine, without the need of virtualizing a complete system. These applications usually create a virtual filesystem where Linux applications can run and have a translation layer between the virtualized environment and the host machine, allowing instructions from a Linux environment to be executed in a Windows environment (Lewis et al. 2028, 1).

In 2017, Microsoft released the first version of Windows Subsystem for Linux, a tool for running Linux programs natively in Windows 10 without the need for virtualization of the system's hardware (Lewis et al. 2018, 1). Additionally, Microsoft provided a list of supported Linux distributions that can be installed in the host machine with little effort, bringing a functional Linux environment to the Windows ecosystem (Lewis et al. 2018, 2).

From a software development standpoint, choosing the right operating system to develop applications is an important matter and sometimes even critical. In modern organizations where agile methods are utilized for software development, the developer not only writes code, but it is also accountable for testing and fixing issues, as opposed to the earlier methodologies where coding and testing were done by separate teams.

In web development, most of the applications that a programmer works on are likely going to be running in a Linux server, making the choice for a development environment immediately look for a platform that resembles the one where the final application is intended to be executed. Linux servers are found more frequently because they are free and faster to set-up in comparison to their Windows counterparts. Windows Subsystem for Linux provides a way to test and develop applications in Linux while still using Windows as a main operating system.

This research utilizes WSL to develop and test all the pieces of the final solution, taking into account the environment that it will eventually run in and the possibilities of developing from a Windows machine, allowing for more end-to-end testing considering that includes both systems and servers in Linux environment, and front end testing in Windows, which is the operating system most users are likely going to be accessing the application with.

## 6.2 Environment setup and configuration for development

Most of the required software for the development phase can be installed and configured in WSL without much difference compared to a native Linux system. The requirements for this use case can be divided into two main sections: The part of the application containing the CMS and database (which in this specific case, also includes the web shop's front end), and the front end for the main website.



The main part consists of a Drupal instance, which requires the installation of a tech stack known as LAMP (Drupal 2020a). LAMP stands for Linux Apache MySQL PHP: Linux is the operating system that contains all the rest of the components, Apache is the web-server used to listen to HTTP requests and serve requested files over the network, MySQL is the relational database used to contain the application's data, and PHP will allow the system to execute PHP code and integrate it with the web server, making it possible for PHP to render HTML documents served by the webserver upon HTTP requests. In this setup, we are using the most common technology stack to host the Drupal instance, even though some of the components of the stack could be replaced by similar software: We use Apache since Drupal automatically generates .htaccess files with correct configurations, also we stick to MySQL database to avoid incompatibilities between third-party modules and database server (Drupal 2020a).

The second part of the solution has a much smaller set of requirements since the main website is a SPA built-in JavaScript, the only requirement is to have Node installed in the system: Similar to PHP, Node allows the computer to execute and run programs written in JavaScript. Node comes with an important piece of software called npm (Node Package Manager) which provides tools for JavaScript development. When building in the JavaScript ecosystem, it is recommended to use open source libraries and modules for desired features and functionalities, this way we avoid re-writing things that have already been done and that are maintained by someone else. Npm can download and install the desired libraries in the project directories and keeps track of all the installed packages and their versions. One of the benefits of this is that downloaded libraries do not need to be committed in version control, but instead we commit only the npm generated file containing the information of the required packages and their versions, making version control substantially lighter.

Npm can also execute the code in the JavaScript project using Node instances in the background, so the application does not need to use a separate web server for development, instead, a Node server is spawned ad-hoc which is capable of serving the webpage in a browser and can also reload the page every time a file is saved in the project's directories, making it easy to see changes quickly in the browser without having to manually refresh the window.

All these components and applications can be installed in WSL without the need of installing anything extra in the host system and still being able to do all the front end testing in the host machine since Apache and Node servers can listen and respond to requests coming from the host machine. This essentially means that Drupal and the JavaScript

applications running in the Linux system can be displayed using a regular web browser in the Windows system. Moreover, the WSL integration with Windows 10 allows the developer to use Visual Studio Code running in the host machine to edit code directly in the WSL filesystem, making development faster and more efficient.

## 7 TECHNOLOGY CHOICES FOR WEB FRONT

### 7.1 Choosing a web framework

In modern web development, it is common to use tools and functionalities from the open-source spectrum. One of the reasons why software development, in general, evolves quickly is that many functionalities have been already created and there is no need to re-invent these when needed. Libraries and extensions are available to many programming languages and include features commonly found in most applications, for example, if an application needs to have a way to manage date data, then there can most likely be a library or module with a set of features used to do exactly that. Open-source software is not only available for any intended use but can also be modified and adapted to any use case (Deek & McHugh 2008, 1). This leads to newer developments and improvements upon the previous versions.

Not only specific functionalities are found in the open-source community, but also entire frameworks designed to offer alternatives for the development of a specific type of software without having to build everything from scratch. As such, single-page applications and other types of web applications have seen development through the years and multiple open-source frameworks have been created, each having a specific set of features as well as disadvantages.

In this study, we will look at two of the most popular frameworks for web development: React and Vue.

#### **Comparison between frameworks**

React and Vue have some similarities, especially in the way they process information and interact with the user interface shown on the screen. Both frameworks use what is called a Virtual DOM: Essentially a copy of all the elements present on the screen, together with metadata and other elements not visible but meant for computer programs such as search engines. The running engines underneath React and Vue allow the applications to change only the parts of the DOM that need to be updated (Vipul & Snopatki 2016, 2). Based on the user interactions with the application and automated actions, the application can make dynamic changes and thus decrease the need to reload the page multiple times (Vipul & Snopatki 2016, 1). Additionally, both libraries allow the application to reuse components across the application, reducing the need for writing extra code for similar UI elements (Nowak 2019).

