

Applying hypothetico-deductive reasoning to design, develop and evaluate dynamic website

Olga Savka

Bachelor's Thesis
Degree Programme in BIT
2021



Author(s) Savka Olga	
Degree programme Business Information Technology	
Report/thesis title Applying hypothetico-deductive reasoning to design, develop and evaluate dynamic website	Number of pages and appendix pages 61 + 9
<p>Emergence of new design trends, growing requirements to website responsiveness, unsatisfactory conversion rates, changed business strategy or branding policy, - any of these conditions can serve as a reason to redesign a business website.</p> <p>The thesis reports on a case of business website redesign determined by a combination of the above-stated conditions. Within the thesis project, the WordPress based website of Alejandra Beauty Salon was redesigned. The new design was implemented through developing own WordPress theme named Alejandra2020.</p> <p>Despite a large offer of Content Management Systems (CMS), WordPress remains the most popular one. About one-third of all websites are using WordPress software. The author questions expediency of adhering to WordPress CMS and developing own WordPress theme.</p> <p>Hypothetico-deductive reasoning is applied as a research method. The hypothesis about a suitable development approach is based on literature review and the author's previous experience. The hypothesis is verified through the project's implementation and usability testing of the resulting product. The chosen web development approach is evaluated with account to the development process and discovered usability issues.</p>	
Keywords WordPress, Content Management System, PHP, SASS, hypothetico-deductive reasoning, usability evaluation, headless CMS	

Table of Contents

1	<i>Introduction</i>	1
1.1	Background.....	1
1.2	Objectives.....	3
2	<i>Theoretical framework</i>	3
2.1	Research methods.....	3
2.2	Website Development Tools and Methods.....	4
2.3	User Study	14
2.4	Usability testing	15
2.5	Co-creation	17
3	<i>Implementation</i>	18
3.1	Design	18
3.2	User Studies.....	19
3.3	Development	21
4	<i>Evaluation</i>	36
4.1	Heuristic evaluation of the user interface	36
4.2	Usability testing of the user interface	44
4.3	User interface usability errors correction	50
4.4	Usability testing of the admin panel	52
4.5	Evaluation of the selected development approach	53
5	<i>Discussion</i>	58
6	<i>References</i>	60
	<i>Appendices</i>	62
	Appendix 1. Semi-functional prototype, mobile version	62
	Appendix 2. Admin User Manual	63

1 Introduction

1.1 Background

The thesis work was commissioned by the Alejandra Beauty Salon with the purpose of creating a new company's website. The new website is meant to replace the old one, which does not fully correspond the business requirements any longer.

Alejandra Beauty Salon was founded in Jätkäsaari, Helsinki, in 2012. Jätkäsaari district is a newly built residential area. Alejandra salon was one of the first enterprises offering their services in the district. The services provided by the salon include hairdresser and barber, hair straightening and extension, cosmetology services, nailcare, eyelash extension, permanent pigmentation, massage.

The old version of the website was started in 2017 by the previous developers. Its look is based on the Hellen WordPress theme customized for Alejandra. The website is integrated with a third-party booking system - Phorest. Apart from the website, the salon maintains its online presence through social media – regularly updated Facebook and Instagram pages.

The negotiations on the project started on September 1st, 2020. During the kick-off meeting with the salon's owner, the main requirements were established.

1. The new website must have a brand-relevant look featuring the style and palette of the company's logo.
2. The conversion functionality needs improvement. Clear navigation to the targeted master-associated reservations should be added. Call-to-action buttons and links should be properly placed.
3. New website must be more informative. Each salon's master will have a dedicated profile page with description of services and used products, as well as a gallery of works and products. The salon's story and interior picture are required. The contact page should contain detailed and visual contact information. A gallery of relevant and fresh images of the salon's works will be featured on a separate page.
4. English and Finnish versions of the website should be identical.
5. Finally, the salon's owner should be able to manage the site's content. This includes modifying textual and imaginary content, adding or deleting information about masters or their profiles whenever a new employee joins a team, or an existing master extends the range of services and products in use. The admin user should be able to add a page with information on current offers and discount campaigns whenever there are any. The same

way, the admin user should be able to exclude such page from the navigation menu when there are on ongoing campaigns.

The old site's version has a number of drawbacks. Its visual appearance does not reflect the company's brand image, it lacks information on the salon's staff, there is no association among the masters and their services, as well as no direct navigation to each master's booking section, no map with the salon's location, no salon's story and interior photo, no phone call button in mobile version, lots of mismatches between the Finnish and English versions. Finally, the old version uses HTTP protocol and has no Secure Sockets Layer (SSL) certificate.

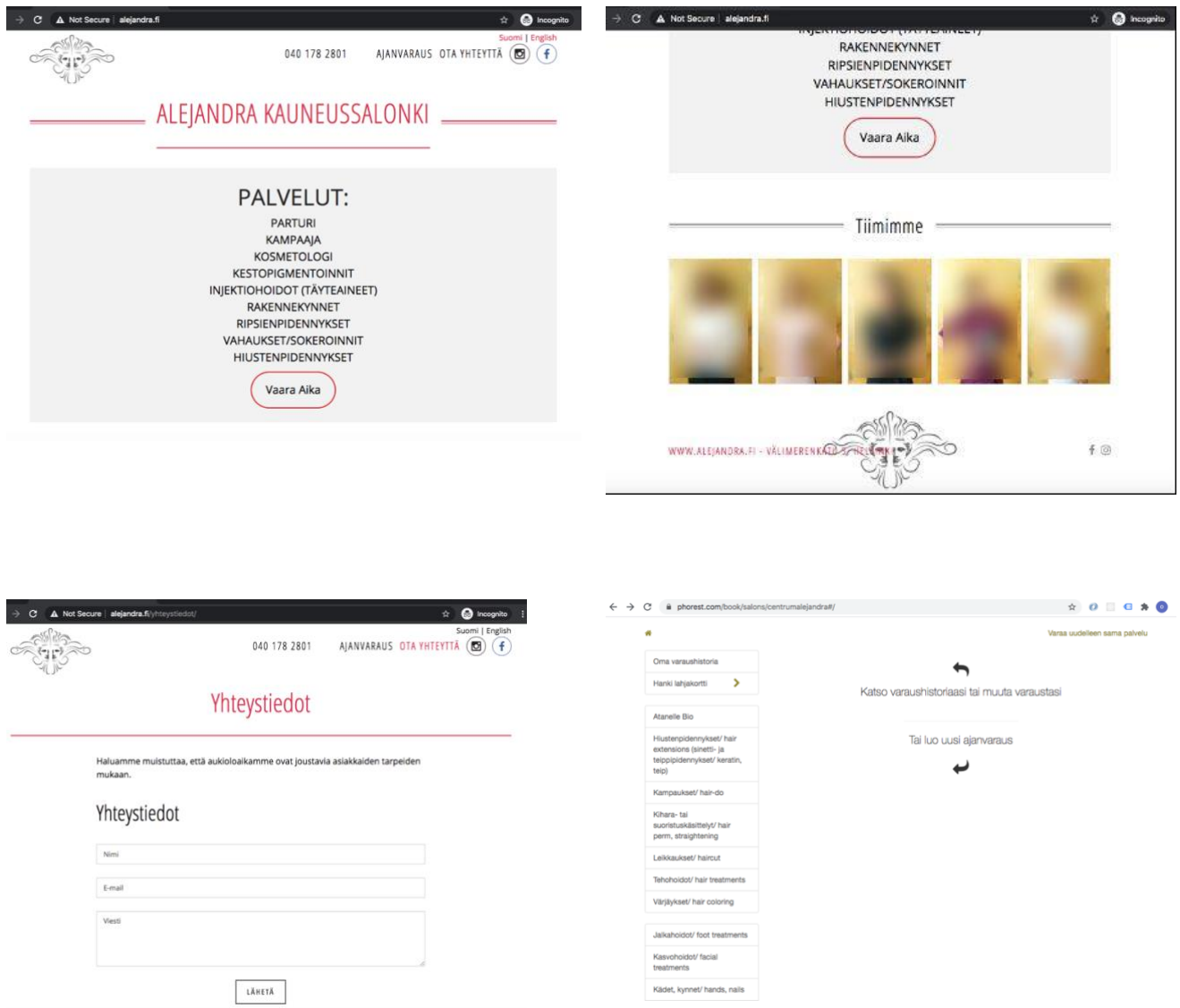


Figure 1. Screenshots of the old version pages, the Finnish-language part: the front page and contact page. Booking menu link leads directly to the Phorest application.

1.2 Objectives

The objective of this thesis project is to renew the design and develop a new version of the website with unique brand-relevant look, extended functionality, editable content, and enhance the website's security.

The set of requirements to new website is predefined by the commissioning party. Guided by the customer's requirements, the author develops a hypothesis on the suitable implementation method. The author intends is to explore the relevance of using a WordPress theme created from scratch in order to meet the stated demands. The exact evaluation criteria are to be defined in course of literature review.

Usability evaluation of the ready website will be conducted.

2 Theoretical framework

2.1 Research methods

The research method of this thesis can be qualified as a hypothetico-deductive reasoning.

This approach implies the following steps:

- Problem identification
- Developing a hypothesis
- Making implications by deduction
- Practical testing of the hypothesis

Based on the testing results, the hypothesis is either rejected or refined. (Walliman, 2011)

Hypothetico-deductive method is also described as a problem-solving method that unites both deductive and inductive logics. Some distinguish five stages of the method:

observation, hypothesis, prediction, verification, and conclusion. However, the borders between the stages are often vague, or the sequence of stages may be mixed.

1. At the observation stage certain relation or pattern is noted.
2. A hypothesis is made assuming that the relation or pattern is real.
3. A prediction is deducted in form of a conditional if-then statement, operations required for verification of the statement are determined.
4. The prediction is verified through new observations performed in form of defined operations: controlled experiment or observational study.
5. Conclusion is made stating if the hypothesis was true or false depending on the extent to which the study results comply with the prediction. (Aronson, 2006)

"This is a process of operations of raising precise questions converting the questions into

hypotheses. The logical consequences of the hypotheses are obtained. They are tested and verified.”(Krishnaswamy, Sivakumar and Mathirajan, 2006)

Öivind Andersson points out that initial hypothesis is a “preliminary attempt” of explanation without much observation done. The consequences are deduced from the hypothesis. Further the hypothesis is tested against observation which may take form of controlled experiment, e.g. (Andersson, 2012)

In this project, the method phases can be presented as follows:

Problem statement.

A website should be developed to comply with the requirements predefined by the commissioning party, which is dynamic website with unique design and editable content.

Hypothesis.

Based on my previous experience and documentation review, I assume developing my own WordPress theme from scratch is a suitable implementation scenario.

Implications.

Own theme development must be relatively easy and time-saving way to code a website exactly reflecting the design solution. Own theme can be applied at the WordPress platform without harm to its other functions and security.

Testing.

Creating the website by developing own WordPress theme, evaluating the process and the results. Exact testing criteria to be defined in course of literature and documentation review.

2.2 Website Development Tools and Methods

The existing business requirements imply need for styling freedom for developer and admin panel to edit the content for the admin user.

While choosing the right web development tools, the following options were considered by the author:

1. React.js application with own backend and admin interface
2. React.js application with a decoupled CMS
3. WordPress as classic CMS with ready-made templates
4. WordPress as classic CMS with own developed theme

ReactJS Application

Having most of hands-on development experience in JavaScript and React.js, the author considered creating a React application.

React.js is a JavaScript library for developing high-performance interactive UIs. Efficiency is achieved through use of the virtual DOM (Document Object Model) pattern – virtual representation of the real DOM. Real DOM updates are minimized due to diffing the current and changed DOM states. React DOM package compares the UI-tree with its previous version and modifies only the changed nodes.