Vue comes with a set of additional features designed for state management and routing, which React, on the other hand, does not and instead relies on the efforts of the open-source community to address these needs (Vue.js 2020). The set of requirements for the website being built in this research is small enough not to require a complex state management system in the application, thus making React a better choice since it does not come with unnecessary features.

Vue's templating approach to writing user interfaces allow a developer to write JavaScript, HTML and CSS code all in the same file, much like in more traditional web development, but with each section being more separated from one another and allowing data binding between the HTML elements and the JavaScript code. React, as opposed, utilizes a different syntax called JSX (discussed further in the next section). Despite JSX having a slightly steeper learning curve (Nowak 2019) compared to the more native Vue approach, the author's previous React background makes JSX syntax a more sensible option for this project, allowing for faster development.

Finally, Vue's integration with Typescript is not yet fully worked out (Vue.js 2020). In comparison, React has robust support for Typescript, making development less prone to errors and faster overall (see sub-chapter 7.2).

React was chosen as a web framework for this project, as in this case, it will allow faster development and potential to be easily expanded in the future.

### **Usage of the React framework**

The frontend of this website is built using React, a declarative JavaScript library created by Facebook and maintained by both Facebook and the developer community (Hunt et al. 2016). Since its appearance, React has been constantly extended and received several new features that allow React developers to take advantage of a declarative programming paradigm. According to Bertoli (2017, 9), a declarative programming approach uses less time programming specific tasks for the computer to execute, and instead, it simply asks for things to occur. This is possible in React because the logic behind many of its functionalities is already defined within the code of the library, and the library offers directly importable modules containing functions that can be utilized by the developer (Bertoli 2017, 49). In addition to the React library enabling declarative programming, modern JavaScript also contains a plethora of functions and components that help to write less code and create features quicker than compared to previous practices. A good example of JavaScript enabling declarative patterns is the use of the `Array.prototype.map()` function to iterate through arrays, replacing previous patterns using manual loops for each element inside an array.

These features made React a very compelling choice for the research in this document. The most important factors that made this JavaScript framework a good choice for this use case are the following:

- Declarative paradigm: Build features with less code, making use of the built-in for managing state and component updates.
- Usage of virtual DOM: Allows the single-page application to run faster and create a richer user experience avoiding page reloads.
- Re-usable components: Specific pieces of the design layout for the website utilize the same elements. With React, it is possible to define these pieces only once, and re-use these components as many times as needed (Horton & Vice 2016, 19).
- Compatibility with third-party libraries and modules: Many functionalities do not have to be re-invented for this solution. Using ready modules from the JavaScript ecosystem saves time and makes the code more reliable overall.

Understanding of the HTML syntax is essential to be able to build web user interfaces. The JavaScript language, however, does not have a native method for generating more complex HTML documents and most importantly a way to visualize the hierarchical structure of the resulting HTML document (Fedosejev & Bush 2015, 28), but instead, it can create individual elements and manipulate existing DOM elements.

To bring an alternative to this, React introduced a syntax called JavaScript Extensible Markup Language, most often referred to as JavaScript XML (since the XML acronym has been in use before) or simply JSX, which is also the file extension used for files containing this syntax. This syntax allows HTML code to be written inside a JavaScript file allowing the developer to visually examine the structure of the HTML document that gets generated when the React engine renders a page.

Furthermore, React offers an extended set of features and integration with the JSX syntax which allows pieces of HTML code, like a container element with text and pictures inside, to be saved in a JavaScript variable. This variable, containing the HTML document inside, can then be manipulated as any other variable in JavaScript, it is possible to, for example, pass a variable containing HTML to a function that may or may not return the HTML depending on other conditions.

## 7.2 Typescript

JavaScript works as an interpreted language, executed by the browser's JavaScript engine that converts the JavaScript code into instructions for the computer (Nance 2014, 21). This aspect of JavaScript makes its development lighter compared to other compiled languages such as C#, which needs to be compiled into an executable file to function. The usage of JavaScript has grown larger with time and together the need of creating larger applications has also grown greater, thus JavaScript's weaknesses have also become clearer and have motivated developers to find solutions to these disadvantages.

Typescript tries to solve some of these problems by adding a compilation layer to JavaScript that brings type checking and better error reporting, also

Typescript is a compiled language based on JavaScript which adds features absent in the JavaScript language. Unlike JavaScript, Typescript is not executed by the browser or Node, but it is always compiled before execution (Nance 2014, 22). Typescript allows the usage of static typing, classes, interfaces, generics, and modules, which native JavaScript does not have

Some of the advantages of Typescript's static typing are:

- More concise code
- Code editors can pre-compile code and highlight typing errors as the developer writes code.
- Public and private members can be used when writing Typescript, adding some Object-Oriented paradigms to JavaScript that were otherwise impossible.

Typescript also comes with some downsides, since Typescript project requires extra steps for setting up compilers, and deeper knowledge of JavaScript specifications to get the compiler to compile code into the desired compatibility mode. Furthermore, the compiled code is usually optimized for machine-readability, making it substantially harder to debug compiled Typescript code.

### 7.2.1 Styled components

This library adds the ability to load only the necessary CSS styles when loading a page, since styles are not written separately in a traditional CSS file, but instead they are appended to each React component and thus loaded only when a specific component is loaded (Styled Components 2020).

Another benefit of the Styled Components library is that code maintenance is also simplified, as the developer only needs to look for the desired React component declaration and can edit the CSS styles directly, without having to skim through large CSS files searching for specific classes (Styled Components 2020).

Lastly, this library also takes care of assigning classes to HTML elements automatically, which means that the developer does not have to think about new names and ensure they are unique, to avoid applying styles to undesired elements (Styled Components 2020).

This library is used in this application for styling specific elements and their properties such as colour and size. In addition, styling for larger areas and layouts are built using a separate CSS framework (See chapter 9.3).

## 8 CONTENT MANAGEMENT SYSTEM

### 8.1 Drupal and decoupled Drupal

Drupal is a web application whose main purpose is to allow site administrators to manage the content of a website dynamically, allow content to be changed and published quickly, and without having to edit code directly. Drupal is built following the MVC principle of separation of concerns, where there is a separate layer responsible for user interaction, logical operations, and database interactions (Abbott & Richard 2016, 4).

Part of the solution presented in this research is a webshop developed earlier by the author using Drupal. One of the motivations behind the idea of using a SPA was to make its content also be editable, but without the large overhead of creating an administration tool from scratch.

Considering that a Drupal instance is highly expandable and modular, it was decided to leverage Drupal's content management functionalities and use them on the website's main page, but without using the more outdated and unsuitable native front end.

#### **How to consume content from a Drupal instance with a separate React application?**

The main challenge in this scenario is to connect the Drupal instance with the separate React application to allow content from the Drupal database to be fetched and displayed in an independent application. A solution to this problem is to make use of HTTP requests to transfer data over the internet, from the Drupal end to the website front. This process will be explained in more detail through the next sub-chapter.

The term "decoupled" comes from the idea that a Drupal-based web service does not use Drupal's built-in presentation layer, but instead Drupal turns into a data management tool and the content itself is displayed to the end-user by a separate application. The Drupal instance used in this research uses a mix of both ideas, hosting a webshop and displaying its content using Drupal's presentation layer, while the rest of the content corresponds to the main SPA used for the front page of the company. This approach allows site administrators to manage the content of both sites from a single system.

Drupal is built on a monolithic architecture capable of controlling all parts of the application, from the data layer to the presentation layer (Buytaert 2019). Decoupling the presentation layer has some trade-offs:

- Content preview: Drupal offers the ability to preview new pages before publishing them (Buytaert 2019).



- In-place editing: Edit blocks directly on the page when logged in.
- Layout management: Change the order of some of the elements on the screen.

Alternatively, custom modules could be built to bring some of these functionalities. It was decided, however, that they do not bring enough value to justify the investment of building such modules.

## 8.2 Drupal JSON API

Drupal is completely open-source software, which means its codebase is free and available for anyone to use free of any costs. This also allows developers to create their modules and extensions that bring new functionalities for more specific use cases to the base application that otherwise would not be there. One of these modules is called JSON API.

This module scans the content structure in Drupal and exposes a web endpoint through which external services can request data content from Drupal directly and get responses in JSON format containing different data from the elements in the content structure such as titles, texts, and configurations. The module implements the API following the standard of the JSON API specification and has been included as one of the modules that come by default together with the basic installation of Drupal 8.

### **Security aspects of the JSON API module**

Drupal has a large set of internal built-in security features for controlling multiple parts of the application. In this case, the focus I set on the security considerations of exposing Drupal content in JSON format.

Drupal's internal Entity API manages content and access to its parts. Content in Drupal is separated into the following parts:

- Fields: These can be any type of input that allows a system administrator or content editor to enter any type of saveable data into the system (Glaman 2016, 154). A field can be a text input, a date input, an image upload button, or an audio record button.
- Content types: These are collections of fields with a specific order. Usually, on a website there are separate pages containing different content in them e.g. a front-page may contain a background image with a large title, whereas a contact page may contain a few smaller titles and their respective paragraphs (Glaman 2016, 36). Content types can be used to create these distinct structures as a mould for constructing pages with the same content layout (Abbott & Richard 2016, 54).

- Nodes: When using a content type to create a new page, Drupal creates a new node in the database which contains the actual content saved using the content type (Glaman 2016, 28). Nodes also receive additional data like date of creation, the user who created it, and a unique URL where the saved node can be accessed.

Drupal uses a modular security system, which allows the site administrators to assign access permissions to each component in the system, including fields, content types, and nodes. These access permissions are given to users based on their role (Abbott & Richard 2016, 38). All interactions with Drupal are considered to come from a user, even if this is not necessarily logged in to the system, they are considered anonymous users and Drupal has a specific security role for anonymous users (Abbott & Richard 2016, 38).

An anonymous user can be a visitor to a page or a program that accesses the data like a screen reader or a web crawler from a search engine. When creating content types and nodes in Drupal, the site administrator defines whether these nodes are available to anonymous users, and thus when new nodes are saved any user can access these new nodes (Abbott & Richard 2016, 38). Content exposed through the JSON API module follows these same rules: The endpoints to fetch data are opened to anyone, just like the website itself, but only allowed content can be retrieved from an endpoint (Drupal 2020b).