In addition to faster performance, React simplifies application development. Thanks to its declarative API, a developer only takes care of what UI looks like upon each change of view's state. (React Documentation, 2020)

React application are often developed following the Single Page Application (SPA) model. SPA model is also used with other JavaScript libraries and frameworks: AngularJS, Vue.js, Ember.js, etc.

In SPA, UI representations are managed on the client's side, and DOM is manipulated with JavaScript code. This eliminates need for browser refresh upon each user input. SPA provides better user experience with shorter user wait time. While code responsible for UI behavior is no longer kept on the server, front-end can be maintained independently. (Scott Jr., 2020)

One of the problems with SPA is Search Engine Optimization. Being initially set for indexing static content within HTML, search engines have difficulties in indexing and ranking the dynamically generated content contained within JavaScript code. A SPA application requires profound testing, since any JavaScript error may leave a page or even whole website unindexed. While Google takes steps to adapt Googlebot for SPAs, other search engines remain unreliable. (Mioni, 2019)

Concluding the above given statements, React single page application is a good choice for websites with abundant interactive elements: user inputs, data downloads, adding and deleting items under own account, etc. React-based implementation gives access to the numerous react libraries and packages for accelerated development, versatile styling and animation.

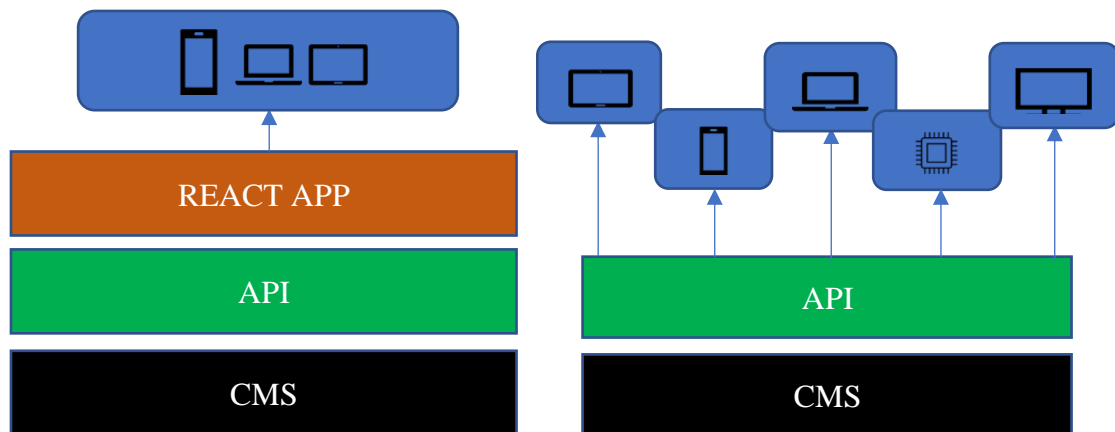
On the other hand, a React SPA would require additional efforts and further research to ensure good search engine optimization. To enable content editing, a React app would further need own admin panel and a backend.

ReactJS application with decoupled CMS

As an option, the author considered utilizing React UI with a decoupled CMS. In classic CMS implementation both content and user interface are managed and delivered by server-side, in the single environment. A decoupled CMS is devoid of its front-end presentational functions. It is set up to deliver content for consumption by any application, e.g. a React app. (Krofegha, 2020) Compared to a stand-alone React application, use of a decoupled CMS eliminates need of own admin panel and backend.

Decoupled and headless CMS history is started with advance of Web API – Application Programming Interface. API provides access to a system via HTTP requests, with data presented in a standardized format: XML or JSON. With the rise of mobile device usage, APIs gain importance in delivering content to mobile devices. Further, the concept of “omnichannel content delivery” was developed as a strategy of reliable content delivery across multiple channels whatever end-user may utilize. In decoupled and headless CMSs server and interface(s) communicate with each other through an API. (Heslop, 2018)

Though terms ‘headless CMS’ and ‘decoupled CMS’ are often used as synonyms, there is a difference in architecture design. Decoupled CMS has one isolated front-end which communicates with the back-end via API. In a headless CMS architecture, API is linked to multiple front-ends. (Winkels, 2019)



Decoupled CMS with React UI

Headless CMS

Already in 2015, at WordCamp US event – the regular gathering of WordPress enthusiasts, WordPress co-founder Matt Mulleberg highlighted the future opportunities of WordPress REST API with JavaScript apps running on top. He said: “I believe quite strongly that JavaScript and API-driven interfaces are the future of not just WordPress but the web”.(Gooding, 2015)

During the university work placement, the author participated in building a React SPA that fetched content with WordPress API. The project's goal was to replace the old classic WordPress website. The new website had completely new design, structure and style, several additional interactive elements. It was decided to continue using WordPress platform as decoupled CMS, while the new React UI was created from scratch. New post types in admin panel were created with 'Advanced Custom Fields' (ACF) plugin, while corresponding API endpoints were added with help of 'ACF to REST API' plugin. API calls were performed with Axios HTTP client.

A decoupled CMS approach is advantageous choice for developers with limited back-end development and API development skills.

Classic CMS

Another possible approach is to stick to the traditional CMS tools.

A typical content management system is a server-based software that interacts with the content stored on the same server or in a separate storage. The system comprises editing interface, content storage, publishing engine. Content management include the CRUD operations on the content: creating, updating, deleting.(Barker, 2016)

With a classic CMS, users can create and update websites without any coding skills, which explains popularity of CMS platforms.

W3Techs – a division of Q-Success consulting company – is a service that collects data on usage of technologies in building and running websites. Its up-to-date surveys are available on the official website. According to W3Techs' survey as of October 2020, the CMS market is dominated by WordPress, Shopify, Joomla, Drupal and Wix. WordPress is an obvious leader with market share of 63,7%, it also takes position of the fastest growing CMS.(W3Techs, 2020)

Content Management Systems

Most popular content management systems

© W3Techs.com	usage	change since 1 October 2020	market share	change since 1 October 2020
1. WordPress	39.0%	+0.5%	63.7%	+0.2%
2. Shopify	3.1%	+0.2%	5.0%	+0.2%
3. Joomla	2.2%	-0.1%	3.6%	-0.1%
4. Drupal	1.5%		2.5%	
5. Wix	1.5%		2.5%	

percentages of sites

(Source: w3techs.com)

According to wordpress.org, over 39% of all the websites are WordPress-powered. (*WordPress Documentation*, 2020)

Since detailed comparison of different CMS platforms is beyond the thesis scope, the author will point out advantageous features of WordPress as classic CMS.

First released in 2003 as an open-source blogging platform, WordPress grew into a the most wide-used CMS.

The following are attractive features of the CMS including those stated on the official website of wordpress.org:

- User friendly simple admin panel
- Role-based access to admin UI
- Easy language customization
- Time-tested tools with huge community support through forums, blogs, offline meetups
- Detailed documentation
- Suitable for any type of a website
- Regular updates and releases keeping the system up to date
- Full compliance with W3C standard, which ensures the browser compatibility. W3C is Open Web Platform working at standardization of Web technologies
- SEO out of the box supplemented with additional SEO plugins for better control
- Despite big choice of ready-made solutions, WordPress allows for creating own themes, plugins and content types, which make it more flexible and adaptable to concrete business needs.

The authors of “Building Web Apps with WordPress” discuss pluses and minuses of building web application with WordPress. According to them, the advantages of a WordPress site are:

- Easy content management
- User management with controlled access to content management
- Big choice of free plugins that allows for fast adding new features without need to code them

The downsides of WordPress are:

- Potential security problem due to the open-source character
- Not all plugins are reliable, some do break at WordPress updates

- Unlike JavaScript frameworks, WordPress sites cannot handle asynchronous real time events. This makes it unsuitable for such applications as chat rooms or multiuser games.(Messenlehner and Coleman, 2019)

Table 1. provides a list of criteria to compare the above stated development approaches. After the project's completion the defined criteria will be used for evaluation of selected approach.

Table 1. Classic WordPress (WP) vs. WordPress as decoupled/headless CMS as applied to Alejandra case

Criteria	React app + headless WP with WordPress REST API	Classic WP	Alejandra case
Performance speed	Higher due to virtual DOM and SPA architecture	Lower on interactive elements due to frequent browser refreshes, Excessive use of plugins slows down the website	In the Alejandra website, most of user interactivity lies with the reservation system, which is provided by the third-party service and will be embedded into the website within an iframe. The rest of the website has limited number of interactive elements. Therefore, React SPA is not required to accelerate user interaction. Excessive use of plugins will be avoided through hard coding a part of functionality using JavaScript.
Content management	WP admin panel can be used with limited functionality, only pure content – texts, media files – are updatable for the admin user	Full functionality of WP is available, including: WYSIWYG (What You See IS What You GeT) editor, live preview option, admin user can not only update content, but modify the layout, add/delete full pages and sections, update navigation menus, admin user has certain freedom of styling customization.	Maximum possible freedom of actions for admin is required since the company has no IT department. For the same reason a convenient, user-friendly and time-tested admin UI with maximum capabilities is a reliable choice.
Availability of WP plugins	Only plugins that control functions of the admin panel. The functionality of installed plugins cannot be transferred to REST API and used as headless CMS	All available WP plugins	Some WP plugins that have been successfully used for the old website, can be preserved for the new version
Security		Some plugins may cause security threat. Regular updates are required	Security risks can be minimized by careful selection and limited use of plugins. The admin user will be instructed on maintenance

SEO	Problematic SEO	Out of the box SEO, SEO plugins	Convenient SEO management is important since neither the developer nor the admin user is a SEO specialist.
Content fetching	Developer needs to fetch data from the REST API which may imply API access problems, numerous call requests for different data types, inconvenient data formats	The author needs to learn PHP language to query the data from the database	Possible difficulties with content fetching are not clear at the early stage of the project
Scalability	In the future, REST API can be used for other device-specific and platform-specific implementations in the future. More interactive elements can be added without harm to performance.	A classic multi-page website	Alejandra website has relatively simple and predictable functionality, it is unlikely to scale up considerably in the future. Responsive design and mobile-first approach will be used for small screens.
Freedom of styling	Anything that can be implemented with JavaScript and CSS	It depends on the choice of tools: ready-made theme, child theme or own theme	By developing own WP theme, the author can utilize CSS and JavaScript. The WP theme development standards must be followed, but they do not restrict styling options

Ready-made WP theme vs. own developed WP theme

WordPress Codex – the repository of all the WordPress documentation – defines WordPress theme as “files that work together to create the design and functionality of a WordPress site”. (WordPress Codex, 2020)

Themes differ from plugins in the control area: a theme defines presentation of the content, while a plugin is responsible for site’s behavior and feature. (WordPress.org, 2020)

A theme is responsible for placing the content stored in the database into the right page layout. This way theme can generate pages dynamically. Theme is handling the formatting, while the content – textual, visual, - can be managed separately. Formatting and layout are coded into the template files using HTML and PHP. The look of pages is determined in separate CSS stylesheets. (MacDonald, 2020)

Themes can be compared to Views in the Model-View-Controller design architecture. Views contain the code responsible for displaying data to end-users, themes contain templates and logic. (Messenlehner and Coleman, 2019)

Interesting experience of building an own made WordPress theme is described by Aaro Helander in his thesis “Building a WordPress site for Sports Academy of Kouvola” written in 2015. Aaro started with creating own theme from scratch. He coded a fully functional prototype using Bootstrap 3.0. Then he converted it into a WordPress theme, installed and activated from the WordPress admin dashboard.

After struggling for a while with own theme development, Aaro decided to switch to purchasing Divi 2 premium theme by Elegant Themes framework. He notes: “The most exciting part of this project was when I realized that I’m far better off throwing my own design out of the window, and using the Divi 2 theme from Elegant Themes to create the same layout again.”

According to Aaro, the benefits of such choice were as follows:

1. High development speed – it took him a week to create the website with Divi 2 page builder
2. Guaranteed security of the resulting website
3. Minimum use of plugins which improves the loading speed
4. In-built responsiveness of the web pages
5. Regular theme updates and new releases by Elegant Themes
6. Developer support.

Therefore, downsides of an own made theme include excessive use of plugins, unreliable security, slow coding process. (Helander, 2015)

“Build Your Own Wicked WordPress Themes” work is written by a team of authors who discuss the advantages and methods of own theme creation with regard to WordPress 3.0. Even though there have been a number of new releases since version 3.0, the book still provides many valuable insights. Among other reasons to build own theme, the authors note ability to convert own design solutions into WordPress templates. (Cole *et al.*, 2010)

The theme’s scope should be relevant to the purposes of a website. “Remember, WordPress is supposed to make the lives of authors and visitors easier. If a theme includes too many features, using them on a regular basis becomes cumbersome.”

Following the “content-driven design” approach, the authors differentiate theme types:

- Blog Theme

- Image-centric Theme
- Video Site Theme
- Business Site Theme
- Mix of content types

The primary content types must be defined ahead of the development. (Cole *et al.*, 2010)

The authors suggest using existing theme frameworks to simplify own theme development. A framework contains ready-made mark-up, basic styling and functional elements. “For beginner themers, using a framework means that you can spend less time putting together your code, and more time concentrating on your design. It’s also a great way to learn about how WordPress themes are put together, especially if you’re fairly new to PHP.”(Cole *et al.*, 2010)

Unlike the aforementioned authors, Joe Casabona in his book “Building WordPress Themes from Scratch” suggests building own theme framework. Written in 2012 with regard to the WordPress version 3.3, the book is a step-by-step manual on building own custom themes and plugins. A functional prototype is built with HTML and CSS on the basis of the design solution drawn in Photoshop. The prototype is converted into a functional WordPress theme with editable content. To make various content editable from the admin panel, the author uses Custom Post Types. Further, the author adds flexibility to the theme and enables design customization from the admin panel without need of altering the code or creating a child theme. This is achieved with Theme Options and Widgets. In the last chapter the author teaches how to create own plugins. WordPress Codex documentation is used for references as a best practices source. (Casabona, 2012)

The author explains idea of developing own theme framework by the principle of reuse. Since the core theme structure is usually the same, a framework of reusable folder structure, code parts, functions and templates saves time and efforts when developing multiple themes. Own framework can be as well substituted with a ready-made third-party framework. In this case own code will form a child theme to the framework. However, the author points out that some precautions must be taken:

1. own CSS code overwrites the parent’s code, so parent’s styling must be imported into the child theme
2. a child theme inherits parent’s functions, therefore if own function happens to have the same name as a parent’s function, it results in error or malfunctioning. (Casabona, 2012)

2.3 User Study

Understanding interests of all the stakeholders is important to create a usable product. Stakeholders include both end-users and business representatives. In case of Alejandra website, the business owner also plays end-user's role as the website's admin panel user.

Sometimes stakeholders are referred to business only as an opposite to end-users. Lisandra Maioli in her book "Fixing Bad UX Designs", regards stakeholders as business representatives. She notes that user studies should be preceded with stakeholder interviews as helping to formulate questions for users. It is recommended to hold semi-structured interviews in form of a comfortable conversation with prepared topics. The rules of stakeholder interview include the following:

- avoiding leading questions, bias and assumptions.
- being focused on the company's culture, business problems and goals.
- not asking for any concrete design solutions.
- listing business problems and symptoms related to bad UX of the current product; (Maioli, 2018);

There is a great variety of methods to study the needs of end-user. Nielsen Norman Group website describes 20 types of user research methods classified against three criteria types:

- Attitudinal vs. Behavioural
- Qualitative vs. Quantitative
- Context of Use (Rohrer, 2014)

One of the ways to understand end-user is creating a user persona. This method is widely used in initial stage of a design project. A persona is a user profile that might serve as a presentation of the target audience. These are three types of persona:

1. Proto persona is a lightweight persona option built based on assumptions and existing knowledge about the clients.
2. Qualitative persona is based on small-scale qualitative research, e.g., interviews or usability studies.
3. Statistical persona is based on statistical analysis of large-scale surveys. (Laubheimer, 2020)

In this project, a proto persona method is utilized. While a traditional persona is drawn based on the user research with information received directly from the target audience, a proto persona is

based on the business vision of its customers, on what the company already knows about its clients. (Maioli, 2018)

The content of a proto persona includes:

- User's personality
- Behaviour patterns
- Demographics (age, occupation, etc.)
- Needs and goals (Maioli, 2018)

Proto personas help the developer to understand the client's vision of the future website during the co-creation designing. The client knows the salon's audience to certain extent while the developer does not. Additionally, it helps to involve the team into the project discussion.

2.4 Usability testing

Usability is a product's quality characteristic. It defines various aspects of how user interacts with the product, i.e.:

- if the product is easy to use.
- whether it takes long to learn how to use it.
- if it is easy to memorize how to use it.
- if user can be efficient when using it.
- if user is likely to make mistakes.
- if user feels satisfied with the product; (Nielsen and Loranger, 2006)

The above stated criteria can be evaluated through user testing or expert inspection.

According to Carol M. Barnum, a usability test is determined by the following:

What is tested?

Where it is tested?

How it is tested?

What to test is defined by the stage of product's development. From this viewpoint, usability tests are *formative* and *summative*.

Formative tests held during development for early problem detection. These are small studies with 5-6 participants. It is required to define user group, prepare task-based scenarios, utilize 'think-aloud' process, test in iterations.

Summative tests are performed on ready product to check compliance with requirements. Those are either large or small studies depending on product's size and complexity. (Barnum, 2010)

By location usability tests are classified as *lab* tests, *field* tests and *remote* tests.

Lab tests are either conducted in formal labs with specialized equipment, or informally with a laptop and a table. *Field* tests are held in user's environment either with portable equipment or without it. While lab testing ensures undisturbed controlled testing environment, field tests give understanding of the actual conditions the product will be used under.

Remote testing is not tied to specific location. It can be either synchronous moderated or asynchronous unmoderated. Moderated testing is conducted in tester's presence. Tester can record all what is happening on the participant's computer. However, with desktop-sharing tester cannot observe the tested participant. It is also harder to convey testing instructions and cope with possible technical obstacles on the participant's side.

Asynchronous remote testing is performed in automated mode with help of web application. It allows to reach unlimited number of people, which is beneficial for statistical validity. Asynchronous tests exclude complicated, as well as possibility to identify reasons of test failure. (Barnum, 2010)

How to conduct a test? The answer depends on the goals and defines the test's design structure: typical, benchmarking, comparison of designs, competitive evaluation. (Barnum, 2010)

In a *typical* test, participant gets tasks based on scenarios and provides feedback on experience. This approach is normally applied in formative testing of products under development.

Benchmarking is usually a summative evaluation of a ready product or result of development cycle. The test provides quantitative data in form of UX metrics, as e.g. application retention rate or time required to make a purchase. (Moran, 2020)

For *comparison of designs* a user gets two or several design options, and the tester checks if there are preferences.

A similar approach is found in *A/B testing*: in this case comparison is done between two designs that are identical except for one variation. (Maioli, 2018)

In *competitive evaluation*, participant performs tasks on the tested product and competitor product, wherein preferences are revealed. Either overall design or separate features and elements are compared. Such evaluation may take form of usability testing or an expert evaluation. Competitive evaluation reveals strong and weak points, shows how competitors solve similar design problems. (Schade, 2013)

By type of produced data usability testing can be *qualitative* and *quantitative*. Qualitative data is a direct user assessment of usability, it helps to detect major UI problems. Qualitative approach is usually used for formative testing held in small groups. Quantitative data takes form of metrics, such as satisfaction and task-completion ratings, time required to complete a task, etc.

Quantitative studies are normally conducted as summative: in large groups on ready products. (Budi, 2017)

Ideal usability test should be conducted with real users provided with real meaningful tasks. Real users are those for whom the product is designed and created. Testing must be performed with view to real product's goals and within the actual usage context. While user personas can be utilized to pick the right participants, the goals and context are reflected in scenarios – 'situations in which your personas would use your product to match their goals' (Barnum, 2010)

Usability testing is fully focused on end-user. Wider usability toolkit includes techniques with professionals' participation, such as cognitive walkthrough, heuristic evaluation, participatory design. Heuristic evaluation, also called expert review, is the tool often used hand-in-hand with usability testing. (Barnum, 2010)

10 heuristic principles offered by Jacob Nielsen serve as evaluation criteria in heuristic evaluation. They are not strict rules, but rather practical guidelines. 10 heuristics include:

1. *'System status visibility'*
2. *'Match between system and real world'*
3. *'User control and freedom'*
4. *'Consistency and standards'*
5. *'Error prevention'*
6. *'Recognition rather than recall'*
7. *'Flexibility and efficiency of use'*
8. *'Aesthetic and minimalist design'*
9. *'Help users recognize, diagnose, and recover from errors'*
10. *'Help and documentation'*(Nielsen, 2020)

The results of heuristic evaluation may be presented as a list of problems related to certain heuristic principles and scaled by their severity from 0 – 'not a usability problem' to 4 – 'usability catastrophe'. (Maioli, 2018)

2.5 Co-creation

Co-creation is used as a designing approach in this project.

Co-creation is a process of product or service design in which a significant share of design ideas comes from the customer. Co-creation process may engage different stakeholding parties: end-users, employees, business owners. The stakeholders play active role through all the design cycles.

In co-creation, the interaction process between the designer and the customer is seen as value itself. As the authors of “The Co-Creation Paradigm” state, co-creation is a step aside from the conventional value creation concept, since the value is created together with the customer. “Build it and they will come” approach is substituted with “build it with them, and they are already there”.(Ramaswamy and Kerimcan, 2014)

3 Implementation

The project’s implementation includes three phases: design, development and evaluation.

The evaluation part is subdivided into:

- evaluation of the ready website’s usability,
- evaluation of the selected development tools against the criteria set in course of literature review and presented in Table 1.

3.1 Design

The new design was developed in co-creation with the client in process of several iterating design cycles. It started with detailed talks on the customer’s vision of the new website. We discussed the drawbacks of the current site version, the prospective website’s structure and navigation, content, imaginary, layout, use of colours, use of logo. That was followed by creation of the sitemap and a number of semi-functional prototypes.

The new site structure and components

The new website has the following structure:

1. Pages: Home, Booking, Contact, Gallery, Master profiles, Offers, - all pages have Finnish and English versions.
2. Layout components: header, footer, hero image, story section, master profile preview card.
3. Functional features: mobile menu, top desktop menu, footer menu, image gallery lightbox with slider, contact form, map, embedded responsive reservation pages of Phorest booking system, call-to-action button, floating phone button.

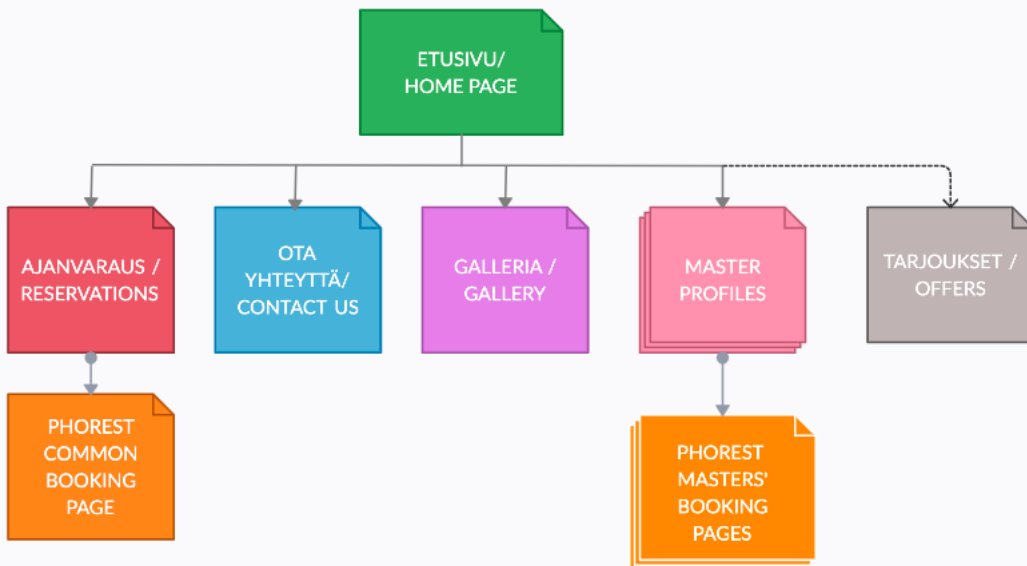


Figure 2. The new sitemap

Admin user will add the Offers/Tarjoukset page to navigation whenever there are offers.