JSON API not only allows content to be fetched from the Drupal's content database but also includes methods for creating new resources, removing resources, updating resources, and uploading files to the server's filesystem. For this use case, it is not necessary to allow any of these additional methods since we only want to consume the data and not modify it. For this purpose, the JSON API module can be set to read-only mode, where only read operations are allowed by the module (Drupal 2020b).

## 9 DEVELOPMENT OF A SINGLE PAGE APPLICATION

### 9.1 Project Setup

To start to build this application, we will be utilizing a tool called Create React App.

Create React App is a standardized React application skeleton generator, currently maintained by Facebook (Also creator of the React library originally) and available to anyone as an open-source project. This tool essentially contains an executable set of computer instructions that perform the following operations:

- Creates a directory with the given project name and inside it creates a basic folder structure with a public directory and a source directory
- Creates a readme file containing instructions on how to execute the included scripts
- Initializes an npm project inside the main directory containing information on the project and its dependencies
- Creates a gitignore file containing the most common file and directory exclusions in version control. This makes it easy to initialize a new repository in the main directory and commit the project to a remote repository.
- Creates a set of sample files that together show a basic React application in the browser when executed

Additionally, Create React App also contains additional parameters that can be used to generate a project with more specific configurations, one of them is used to configure the project to use Typescript instead of simple JavaScript, creating a tsconfig file in the project's main directory with instructions for the Typescript compiler. Some of the values in the tsconfig file cannot be modified since they are set to be compatible with the rest of the Create React App configuration parameters, taking some of the flexibility away but ensuring a functional application when running the built-in scripts.

The resulting project after running the Create React App scripts has a functional project template for creating a Single Page Application with React and Typescript which is compatible extendable using the modules and packages found in the JavaScript ecosystem. Additional packages are needed for the project to get all the functionalities required for the SPA to work and be able to connect to the Drupal instance.

## 9.2 Enabling HTTP interactions

JavaScript has built-in functionalities for sending HTTP requests to remote servers over the internet. These requests allow the application to fetch content and other data after a page has already been loaded, enabling the application to change its content without the need of reloading the whole page when navigating through the website. For this project, an additional JavaScript module called Axios is used to extend the native HTTP functionalities and provide easier handling of JSON objects. The Drupal API delivers content in JSON format. Since JSON is built on the JavaScript language it can be quickly parsed, and its context utilized in rendering the webpage elements. Additionally, AXIOS provides built-in functionalities that parse JSON objects automatically when processing HTTP responses, taking one additional step out from the development process.

Installing Axios is done using the npm command “install” which connects to the npm repository and downloads the latest version of the module and installs it into a folder called “node\_modules” which contains all the code corresponding to the third-party installed packages and modules. After the installation script is done, Axios’ functions are available for use in the project’s Typescript files.

Axios requires specific parameters to connect to the right server and fetch the necessary data, these parameters are passed to the Axios HTTP request functions before the requests are executed. This, however, can lead to code-repetition if similar requests are executed in different parts of the application, for example, it is necessary to pass the HTTP address of the target server to the Axios object and doing this multiple times in an application can lead to mistakes in the code (misspelling the HTTP address and therefore getting an HTTP error inside the application) and making maintenance of the code more complex should the HTTP address be changed to a new one, then maintainers would need to change every instance of it within the codebase.

To address this problem, Axios provides a function that creates an HTTP client instance with the given configuration variables and saves it to the memory of the application. This function can be called once in a Typescript file and then the result object can be exported from its original file and imported in other files across the application. This means that the configuration is set only once and there is no need to re-enter it once the program starts running. Once Axios is configured, the application can make all the necessary HTTP requests for this solution.

### 9.3 Implementing the Bootstrap framework

An important consideration when building this application was to ensure it will work in devices with different screen sizes, scale correctly, and make some of its components used in both horizontal and vertical orientations. Depending on the complexity of the design requirements there are different strategies for building responsive web applications. Considering the simplistic and clear design for CompanyX's website, it was decided to make use of a CSS framework that would contain ready responsive functionalities and that could be integrated with the React application easily.

Bootstrap is one of the most used CSS frameworks in web development. It makes use of a grid system to determine the positioning of different elements on the screen based on the screen viewport's height and width by setting breakpoints. The Bootstrap framework can be implemented in a React application by downloading the source code and including it in the main JSX document. This method requires the HTML elements in the application to implement the framework's classes, through which elements get the appropriate styling.

The solution for CompanyX, however, utilizes a different approach by adding another npm package called React-Bootstrap: This module essentially loads the Bootstrap framework's stylesheets to the project and provides ready JSX elements containing the needed class references already in their definition. This approach makes the application more declarative and less prone to styling mistakes by using ready elements.

### 9.4 Code editor

With the previous components and dependencies installed, the next step is to start coding the UI elements and implementing functionalities of the website. Microsoft's Visual Studio Code is used in this project as the main code editor because of its integration with WSL and Built-in Typescript features. Visual Studio Code is capable of being opened directly in the host machine while using a connection to the WSL's filesystem and thus allowing the codebase to be edited with all the features of the native Visual Studio Code application.

## 10 TESTING AND FINDINGS

### 10.1 Testing the web application and CMS usability

The application needs to be thoroughly tested before it can be deployed to a production server and open to the public. To enable testing of all the functionalities, it is necessary to run the application's full-stack in a machine that only testers can access.

Development of the application already provides most of the tooling needed for testing as well, since development needs to also have a functioning application where the programmer can see changes and how they affect the functionalities of the application.