Semi-functional prototype

The semi-functional prototype was implemented using Figma(*figma.com*, 2020). It reflects the layout of pages, navigation routes, colour palette, use of icons and images. The prototype uses sample text content. See Appendix 1.

3.2 User Studies

The author talked to the three salons master in format of a semi-structured interview. Interviews with the hairdresser, nail master and cosmetologist shed light on their client audience.

The shared characteristics of the client groups include the following:

1. about 99% of all the clients are women
2. the age ranges from 19 to 50 years old
3. approximately half of the clients are residents of the Jätkäsaary district, even though there are many clients from other parts of the capital area.
4. most of the clients are regular visitors, the share of one-time customers is relatively small.

At the same time the interviews revealed some differences between the salon's client groups depending on the kind of services they use. The results were used to create two proto personas.

A *hairstylist* client persona.

“Hanna. 43 years old. Hanna works as a financial specialist in the bank. She is divorced. Hanna is not overloaded with family problems or children upbringing. She travels a lot, actively goes in for sports, practices yoga, avoids junk food. Hanna takes good care of herself, she values quality over price. She visits the salon on the regular basis, but not very frequently. She prefers hairstylist services with a long-term stable effect, which saves her worrying about her hair between the rare visits.”

A *cosmetologist* client persona.

“Erica. 22 years old, a student. Erica is not married. She does not like to share details of her private life, so very little is known about it. Erica has been living in Jätkäsaari for several years by now. She learned about the Alejandra salon accidentally and was surprised she did not notice it earlier despite living nearby. She cares about her skin health. Her visits to the cosmetologist have two purposes: to improve the skin condition and have relaxing past-time. Even though the Alejandra cosmetologist offers a very wide range of services, Erica always picks one and the same specific therapy that corresponds her needs.”

The author made the following conclusions from the gathered information:

1. Location content, such as the map, exact address, mentioning of the Jätkäsaari district for SEO purposes, - are crucial. Considerable number of visitors come from other parts of the city. It is also known that locals may be unaware of what the neighborhood can offer. It must be noted that Jätkäsaari is a newly built district, so many people are still unfamiliar with it.
2. The salon offers wide range of services and attracts different types of people. This explains importance of diversified content subdivided section-wise and targeted navigation routes. It should be viewed as part of acquisition marketing.
3. At the same time, it is known that most of the customers are existing regular clients. Assuming they are familiar with the services and know, what they want, the shortest possible way to booking page must be provided as means of customer conversion. The call-to-action button on the top of the front page performs corresponding role.
4. Most of the client are working-age busy people, so the website should convey on-point reference information. Ad hoc content can be posted in social media.

Additionally, based on the author’s observation during the autumn 2020, the salon’s team is actively maintaining the salon’s social media pages. The website’s galleries may feature a limited number of carefully selected photos, while a wider choice of images may be always found in Instagram and Facebook.

3.3 Development

Development of Alejandra2020 theme was started in September 2020.

The code is found in the GitHub repository <https://github.com/OISavMe/Alejandra-Theme>

Local development environment

Local environment, also called a test environment, provides freedom of experimenting while keeping the live website safe and intact.

Website is run on local host, while modifications are done in Visual Studio Code editor and from the dashboard of the local WordPress installation. Apart from code changes, a developer can try different plugins and themes.

Local WordPress development requires a server stack consisting of:

- an HTTP server processing HTTP requests and serving out web pages,
- a database server to create and manage a local database,
- a relevant programming language.

In this project MAMP software stack is utilized. MAMP is an abbreviation for Macintosh, Apache, MySQL, and PHP. Apache is an HTTP server – the Apache Software Foundation open-source project. MySQL is Oracle provided open-source database. PHP - Hypertext Pre-processor – is a web-oriented programming language.(Toh, 2020)

Development tools

HTML mark-up, PHP and JavaScript languages, as well as CSS with SASS pre-processor have been utilised for the theme creation.

PHP stands for “Hypertext Preprocessor” is an open-source scripting language. It is qualified as general-purpose language, but it is most suitable for web development. PHP is widely used to code dynamically generated web pages, and it can be conveniently embedded into HTML code. Official documentation is found at www.php.net. (*PHP Documentation*, 2020)

```

functions.php
You, seconds ago | 1 author (You)
1  <?php
2  require_once('custom-posts.php');
3
4  // Loading style and javascript files
5  function scripts() {
6      wp_register_style('style', get_template_directory_uri() . '/dist/app.css', [], 1, 'all');
7      wp_enqueue_style('style');
8
9      wp_enqueue_script('jquery');
10
11     wp_register_script('app', get_template_directory_uri() . '/dist/app.js', ['jquery'], 1, true);
12     wp_enqueue_script('app');
13 }
14 add_action('wp_enqueue_scripts', 'scripts');
15
16 // Adding menus
17 add_theme_support('menus');
18
19 function register_my_menus() {
20     register_nav_menus(
21         array(
22             'header-menu' => __('Header Menu' ),
23             'footer-menu' => __('Footer Menu' ),
24         )
25     );
26 }
27 add_action( 'init', 'register_my_menus' );
28
29 // Disable the Admin Bar
30 add_filter( 'show_admin_bar', '__return_false' );
31

```

Figure 3. functions.php file

Since a theme's main function is to control the look of pages, large amount of coding happens in stylesheets. In this project, SASS (Syntactically Awesome Stylesheets) pre-processor is used to encode styling. SASS is pre-processor of CSS language, which allows to avoid code repeating. With help of variables, operations, mixins, functions and selector nesting coding becomes faster and easier, while the resulting code is readable and clean. (Prabhu, 2015)

By default, any WordPress theme requires the *style.css* file as a mandatory source of styling. Therefore, SASS code must be compiled into plain CSS. For this purpose, Webpack is utilised in combination with Laravel Mix library.

Webpack is a bundler program that is also used to transform and package resources, which are then loaded asynchronously on demand. Webpack compiles resources with dependences into static files. (Webpack)

Laravel Mix is an API wrapper for Webpack, it simplifies configuration of Webpack build processes. Initially created for Laravel PHP framework, Laravel Mix is used for both Laravel and stand-alone projects, as in this case. As the documentation states: "Laravel Mix is a clean layer on top of webpack to make the 80% use case laughably simple to execute". (Laravel Mix, 2020)

With Laravel Mix and Webpack the SASS files are compiled into *app.css* file in *dist* folder, which now contains all the styling code of the theme.

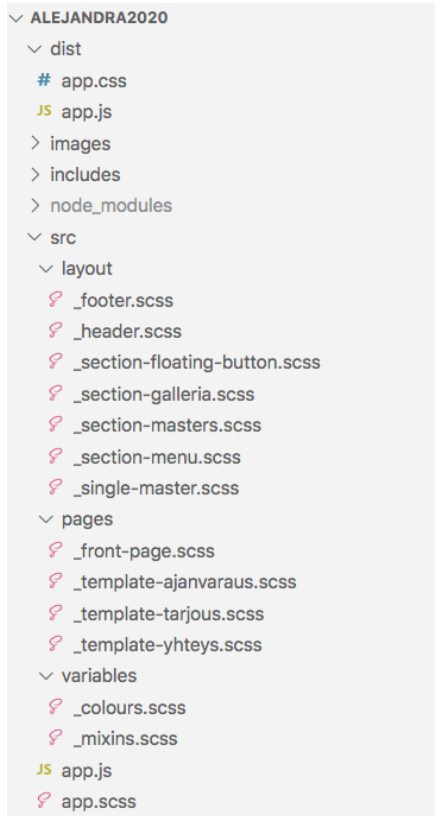


Figure 4. Hierarchy of the stylesheets

File structure overview

Comprehensive instructions on required theme files and their hierarchy are found in WordPress.org Theme Handbook.

The only two mandatory files are: the key template file *index.php* and the main stylesheet *style.css*. Template files define the site's layout and the way the content is displayed. They contain HTML mark-up, PHP code and built-in WordPress functions. For convenience and readability, a theme is usually composed of dedicated templates and reusable template partials, which are organized according to the WordPress hierarchy standards. WordPress goes down through the file hierarchy and calls matching template files. A template is recognized either based on the template name specified in the file comments or based on template naming rules. (WordPress.org, 2020)

Apart from the template hierarchy, the file structure depends on use of the custom posts and advanced content fields. Both are required to provide the future publisher with opportunity to edit and update the website's content.

In this project the following questions are taken into consideration when organizing the files.

1. Which page elements are reused for multiple pages?
2. What content the publisher needs to edit from the admin panel?
3. How to organize editable content so that the publisher can conveniently edit it using fewer admin tabs?

In this project the template hierarchy consists of the templates stored in the root folder and template partials stored in 'includes' folder. See Figure 5.

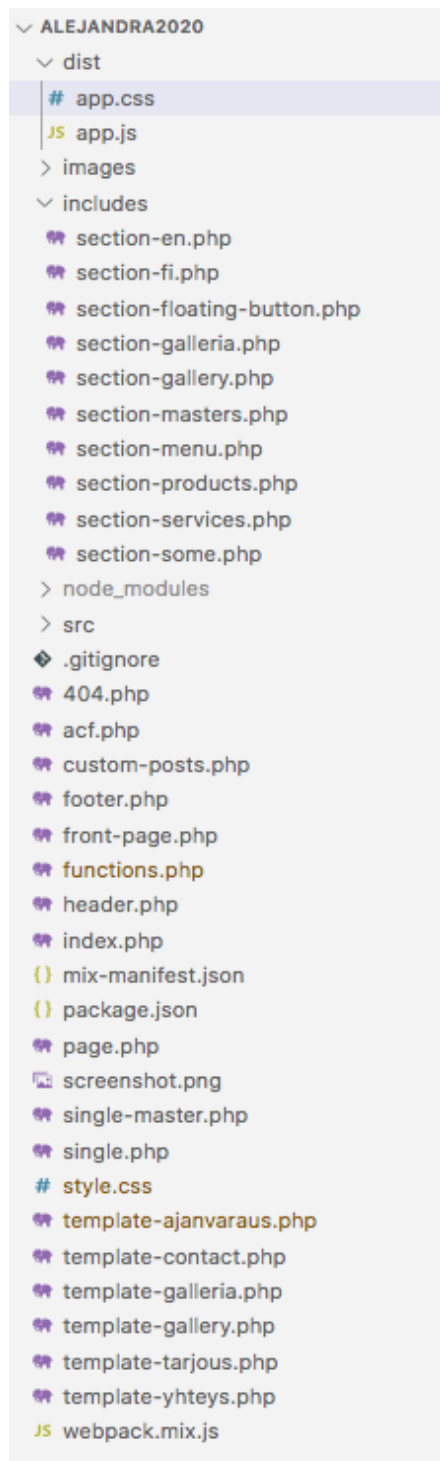


Figure 5. Hierarchy of the templates

The partials are imported into templates using WordPress function:

```
get_template_part( string $slug, string $name = null, array $args = array() )
```

```
template-galleria.php
You, seconds ago | 1 author (You)
1  <?php
2
3  /*
4  Template Name: Galleria
5  */
6  ?>
7
8  <?php get_header();?>
9
10 <div class="master">
11     <div class="img-page-container">
12         
13     </div>
14 </div>
15
16 <section class="master-info">
17     <h2 class="page-title"><?php the_title();?></h2>
18     <?php get_template_part('includes/section', 'galleria');?>
19 </section>
20
21 <?php get_template_part('includes/section', 'floating-button');?>
22 <?php get_footer();
23 ?>
24
25
26
```

Figure 6. Line 18: importing '-galleria' template partial into the 'Galleria' template

Each page of the website has a dedicated template, the same page template is used for Finnish and English versions of a page. All pages share the same footer and header sections, logo containing element is the same for all pages but for the front page.

A default page template is set using layout of Ajanvaraus/Reservation page. WordPress always uses file named page.php as a default page template. See Figure 7.

```
page.php
You, 2 months ago | 1 author (You)
1  <?php
2
3  You, 2 months ago • link fields added to services sections and to pho...
4  ?>
5
6  <?php get_header();?>
7
8  <div class="master">
9     <div class="img-page-container">
10         
11     </div>
12 </div>
13
14 <section class="master-varaus" id="master-varaus">
15     <h2 class="page-title"><?php the_title();?></h2>
16
17     <?php if (have_posts()) : while (have_posts()) : the_post();?>
18     <?php the_content();?>
19     <?php endwhile; endif;?>
20 </section>
21
22
23
24
25 <?php get_footer();
26 ?>
```

Figure 7. Default page template stored in page.php file.

Enabling content editing

In order to make certain parts of content available for editing by admin user in the future, the Custom Posts feature is used in combination with advanced custom fields.

In WordPress, various content is classified as different Post Types. The default Post Types are: Post, Page, Attachment, Revision and Navigation menu. (WordPress.org, 2020)

It is possible to create own content types using Custom Post Types. Advanced Custom Fields by ACF plugin makes it possible to create input fields for each piece of content editable from the admin panel.

Following the method suggested by Joe Casabona, a list of required editable features is created and associated with custom post types.

Layout section	Editable content	Custom Post Type
Front page: cards linking to Master Profiles	Name	
	Photo	
	Services list	
Master profile page	Link to Phorest booking service	Masters (fi/en)
	Summary for the card on profile page	
	About section	
	Products description	Services and Products (fi/en)
	Services description	
	Master's photo gallery	Photos (fi/en)
Gallery page	Common photo gallery	
Front page: the story section	Photo	Our Story (fi/en)
	Story text	
Contact page & contact info in menus	Facebook URL	All Contact info
	Instagram URL	
	Phone number	
	Address	
	Google maps link	
Offers page	Offer text	Offers (fi/en)
	Offer image	
	Offer link	

Custom Post types can be arranged in two ways:

1. Coded in as a part of the created theme in *functions.php* file. In this case the custom post types are integrated into the concrete theme.
2. In a separate plugin. In this case custom post types can be reused for multiple themes. (Casabona, 2012)

In this project the first approach is used. Custom post types are created within *custom-posts.php* file which is then imported into *functions.php* file.

The code below demonstrates creation of the Masters custom post type. The created type includes support for the Advanced Custom Fields plugin.

```
custom-posts.php
6 // Masters posts
7
8 function master_post_type() {
9
10 $args = array (
11 'labels'=> array (
12 'name' => 'Masters', // label shown in admin menu
13 'singular_name' => 'master', // name for one object of the post type
14
15 ),
16 'hierarchical' => true,
17 'public' => true, // visible both in the admin panel and front end
18 'has_archive' => true, // enables archiving of the custom posts
19 'menu_icon' => 'dashicons-universal-access-alt', // post type icon in admin menu
20 'rewrite' => array('slug' => 'masters','with_front' => false),
21 'supports' => array(
22 'title',
23 'editor',
24 'author',
25 'thumbnail',
26 'custom-fields',
27 'page-attributes',
28 'post-formats',
29 'custom-fields'), // features supported by the post type
30 'show_in_rest' => true, // for API endpoint
31 );
32
33 register_post_type('master', $args);
34 }
35
36 add_action('init', 'master_post_type'); // fires after loadng is completed but before any headers sent
37
```

Further, the corresponding Field Group is created from the WordPress admin panel, it contains two group-type fields with subfields which reflects the website's layout.

Advanced Custom Fields Field Groups Tools

Edit Field Group [Add New](#)

master fields

Order	Label	Name	Type
1	master card	master_card	Group
2	master profile	master_profile	Group

[+ Add Field](#)

Master Card group subfields:

Sub Fields

Order	Label	Name	Type
1	master name *	master_name	Text
2	master services	master_services	Text Area
3	master image	master_image	Image
4	summary	summary	Text Area

[+ Add Field](#)

Masters' profiles now can be edited from the admin panel:

The custom posts are fetched to the required templates using WP_Query – the WordPress Query class, and customized WordPress Loop. See the code below.

```

1  <?php
2  // Iterating though the Masters posts
3  $args = array(
4      'post_type' => 'master', // name of the post type
5      'post_status' => 'publish',
6      'posts_per_page' => -1, // displays all available posts on one page
7      // 'order' => 'ASC', // earliest on top
8  );
9
10 $the_query = new WP_Query( $args ); ?>
11
12 <?php if ( $the_query->have_posts() ) : ?>
13 <?php while ( $the_query->have_posts() ) : $the_query->the_post(); ?>
14 <?php $mastercard = get_field('master_card');?>
15
16 <a href="<?php echo the_permalink()?>" class="master-card">
17     <div class="pic-container">
18         " />
20     </div>
21     <h1 class="card-heading"><?php echo $mastercard['master_name'];?></h1>
22     <div class="grey-container">
23         <p><?php echo $mastercard['master_services'];?></p>
24         <?php
25         if ( function_exists( 'pll_current_language' ) ) {
26             $currentLanguage = pll_current_language();
27             if ( $currentLanguage == "fi" ) {?>
28                 <p class="varaa">Tutustu & Varaa &rsaquo;</p>
29                 <?php
30                 }
31             else { ?>
32                 <p class="varaa">Read more & Book &rsaquo;</p><?php
33                 }
34             }
35         ?>
36     </div>
37 </a>
38
39 <?php endwhile; ?>
40 <?php wp_reset_postdata(); ?>
41
42 <?php else : ?>
43 <p><?php _e( 'Sorry, no posts matched your criteria.' ); ?></p>
44 <?php endif; ?>
45

```

It is possible to style of the admin panel by adding CSS code within PHP function to the *functions.php* file. See the code below.

```

functions.php
32
33 // Styling the admin panel
34
35 function my_acf_admin_head() {
36     ?>
37     <style type="text/css">
38 > .acf-postbox {
39     }
40     }
41     .postbox-container .categorydiv {
42     background-color: #d1f9d8;
43     color: #000;
44     border-radius: 10px;
45     padding: 20px;
46     font-size: 15px;
47     }
48 > .acf-postbox .acf-field .acf-label {
49     }
50     }
51 > .acf-postbox .acf-field .acf-label label {
52     }
53     }
54 > .acf-postbox .acf-field .acf-label label {
55     }
56     }
57 > .acf-postbox .acf-field .acf-field-group {
58     }
59     }
60 > .acf-postbox .acf-field .acf-input {
61     }
62     }
63 > .postbox-header {
64     }
65     }
66 > .acf-postbox .acf-field .acf-input a.button {
67     }
68     }
69     }
70 </style>
71 <?php
72 }
73
74 add_action('acf/input/admin_head', 'my_acf_admin_head');
75

```

Mobile-first approach

The mobile-first approach is used for the project. Development starts with the smallest screen size, then gradually scales up to larger dimensions. 320px – 425px width is taken as the smallest screen width. Code-wise this is expressed in scaling up with help of min-width condition media queries, use of percentages and *rem* units instead of pixels for size variables, use of flexbox and grid containers. See Figures 8 and 9.

```

src > variables > _mixins.scss > ...
You, a month ago | 1 author (You)
1  /* Responsiveness */
2
3  $breakpoints: (
4  | mobile: 425px,
5  | tablet: 769px,
6  | laptop: 1024px,
7  | laptop-large: 1440px,
8  | desktop: 1800px,
9  );
10
11 @mixin max-media($keys...) {
12   @each $key in $keys {
13     @media (max-width: map-get($breakpoints, $key)) {
14       @content;
15     }
16   }
17 }
18
19 @mixin min-media($keys...) {
20   @each $key in $keys {
21     @media (min-width: map-get($breakpoints, $key)) {
22       @content;
23     }
24   }
25 }
26

```

Figure 8. SASS mixins for responsiveness

```

src > pages > _front-page.scss > ...
138
139 /* team section */
140
141 .team {
142   display: flex;
143   flex-direction: column;
144   justify-content: space-between;
145   align-items: center;
146
147   @include min-media(mobile) {
148     width: 90%;
149     margin: 5% auto;
150   }
151
152   @include min-media(tablet) {
153     width: 70%;
154   }
155
156   @include min-media(laptop) {
157     width: 50%;
158   }
159 }
160

```

Figure 9. Application of media query mixins, percentages and flexbox for adjusting the team section's presentation.

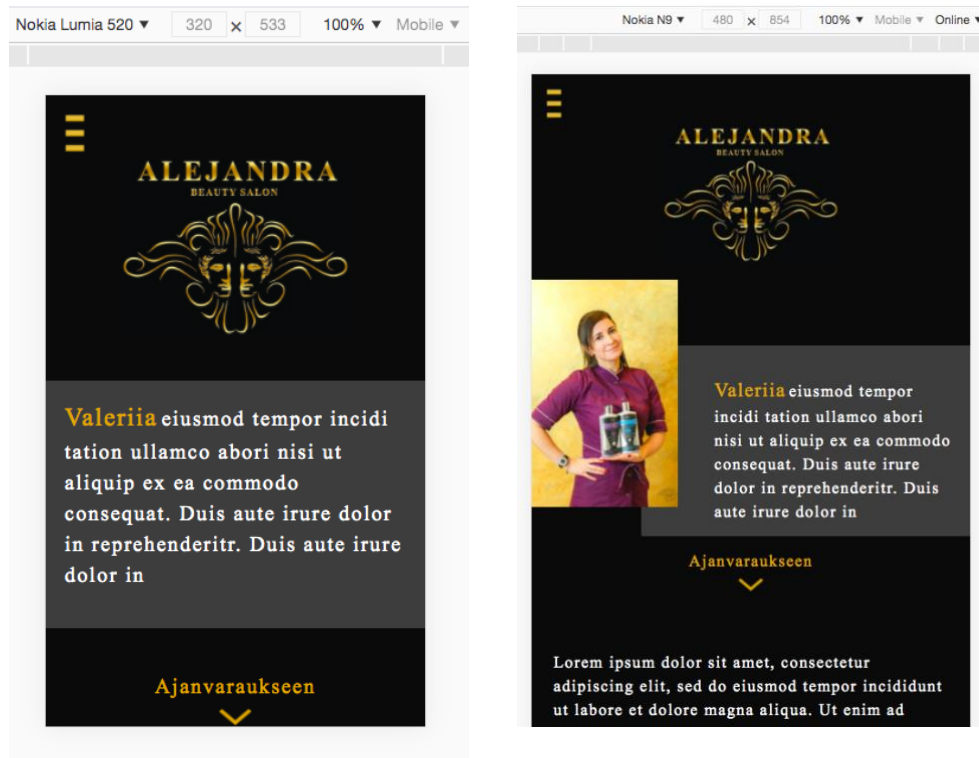


Figure 10. Progressive Enhancement principle applied to the master profile page.