The main difference between testing and development is that the application is executed with the exact code base intended to be deployed to production. This is to make sure that possible issues are found first in a safe environment and before the website is released to the public.

Before testing, the following services must be started:

- The webserver running and file permissions set to allow the CMS engine to execute code and serve requested pages
- Database server running and CMS configuration contains the correct database credentials that allow the engine to access and write data.
- Website's file directory contains updated to the latest master branch from the GIT repository, to ensure the codebase contains production code
- Npm scripts running inside the website's main directory and exposing the application in the set port.

With all the services running, it is possible to open the main website in the browser by using the development machine's set port. The website opens in the browser window using a predetermined web address. The application is only accessible in the development machine and not available through the internet at this point.

The application is tested in three different web browsers: Google Chrome, Mozilla Firefox, and Safari. Additionally, mobile responsiveness is tested using Google Chrome's and Mozilla Firefox's small screen emulators, which display the website in a smaller viewport that equals to the size of different mobile devices. The main aspects of this testing are:

- Does the website display the texts as expected?
- Are all the elements of the website in the right place?
- Do the elements on the screen scale correctly when emulating mobile devices?

- Does the time reservation embedded window work correctly?

The administration side of the website was tested by one of the company's employees responsible for the maintenance of the current website. He was asked to perform several tasks described in Appendix 2 and to give an overall impression of the CMS and its functionalities.

## 10.2 Test findings

Overall, the application worked as expected and could fetch the data from the external CMS correctly, showing titles and paragraphs in their right places. As seen in Figure 10, initial benchmarking showed that the React application loaded about 17% faster compared to a similar page load directly from the Drupal CMS. In this example (Figure 10) it is possible to see the scripting step taking more time in the React application, which tells about the React page being heavily JavaScript-dependent.

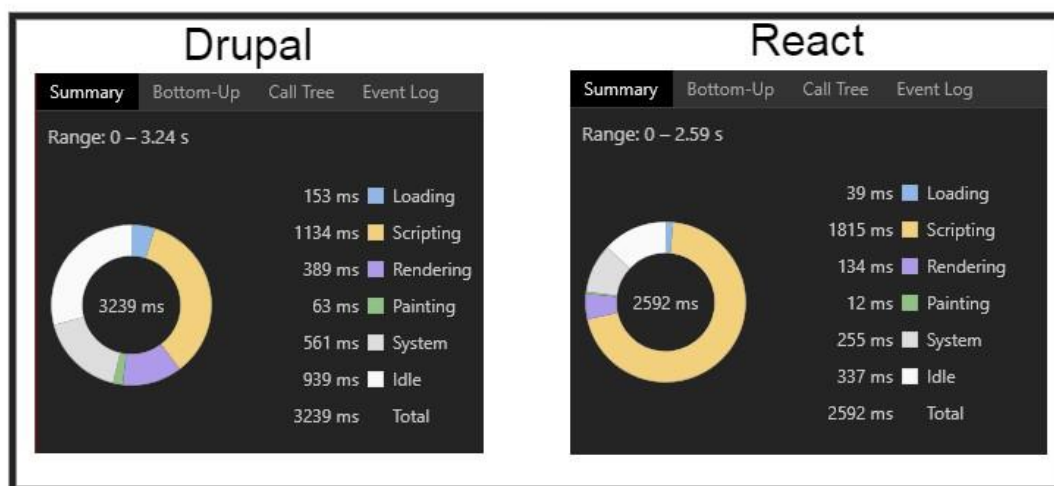


Figure 10 Drupal page and React page loading time comparison

Website responsiveness was tested using Google Chrome's and Mozilla Firefox's device emulators. These tools allow the application to be rendered in a smaller viewport that matches the dimensions of several devices such as mobile phones, tablets, and smaller laptops. This way, it is possible to see how the website looks like on a smartphone's screen.

The navigation bar shows its full contents only when the viewport's width is larger than 1000 pixels (see Figure 3). In smaller screens, the navigation bar hides its contents under

an animated menu that can be opened by the user, using up much less space on the screen.

Some of the sections of the webpage like the one shown in Figure 4 have elements displayed side-by-side. These types of elements work only in screens with a horizontal orientation since they take up more space sideways. Testing showed that these elements move from a side-by-side layout to a stacked layout when the viewport is about 770 pixels wide or less, accommodating better for smaller screens. Overall, the website behaved as expected in different screen sizes and orientations, indicating that the use of a CSS library for handling responsiveness led to a consistent webpage in different device sizes.

### **CMS testing results**

Generally, the CMS used in for the content management was positively graded by the company's representative. Most of the tasks were carried out with some guidance. Editing was clear and changes to the page were visible immediately in the website's front end.

Some concerns rose during the testing round:

- When a user is logged in, the CMS user interface is cluttered with setting and buttons that are unnecessary for most users. This made it harder to find the needed setting and places the user wanted to access. A possible solution to this would be to create specific roles for users so that other options and settings would be hidden from the user interface when logged in.
- The website and its contents are in Finnish, but if the user wants to have administration pages in English then some element of the UI fail to use the correct translation language. After some investigation on this issue, it turned out that Drupal has been aware of this issue and has not been resolved yet.



## 11 CONCLUSIONS

### 11.1 The current state of eBusiness and digital user interactions

This research showed the transition of CompanyX towards positioning themselves in a digitalized market by looking at the context of eBusiness practices and the history of the company with digital services.