Additionally, *Progressive Enhancement* principle is applied, which means adding more content types and elements as scaling up. For example, adding photo to the master page only at reaching min-width 426px. See Figure 10.

Staging environment

The staging version of the website was deployed to the subdomain of the main domain *alejandra.fi*.

Subdomains

<input type="checkbox"/>	Subdomain name	Type	IP/Directory/URL	
<input type="checkbox"/>	stage.alejandra.fi	Directory	htdocs/staging	Change

Figure 11. Subdomains in cPanel of the hosting and domain name provider

For this purpose, the current website with the current WordPress installation was cloned using the hosting cPanel.



Figure 12. Cloning WordPress installation in cPanel

Migration to Production Environment

The new website was transferred from staging subdomain to the main domain *alejandra.fi* by using control panel of the domain name and web hosting provider. The process was quite straightforward due to the user-friendly graphical interface.

Before the migration, both old website and the new staging version were backed up using UpdraftPlus - Backup/Restore plugin with backup files uploaded to local desktop. Further, the old version and staging version were transferred to subdomains. Both backup websites can be found online, but their indexing by search engines is discouraged.

After the migration, Secure Sockets Layer (SSL) certificate was installed, and the website traffic was redirected to HTTPS protocol. All the website's links were automatically changed from *http* to *https*.

SSL certificate adds the following advantages:

- due to the encryption methods, it enhances security of data transfer.
- it rises trustworthiness from the end-user perspective, especially when required to submit sensitive information, e.g., login credentials.
- it boosts Google SEO ranking and ensures smooth browsing experience; (Russel, 2020)

Security measures

The following security measures have been taken:

- 1) The theme was created following WordPress theme building standards

- 2) Following WordPress security recommendations on data sanitizing, all included third-party URLs are checked and cleaned with built-in WordPress function 'esc_url':

```
<iframe src=" <?php echo esc_url($masterprofile['booking_link']['url']);?>" frameborder="0"></iframe>
```

```
<a href="<?php echo esc_url($fb['url']);?>" target="_blank" rel="noopener noreferrer">
```

- 3) SSL certificate is installed
- 4) Only secured HTTPS links of Phorest application are used, HTTP links are no longer possible after SSL certificate installation
- 5) Non-used plugins are removed, all used ones are updated
- 6) WordPress Zero Spam plugin installed to block black-list IP senders in the contact form
- 7) PHP version was updated in hosting cPanel
- 8) A copy of website with a separate WordPress installation was saved in a subdomain to test possible updates before implementing changes in main domain website

Search Engine Optimization

In this case focus is on Google search engine as most widely used.

After new version of the website is moved to root domain replacing the old website, certain SEO issues arise. The new URL structure is different from the old version. This causes two problems: low SEO ranking of new URLs, and 404 error page displayed at old URLs retaining high SEO ranking.

It is not possible to influence search results directly and force Google to output new links and modified meta descriptions. There are, however, certain ways to alert search engines indirectly. Among them is modification of *robots.txt* file and new sitemap submission.

robots.txt is the file read by the search engine crawler to determine which pages to index and which not. In WordPress panel *robots.txt* can be changed from Yoast SEO plugin Tools tab. It is possible to disallow unwanted URLs or add a sitemap link. The author has added old non-existent frontpage permalinks to urge engine un-index them.

Tools - Yoast SEO

The Ultimate addons for Visual Composer plugin requires Visual

[« Back to Tools page](#)

robots.txt

Edit the content of your robots.txt:

```
User-agent: *
Disallow: /wp-admin/
Disallow: /beauty-salon/
Disallow: /kauneussalonki-helsinki/
Allow: /wp-admin/admin-ajax.php
```


Another available tool is Google Search Console. Here, one can inspect URL status and see whether and when a URL was crawled and indexed. It is also possible to alert Google about the changes in the website structure by submitting sitemap to the Sitemaps section.

The author submitted the new sitemap within a few days after the launch into production.

Sitemaps > https://www.alejandra.fi/sitemap_index.xml OPEN SITEMAP

Last read: **26/11/2020**

Total discovered URLs: **0** SEE INDEX COVERAGE

 Sitemap index processed successfully

Sitemaps read

Sitemap	Last read	Status	Discovered URLs
https://www.alejandra.fi/master-sitemap.xml	25 Nov 2020	Success	12
https://www.alejandra.fi/offer-sitemap.xml	25 Nov 2020	Success	2
https://www.alejandra.fi/page-sitemap.xml	25 Nov 2020	Success	10

It must be noted that the new website inevitably faces SEO issues if the structure of permalinks (site URLs) was changed. Re-indexing of new permalinks takes time and impacts SEO ranks. When an old established website with good ranking is replaced, it is desirable to preserve old permalinks for pages with similar content.

In our case, the old website's structure has been changed more than once. Some of the pages displayed in search results were outdated and already excluded from the old site's navigation. For this reason, keeping all URLs was considered inexpedient. The problem of non-existent 404 error pages in search output remains, and it will be monitored.

Yoast SEO plugin is a convenient tool to adjust site's SEO features from the admin panel. Among others, this includes:

- control over search appearance for different content types: posts, categories, pages, taxonomies.
- adjusting SEO titles, meta descriptions, page type identity, social media presentation for pages or posts.

Training for admin user

On November 19th, 2020, the training on use of the customized WordPress admin panel was held. It was an offline meeting, which took approximately three hours. Admin user was taught to add and edit the content using the custom post type fields of the modified admin panel. Preliminary to the training, a task-focused user manual was drafted. The manual is found in Appendix 2.

4 Evaluation

4.1 Heuristic evaluation of the user interface

Usability evaluation does not include the native functionality of the reservation forms since they are third-party provided.

Principle 1: Visibility of system status

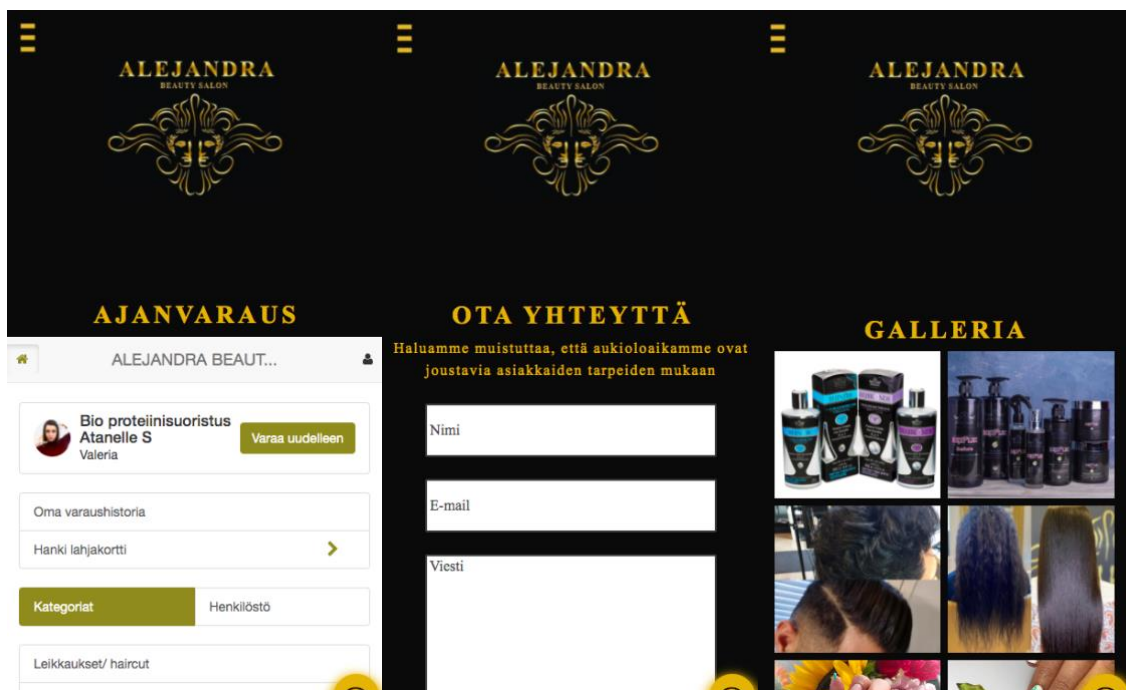
Positive findings

- 1) User knows what the current page is due to the following indicators:
 - In the main menu, the current item is underlined:







Current page is "Ota Yhteyttä"



- main navigation pages have corresponding headings ("Ota Yhteyttä", "Ajanvaraus", "Galleria"), which is especially important in the tablet and mobile versions featuring burger menu:



- at master pages, one can see the current master's picture and name highlighted in a contrasting color. The user is sure whose page was opened:

Etusivu Ajanvaraus Ota Yhteyttä Galleria 0401782801    Suomi  English

ALEJANDRA
BEAUTY SALON

Valeriia tekee hiuksistasi terveelliset ja kauniit vuoden ympäri

Ajanvaraukseen
↓

2) Contact form provides status messages:

OTA YHTEYTTÄ

Haluamme muistuttaa, että aukioloaikamme ovat joustavia asiakkaiden tarpeiden mukaan

ol

ol@ol

The e-mail address entered is invalid.

test

Lähetä

One or more fields have an error. Please check and try again.

OTA YHTEYTTÄ

Haluamme muistuttaa, että aukioloaikamme ovat joustavia asiakkaiden tarpeiden mukaan

Nimi

E-mail

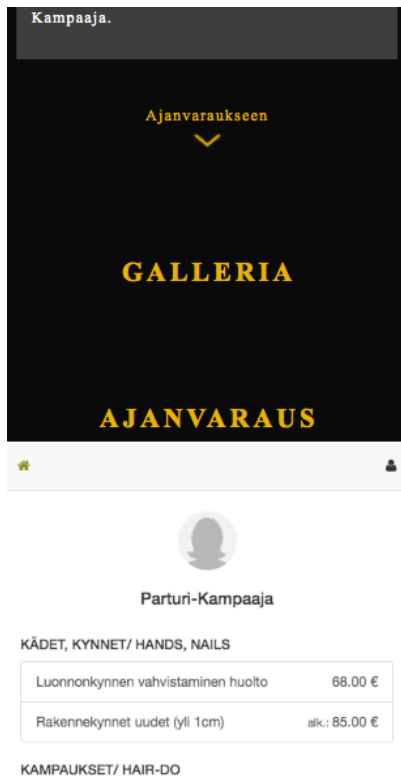
Viesti

Lähetä

Thank you for your message. It has been sent.

Negative findings

- 1) Website has no loading indicators or loading skeletons. Such feature would be especially useful in the Gallery page where images load slower than other elements.
- 2) Master pages could have breadcrumbs as a second-level navigation. Breadcrumbs would inform of a master page's position in the navigation hierarchy.
- 3) There is no content absence message in the Gallery section of master pages. In case no images are published, user cannot be sure if pictures are actually absent or just failed to load.



Principle 2: Match between system and the real world

Positive findings

- 1) Simple and familiar vocabulary is used for navigation purposes. User is clearly prompted to perform certain actions: “get to know us / tutustu meihin”, “contact us / ota yhteyttä”, “read more and book / tutustu ja varaa”, “book time / varaa aika”, “to booking / ajanvaraukseen”.

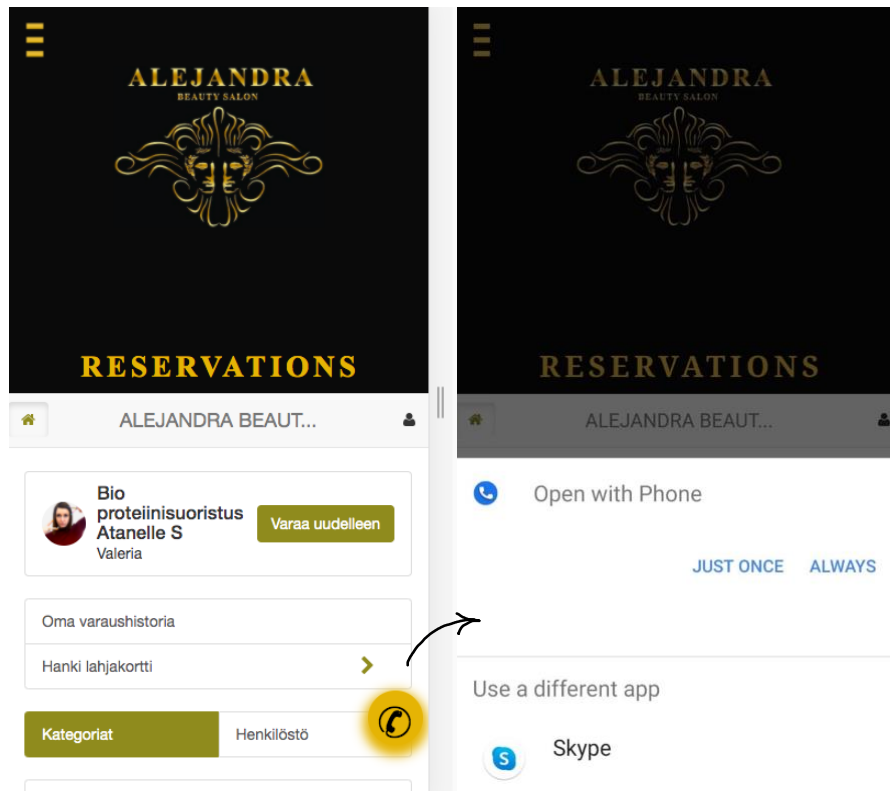
Negative findings

- 1) Professional terminology and brand names are found in the descriptions of services and products. Clients without previous knowledge about beauty brands, products, equipment and services may have hard time finding the needed one.

Principle 3: User control and freedom

Positive findings

- 1) In mobile version, when phone floating button is pressed, a call does not start immediately. First, user is offered calling method options. Then, user is taken to the dial view of the selected application to see required payment warnings, possibility of number corrections, etc. User has several opportunities to cancel actions or change the settings before starting actual call.



- 2) At master pages, long texts have option of “read more / read less” toggling, user controls the length of text to read.

Negative findings

- 1) Lightbox might additionally be closed with Escape key. Keyboard navigation to next and previous images would be useful.
- 2) In Lightboxes, the back-button effect is different from the X icon effect: back button takes user to the previous page instead of taking user back to the Gallery.
- 3) There is no scroll-to-top button even though some pages are quite long.

Principle 4: Consistency and standards

Positive findings

- 1) The website has two types of uniform look for buttons and anchors:
 - a yellow button with a hover effect



- a contrasting yellow text with an arrowhead icon



- 2) Uniform styling is used for elements of the same type, e.g., all headings are capitalized, have the same font size, color and letter spacing.
- 3) Pages have similar layout. Certain elements, such as logo, footer, header or margin of the main container, remain consistent across all pages.
- 4) Gallery sections have uniform look across the website, this is achieved through a reusable gallery component and a one-for-all Photos custom field.
- 5) At Home page, the list of masters is formed of cards with the same design pattern. Uniform design urges user to pay attention to the differences in the professions, names and services provided.

Negative findings

- 1) Since not all content is yet published, there are still mismatches between Finnish and English versions. For the same reason, master pages have different content scopes.

Principle 5: Error prevention

Negative findings

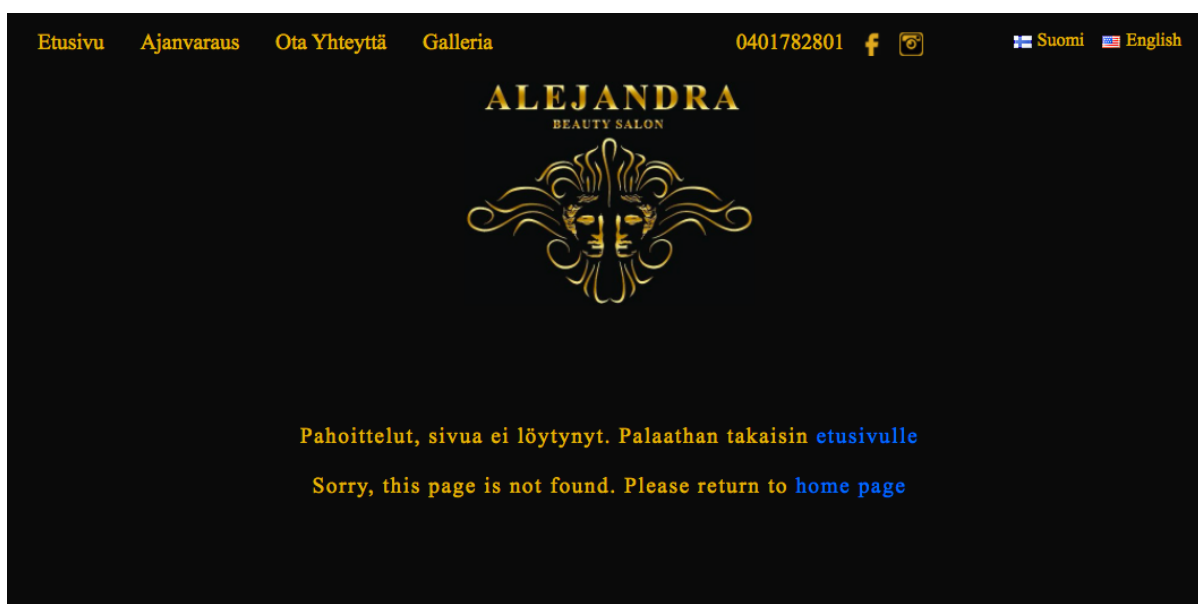
- 1) In the contact form, tooltips could be added to suggest correct actions and prevent mistakes.

Principle 6: Recognition rather than recall

Positive findings

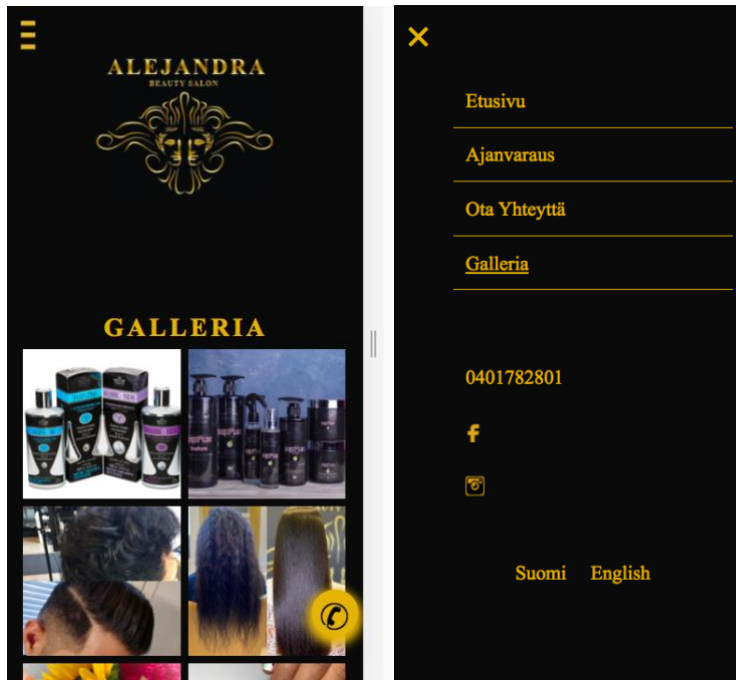
- 1) Contact and navigation information is repeatedly provided on the website, with account to various responsive versions. Therefore, user does not have to search for it.
 - Additional navigation menu is provided in the footer.
 - Salon's phone number is provided in several places. In all versions - in the footer; in the desktop version - in the header; in mobile and tablet versions - in the burger menu window: in the mobile version - in form of a floating phone call button. In all cases, a phone call is possible on click.
 - Social media links are provided in the header, footer and burger menu window.
 - Salon's address is found in the footer and at the Contact us page.

- 2) Upon a sudden landing at a non-existent 404 URL, a user gets clear instructions how to return to a valid Home page. This is a "help-in-context" example.



Negative findings

- 2) In the mobile and tablet versions, language switch possibility is hidden in the burger menu window. User has to search for it and memorize the location. It would be more user-friendly to place the language switch in the page header.



Principle 7: Flexibility and efficiency of use

Positive findings

- 1) The website offers multiple routes leading to reservation forms. A novice user can go to the common reservation form in Reservations page and study the whole range of services. Alternatively, user can go and book time at a specific master page. The latter option can serve as a shortcut for more experienced users knowing what exactly they are going to book. Optionally, a user can save a required master page to bookmarks and always open the exact page. E.g., client who visits salon for nail services, can directly open a nail technician's URL <https://www.alejandra.fi/master/zina/> with the corresponding reservation form.

Negative findings

- 1) The website does not provide possible customization options, such as changing the background theme from black to white.

Principle 8: Aesthetic and minimalist design

Positive findings

- 1) Content and visual design were planned in strict accordance with the project's business goals. Website has no extra features or random content.
- 2) The Offers page is optional and will be added to navigation only when Offers are there.

- 3) The information provided at website is essential and up to date. The admin user is able to remove any information as soon as it gets irrelevant.

Principle 9: Help users recognize, diagnose, and recover from errors

Positive findings

- 1) In the contact form, an error warning in red appears if a field was filled in incorrectly. The warning explains what exactly went wrong.

Principle 10: Help and documentation

This principle is not applicable to the website.

4.2 Usability testing of the user interface

Testing plan

The testing was performed after the ready website was migrated to the main domain. A remote synchronous moderated task-based test was conducted. Though performed on already launched website, the testing is regarded as formative and its results will be used for further improvements.

For remote testing the following video chat applications were used: Skype, MS Teams, Google Hangouts, Zoom. The participants shared the screen with opened Alejandra website. All tests were recorded from the author's computer using the Screen Recording mode. Laptop and tablet versions of the website were tested, including the option with the hamburger menu.

The testing plan included 11 tasks coined out according to the initial project goals with regard to UI redesign. (See Chapter 1). Task completion is evaluated as Pass/Fail. The tasks grouped by goals are listed below.

Goal 1: To improve navigation around the website, add clear routes to the reservation info, using call-to-action button and new menu:

Task 1: "Go to the Reservations page".

Task 2: "Go back to the Home page".

Goal 2: To provide detailed information about each salon's master, as well as clear navigation routes to master-specific reservation forms.

Task 3: “Find and give the name of the master who provides lash extension services”.

Task 4: “Find and tell the price for the classic lash extension”.

Task 5: “Find and give two facts about the background of the salon’s nail technician”.

Task 6: “Find photos of works of the salon’s cosmetologist”.

Goal 3: To provide detailed information about the salon, its location and contact details.

Task 7: “Find and give two facts about the salon”.

Task 8: “Find the opening hours of the salon”.

Task 9: “Tell the salon’s exact address”.

Task 10: “Go to the salon’s Facebook page”.

Task 11: “Send a message with text “test” using the salon’s contact form. Was it sent successfully?”.

Additionally, the participants had to answer the following three question upon the test completion:

Question 1: “Was it easy to navigate around the website? (1-5) (very hard – very easy)”.

Question 2: “Was it easy to find the reservation information? (1-5) (very hard – very easy)”.

Question 3: “Would you suggest any improvements?”

Participants

Six people aged 20 to 40 participated in the test. Some of them have IT-background.