The research created a better understanding of what needs to be done for a small company to find new value in the digital marketplace, by setting requirements for a digital platform, using modern technologies and assessment of the requirements aligned with the company's vision.

Most importantly, this research helped to create an understanding of the implications of digitalization and the impact of the agile nature of digitalized services. With this information, the company is aware that digitalization does not happen only once, but instead is part of a constant process of staying relevant in the digital world that requires the creation of momentum for digitalized innovation and ideas.

#### **How to enable CompanyX to create new value within the digital market?**

By investigating the contemporary status of digital businesses and eCommerce together with CompanyX's business needs and strategies, facilitating the design and implementation of a web platform that allows CompanyX to offer their products and services through the internet, and thus making an entry in the digital market.

#### **What new services does CompanyX want to offer to their customers?**

After studying the state of the company before the changes introduced by the implementation of a new web platform, and together with the company's vision, it was possible to identify the following new services:

- Ability to purchase products directly using a web service
- Ability to make time reservations online
- Ability to check price information, opening hours, and other company information using a smartphone or a tablet.

**What type of digital solution can carry the features required by CompanyX to enable the desired digital transformation?**

The final solution is such that enables CompanyX to create content and publish it online without any previous knowledge of web development. The solution allows website administrators to log in, review current content, make edits and publish new content. Additionally, the same system hosts the web shop's management tools, where administrators can manage stocks, orders and products. The solution is used by end-users through two separate websites, a webshop and the main website.

**Which technologies are necessary to build the digital solution for CompanyX and what does it take to build the solution?**

After gathering requirements and analysing the business opportunities wanted to be addressed with this digital solution, the choice of the right technologies is an important step to ensure stability and reliability of the system.

The chosen technologies follow common standards in web development: A web server, a database server, and the necessary extensions to run PHP code in the server. A Linux Apache MySQL PHP (LAMP) stack was used for the CMS instance, while the websites JavaScript environment is managed by the executable program Npm.

## 11.2 Suggestions for further research

For this specific case and company, the next step would be to measure the impact of the implemented digital solution in the business performance as well as in the customers' behaviour.

The nature of the business of CompanyX is highly seasonal, having sales peaks in specific times of the year. In order to get valuable metrics of the impact of a digital platform, it would be necessary to study customers and business numbers many times a year and analyse those results carefully in the context of digital business. The company could implement tools such as Google Analytics, to get a better idea of how many people visit their website.

This data analysis would open new insights in CompanyX's relationship with their customers and the impact of digitalization of their service offering, setting a good start point for the design of a roadmap for improving the digital services and executing more effective marketing campaigns.

CompanyX has adopted a magnetic card system that allows their customers to enter their gym facilities outside the reception's opening hours. The cards hold virtual balance according to the customers' purchased product and need to be re-filled during the reception's opening hours.

This presents a great start point for expanding the capabilities of the web platform used by CompanyX. This expansion would investigate the possibility of integrating the magnetic card systems with their web services, giving the chance to customers to manage their card's balance online, and purchase new balance as well.

## REFERENCES

### Electronic sources

- Abbott, N., Richard, J. 2016. Learning Drupal 8. PACKT Publishing [Accessed 30 April 2020]. Available at:  
<http://web.b.ebscohost.com.ezproxy.saimia.fi/ehost/ebookviewer/ebook/ZTAwMHh3d19fMTE2MzgZOF9fQU41?sid=7fd8be30-e4d2-4a67-b0e2-ea44d651778f@pdc-v-sessmgr06&vid=0&format=EB&rid=1>
- Bertoli, M. 2017, React Design Patterns and Best Practices. PACKT Publishing [Accessed 1 March 2020]. Available at:  
[http://search.ebscohost.com.ezproxy.saimia.fi/login.aspx?direct=true&db=e000xww&AN=1453391&site=ehost-live&ebv=EB&ppid=pp\\_Cover](http://search.ebscohost.com.ezproxy.saimia.fi/login.aspx?direct=true&db=e000xww&AN=1453391&site=ehost-live&ebv=EB&ppid=pp_Cover)
- Buytaert, D. 2019. How to decouple Drupal in 2019. Dries Buytaert [Accessed 30 April 2020]. Available at: <https://dri.es/how-to-decouple-drupal-in-2019>
- Cebi, Selcuk, 2013. Determining Importance Degrees of Website Design Parameters Based on Interactions and Types of Websites. [Accessed 15 March 2020]. Available at:  
[https://www.academia.edu/36059376/Determining\\_importance\\_degrees\\_of\\_website\\_design\\_parameters\\_based\\_on\\_interactions\\_and\\_types\\_of\\_websites](https://www.academia.edu/36059376/Determining_importance_degrees_of_website_design_parameters_based_on_interactions_and_types_of_websites)
- Chaffey, D., Smith, P 2017. Digital Marketing Excellence: Planning, Optimizing and Integrating Online Marketing. Taylor & Francis Group [Accessed 1 March 2020]. Available at: <https://ebookcentral-proquest-com.ezproxy.saimia.fi/lib/lab-ebooks/reader.action?docID=4834186&ppg=1>
- Chander, A. 2013. The Electronic Silk Road: How the Web Binds the World Together in Commerce. Yale University Press [Accessed 30 April 2020]. Available at:  
<https://ebookcentral-proquest-com.ezproxy.saimia.fi/lib/lab-ebooks/reader.action?docID=4585655>
- Coleman, A. 2019. KPI Report. Wolfgang Digital. [Accessed 30 April 2020]. Available at:  
<https://www.wolfgangdigital.com/kpi-2019/>
- Daekwan, K., Cavusigl, T., Calantone, R. 2005. The Role of Information Technology in Supply-chain Relationships: Does Partner Criticality Matter? Journal of Business & Industrial Marketing [Accessed 5 May 2020]. Available at: <https://www-emerald-com.ezproxy.saimia.fi/insight/content/doi/10.1108/08858620510603846/full/pdf?title=the-role-of-information-technology-in-supplychain-relationships-does-partner-criticality-matter>

- Day, G., Bens, K. 2005. Capitalizing on the internet opportunity. *Journal of Business & Industrial Marketing* [Accessed 5 May 2020]. Available at: <https://www-emerald-com.ezproxy.saimia.fi/insight/content/doi/10.1108/08858620510603837/full/html>
- Deek, F., McHugh, J. 2008. *Open Source: Technology and Policy*. Cambridge University Press [Accessed 1 May 2020]. Available at: <http://web.b.ebscohost.com.ezproxy.saimia.fi/ehost/ebookviewer/ebook/ZTAwMHh3d19fMjE0NDI2X19BTg2?sid=1ecbc0b8-d842-408e-96c1-ff51a90ab675@pdc-v-sessmgr03&vid=0&format=EB&rid=1>
- Drupal, 2020a. System Requirements. Drupal [Accessed 28 April 2020]. Available at: <https://www.drupal.org/docs/8/system-requirements>
- Drupal, 2020b. Security Considerations . Drupal [Accessed 30 April 2020]. Available at: <https://www.drupal.org/docs/8/core/modules/jsonapi-module/security-considerations>
- Fader, P. 2012. *Customer Centricity: Focus on the Right Customers for Strategic Advantage*. Wharton School Press [Accessed 30 April 2020]. Available at: <https://ebookcentral-proquest-com.ezproxy.saimia.fi/lib/lab-ebooks/reader.action?docID=5482545>
- Fedosejev, A., Bush, A. 2015. *React.js Essentials*. PACKT Publishing [Accessed 30 April 2020]. Available at: <http://web.b.ebscohost.com.ezproxy.saimia.fi/ehost/ebookviewer/ebook/ZTAwMHh3d19fMTA1NzU5OV9fQU41?sid=119d19ba-ed15-473a-8b68-4edafb3d6fee@pdc-v-sessmgr06&vid=0&format=EB&rid=1>
- Glaman, M. 2016. *Drupal 8 Development Cookbook*. PACKT Publishing [Accessed 1 arch 2020]. Available at: [http://search.ebscohost.com.ezproxy.saimia.fi/login.aspx?direct=true&db=e000xww&AN=1197539&site=ehost-live&ebv=EB&ppid=pp\\_Cover](http://search.ebscohost.com.ezproxy.saimia.fi/login.aspx?direct=true&db=e000xww&AN=1197539&site=ehost-live&ebv=EB&ppid=pp_Cover)
- Gore, A. 2018. Is My Single-Page Application SEO Friendly? *Vue.js Developers* [Accessed 26 April 2020]. Available at: <https://vuejsdevelopers.com/2018/04/09/single-page-app-seo>
- Horton, A., Vice, R. 2016. *Mastering React*. PACKT Publishing [Accessed 5 May 2020]. Available at: [http://search.ebscohost.com.ezproxy.saimia.fi/login.aspx?direct=true&db=e000xww&AN=1183988&site=ehost-live&ebv=EB&ppid=pp\\_FM-1](http://search.ebscohost.com.ezproxy.saimia.fi/login.aspx?direct=true&db=e000xww&AN=1183988&site=ehost-live&ebv=EB&ppid=pp_FM-1)

Hunt, P., O'Shannessy, P., Smith, D., Coatta, T. 2016. React: Facebook's Functional Turn on Writing JavaScript [Accessed 1 March 2020]. Available at:

<http://web.a.ebscohost.com.ezproxy.saimia.fi/ehost/pdfviewer/pdfviewer?vid=1&sid=3b778e39-1847-4d48-b5cf-59495c0d88fe%40sessionmgr4008>

IBM Cloud Education. 2019. Virtual Machines (VMs). IBM Cloud [Accessed 20 April 2020]. Available at: <https://www.ibm.com/cloud/learn/virtual-machines>

Ledford, J. 2007. SEO: Search Engine Optimization Bible. Wiley Publishing [Accessed 2 May 2020]. Available at: <https://ebookcentral-proquest-com.ezproxy.saimia.fi/lib/lab-ebooks/reader.action?docID=331516>

Lewis, N. Case, A. Ali-Gombe, A. Richard, G. Memory Forensics and the Windows Subsystem for Linux [Accessed 1 March 2020]. Available at: <https://www-sciencedirect-com.ezproxy.saimia.fi/science/article/pii/S1742287618301944?via%3Dihub>

McKeown, N., Durkin, M. 2016. The Seven Principles of Digital Business Strategy. Business Expert Press [Accessed 29 April 2020]. Available at: <https://ebookcentral-proquest-com.ezproxy.saimia.fi/lib/lab-ebooks/reader.action?docID=5089199>

Melnik, I. 2020 Single Page Application (SPA) vs Multi Page Application (MPA): Pros and Cons. Merehead [Accessed 26 April 2020]. Available at:

<https://merehead.com/blog/single-page-application-vs-multi-page-application>

Nance, C. 2014. TypeScript Essentials. PACKT Publishing [Accessed 1 March 2020].

Available at: <https://ebookcentral-proquest-com.ezproxy.saimia.fi/lib/lab-ebooks/reader.action?docID=1822800&ppg=1>

Narebski, J. 2016. Mastering Git. PACKT Publishing [Accessed 1 March 2020]. Available at:

[http://search.ebscohost.com.ezproxy.saimia.fi/login.aspx?direct=true&db=e000xww&AN=1225882&site=ehost-live&ebv=EB&ppid=pp\\_Cover](http://search.ebscohost.com.ezproxy.saimia.fi/login.aspx?direct=true&db=e000xww&AN=1225882&site=ehost-live&ebv=EB&ppid=pp_Cover)

Nielsen, J. 2002. Top 10 Guidelines for Homepage Usability. Nielsen Norman Group [Accessed 19 March 2020]. Available at: <https://www.nngroup.com/articles/top-ten-guidelines-for-homepage-usability/>

Nielsen, J. 2012. Usability 101: Introduction to Usability. Nielsen Norman Group [Accessed 2 May 2020]. Available at: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>

Nowak, M. 2019. Vue vs React in 2020: Which Framework to Choose and When. Monterail [Accessed 1 May 2020]. Available at: <https://www.monterail.com/blog/vue-vs-react-2020>

Perkin, N., Abraham, P. 2017. Building the Agile Business Through Digital Transformation. Kogan Page [Accessed 30 April 2020]. Available at: <http://web.b.ebscohost.com.ezproxy.saimia.fi/ehost/ebookviewer/ebook/ZTAwMHh3d19fMTQ5NDUwNF9fQU41?sid=8d3afa92-a059-427e-bd5c-90aa03c1c609@pdc-v-sessmgr04&vid=0&format=EB&rid=1>

Pratas, A. 2014. Creating Flat Design Websites. PACKT Publishing [Accessed 1 March 2020]. Available at: <https://ebookcentral-proquest-com.ezproxy.saimia.fi/lib/lab-ebooks/reader.action?docID=1685897&ppg=1>

Smith, J. Nair, R. 2005. Virtual Machines: Versatile Platforms for System and Processes. Morgan kaufmann [Accesses 1 March 2020]. Available at: [http://search.ebscohost.com.ezproxy.saimia.fi/login.aspx?direct=true&db=e000xww&AN=130249&site=ehost-live&ebv=EB&ppid=pp\\_Cover](http://search.ebscohost.com.ezproxy.saimia.fi/login.aspx?direct=true&db=e000xww&AN=130249&site=ehost-live&ebv=EB&ppid=pp_Cover)

Styled Components. 2020. Documentation. Styled Components [Accessed 30 April 2020]. Available at: <https://styled-components.com/docs>

Tao, Z., Yaobin, L., Bin, W. 2009. The Relative Importance of Website Design Quality and Service Quality in Determining Consumers' Online Repurchase Behavior [Accessed 15 March 2020]. Available at: <http://web.b.ebscohost.com.ezproxy.saimia.fi/ehost/pdfviewer/pdfviewer?vid=3&sid=e4d892ac-a1c6-486f-824f-6ac55ec45f29%40pdc-v-sessmgr04>

VMware. 2020. What is a Virtual Machine? | VMware glossary. VMware [Accessed 28 April 2020]. Available at: <https://www.vmware.com/topics/glossary/content/virtual-machine>

Vipul A. Sonpatki P. 2016. ReactJS by Example – Building Modern Web Applications with React. PACKT Publishing [Accessed 1 March 2020]. Available at: [http://search.ebscohost.com.ezproxy.saimia.fi/login.aspx?direct=true&db=e000xww&AN=1225885&site=ehost-live&ebv=EB&ppid=pp\\_Cover](http://search.ebscohost.com.ezproxy.saimia.fi/login.aspx?direct=true&db=e000xww&AN=1225885&site=ehost-live&ebv=EB&ppid=pp_Cover)

Vue.js 2020. Comparison with Other Frameworks. Vuejs [Accessed 1 May 2020]. Available at: <https://vuejs.org/v2/guide/comparison.html>

Weinman, J. 2015. Digital Disciplines: Attaining Market Leadership Via the Cloud, Big Data, Social, Mobile, and the Internet of Things. John Wiley & Sons [Accessed 1 March

2020]. Available at: <https://ebookcentral-proquest-com.ezproxy.saimia.fi/lib/lab-ebooks/reader.action?docID=4180342&ppg=1>



## APPENDICES

Appendix 1. Semi-structured interview with the company's CEO (Original interview held in Finnish)

1. What are the positive and negative things about the CompanyX's current website?
2. What is the current business situation in CompanyX?
3. What type of investment is the current website?
4. What are the main things you expect from a new website for CompanyX?
5. How often would you like to show new content on the company's website?
6. What are the most important features this website needs in your opinion?
7. How much are you planning to spend on the new website?
8. When are you planning to start using the new website?

Appendix 2. CMS administration first round training with company's representative (Original instructions given in Finnish)

- Log in to administration pages
- Open the front page's contents, edit some paragraphs and publish the new changes
- Check system status from the status pages
- Check user activity, who has logged in and out
- Log out and review changes in the live site