Participant No	gender	IT background	Fluent Finnish	Device	Browser
1	F	software developer	no	Laptop	Chrome
2	M	no	no	Laptop	Chrome
3	F	software developer	no	Laptop	Firefox
4	F	UX design student	no	Laptop	Chrome
5	F	no	yes	Laptop	Chrome
6	M	no	yes	Tablet	Samsung Internet

All the participants speak English fluently. Four of six participants speak English better than Finnish. The latter allowed to test the language switching function, since the website’s default URL is Home page’s Finnish version.

Test results

Task	Participant No					
	1(en)	2(en)	3(en)	4(en)	5(fi)	6(fi)
1. Go to the Reservations page	Pass	Pass	Pass	Pass*	Pass	Pass
2. Go back to the Home page	Pass	Pass	Pass	Pass	Pass	Pass
3. Find and give the name of the master who provides lash extension services	Pass	Pass	Pass	Pass	Pass	Pass
4. Find and tell the price for the classic lash extension	Pass**	Pass**	Pass**	Fail	Pass	Pass
5. Find and give two facts about the background of the salon's nail technician	Pass	Pass	Pass	Pass	Pass	Pass
6. Find photos of works of the salon's cosmetologist	Pass	Pass	Pass	Pass	Pass	Pass
7. Find and give two facts about the salon	Pass	Pass	Pass	Pass	Pass	Pass
8. Find the opening hours of the salon	Fail	Fail	Fail	Fail	Fail	Fail
9. Tell the salon's exact address	Pass	Pass	Pass	Pass	Pass	Pass
10. Go to the salon's Facebook page	Pass	Pass	Pass	Pass	Pass	Pass
11. Send a message with text "test" using the salon's contact form. Was it sent successfully?	Pass	Pass	Pass	Pass	Pass	Pass
Total time spent on completion of all tasks (min)	8.10	7.50	9	8.10	8.20	9.30

*Participant 4 successfully navigated to the reservations page, but the reservation form failed to load.

**Passed with the author's suggest of the Finnish name of the service

(en) – English version was tested

(fi) – Finnish version was tested

After-test questionnaire results

Question	Participant No						Avg
	1	2	3	4	5	6	

1. Could you easily navigate around the website (1 – 5) (very hard – very easy)	5	4	5	4	4.5	3	4.25
2. Could you easily find the reservation information (1 – 5) (very hard – very easy)	5	5	5	3.5	5	5	4.75

Question 3. Would you suggest improvements?

Participant 1:

- In the contact form, placeholder text should vanish upon user’s typing in.
- Since it is hard to find opening hours, this information should be placed under a separate heading.
- There is too much empty space on top of Home page, the user would like to see more useful information in the header, e.g., address and opening hours. Otherwise, it takes to scroll all way down to bottom to see the address.

Participant 2:

- The pictures in the gallery are of low quality which produces bad impression about the salon.
- In the English version, the salon’s story contains very few valuable facts about the salon, only general description.
- It would be helpful to have English names of services in the reservation forms.

Participant 3:

- In the contact form, placeholders should vanish upon user’s typing in.
- After sending a message through contact form the user expects a pop-up window with status message. The existing status message displayed under the contact form is too small and hard to read.
- Opening hours are hard to find, this information should be stated clearly either at Contact page or in the footer.

Participant 4:

- In the comments section of gallery’s lightbox, there is too much empty white space.
- In the contact form, placeholders should vanish upon user’s typing in.
- The content of reservation forms does not load.

Participant 5:

- The participant finds the Home page layout and navigation confusing. The website has complex navigation combining main menu and separate routes from Home page to master pages. The Home page features a list of masters and services. Such layout and navigation

feel unfamiliar. The participant is not used to it, the participant has not seen such navigation and design before.

- It is hard to find opening hours: the participant was looking for numbers defining opening time, while the information is provided in wording. The participant expects opening hours at Home page.

Participant 6:

- The participant expects salon's story at Contact page. Currently, the story section is placed at the bottom of Home page and is hard to find.
- Opening hours are hard to find, this information should be placed on top of Home page.
- The participant tested tablet version with burger type menu. It is difficult for him to find navigation information. Burger sign has fixed position on page top, and it takes to scroll all way up to open the menu. It would be easier if the burger icon reappeared upon scroll up action.
- Not all master pages have full content. It would be good to include a story about each master.

Some favorable comments given by the participants on what they like about the website are listed below:

Participant 1:

- It is helpful that all masters are presented at Home page.
- Master's information, gallery and reservation form are all in one page, – this is convenient.
- It is good to see the phone number and social media links in the Home page header, i.e., right upon landing on the website.

Participant 4:

- Participant likes the color palette of the website and Home page design which follows the latest design trends.

Participant 5:

- It is very easy to navigate to reservation information due to multiple routes leading to it, including the call-to-action button on top of Home page and "Tutustu ja varaa" links leading to master-specific reservation forms. It is helpful that links are highlighted in a contrasting color.
- Despite unfamiliar layout of Home page, the participant still likes its look.

The author's observations:

- Lack of English service names in the reservation forms makes it difficult for English speaking users to find concrete reservation information.

RIPSET, KULMAT/ LASHES, BROWS

Ripsienpidennykset klassiset	69.00 €
------------------------------	---------

Ensimmäinen tutustuminen	30.00 €
--------------------------	---------

TEHOHOIDOT/ HAIR TREATMENTS

Proteiinihoito Atanelle S	100.00 €
---------------------------	----------

Proteiinihoito Atanelle M	140.00 €
---------------------------	----------

Proteiinihoito Atanelle L	180.00 €
---------------------------	----------

- It is easy to switch between languages, all participants managed it smoothly.
- When asked about the salon's facts, 5 of 6 participants went to Home page right away. This means, they expected to find salon's story at Home page, not at Contact page. Participant 3 started search from Contact page, then proceeded to Home page.
- When asked about opening hours, 4 participants started their search in the footer and header of Home page. Participants 1 and 4 started search from Contact page. All participants scrolled past the opening hours statement displayed above the contact form, but no one noticed it. This means, the statement is "invisible" to users. Some users said they took the statement for contact form instructions, and therefore skipped it.

The image shows a contact form on a black background. At the top, the text 'CONTACT US' is written in large, bold, yellow capital letters. Below this, a smaller yellow line of text reads: 'We would like to remind you that our opening hours are flexible according to customer needs'. The form consists of three white input fields: 'Name', 'E-mail', and 'Message'. At the bottom center, there is a yellow button with the text 'Send' in black.

- Though master pages are not in the main navigation menu, all participants understood very quickly where to look for master-specific information. They all confidently returned to Home page to find a master in question.
- Though contact form's status message is small, all the participants noticed and read it easily.



- Four participants used menu link to navigate to Reservations page, two participants used call-to-action “Book time” button for the same task.

4.3 User interface usability errors correction

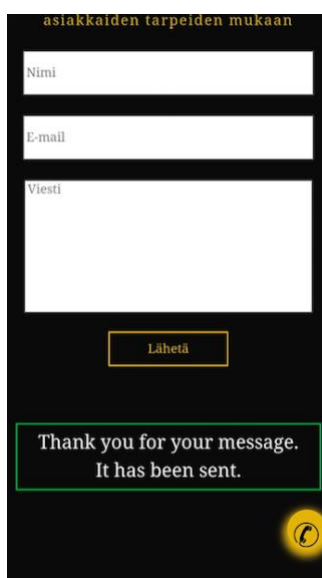
Usability testing and heuristic evaluation revealed a number of drawbacks. Quick actions should be taken on critical errors and errors that are easy to fix. The following compound list of usability errors was formed based on these criteria.

Error 1: In contact form, placeholders not vanishing upon typing.

The error has been fixed. Placeholders worked incorrectly since the author used a standard HTML syntax instead of the plugin specific syntax.

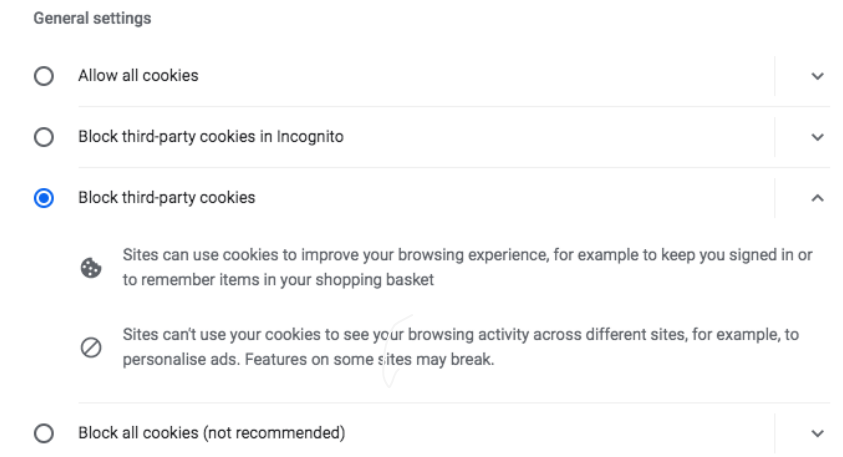
Error 2: In contact form, status message is too small and hard to read.

Font-size is increased, a different color is applied for higher contrast.



Error 3: For one of the test participants the reservations forms failed to load in Chrome browser.

Further investigation revealed the following. The console error in the participant's browser indicated that access was denied to Phorest documents. The failure was caused by Chrome settings blocking third-party cookies.



The issue was fixed by changing browser settings. Nevertheless, the problem is regarded as severe. Reservation is one of the main functions of the website. An average user can be unaware of the browser settings.

Currently, third-party Phorest reservation service is used as embedded into Alejandra website with iFrame. The author embedded the service following Phorest instructions. Further study showed that Phorest does not recommend using iFrame method for Chrome and Safari browsers from 2018 onwards.

Possible redesign options include direct navigation to Phorest page or adding Phorest link as an alternative. The issue will be discussed with the customer.

Error 4: Opening hours are hard to find.

Expediency of changes will be discussed with the customer. The salon has flexible opening hours, it is not possible to provide a standard table of numbers as users expect. A user can see available time slots when making reservation. Nevertheless, general working hours may be important for a client making a phone call.

Error 5: The content-related issues include lack of information on individual masters, lack of images in master galleries, mismatches between language versions, lack of English service names in the reservation forms.

The customer is aware of the content issues, more content will be published with time.

4.4 Usability testing of the admin panel

The testing was performed one month after the admin training. Test included one participant – the actual admin user of the website. A synchronous moderated task-based test was conducted remotely using Skype. The participant shared screen view. Test was recorded from the author’s laptop using Screen Recording.

The test consisted of 16 tasks and was evaluated as Pass/Fail/Pass with prompts. Test plan covered most of the editing action types. The participant did not use the admin manual when completing the test.

Tasks:

1. Change the email address to the right one in the Finnish version contact form.
2. Add a new master (name: Nina, services: Kynsiteknikko, image: pick any from the media tab).
3. Move Nina to the bottom of the front page.
4. Remove Nina from the front page.
5. Add a photo to Elvira’s gallery, including featured image.
6. Move the photo to the middle of Elvira’s gallery.
7. Add one photo to both common gallery and Zina’s gallery, including featured image.
8. Remove the photo.
9. Add new product to Zina’s page. (name: Bohemia, main text: “Bohemia tuote”).
10. Remove Bohemia product from Zina’s page.
11. Add Instagram URL to the Contact information.
12. Add salon’s photo to Home page.
13. Add Offers post (name: Joulutarjous, text: “-10 % kaikista tuotteista”, photo: Joulutarjous.jpg.).
14. Add Tarjoukset page to the header menu Finnish version.
15. Remove the Tarjoukset page from the menu.

Test results

Task no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Result	Fail	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass with prompts	Pass	Pass	Fail	Pass with prompts	Pass with prompts	Pass

Comments:

- 1) One typical problem was repeated: participant attempted to edit page from the corresponding page section. Meanwhile, editing is custom-post-based: custom posts should be created or updated to modify the content. Failure to complete task 12 (adding

salon's photo to Home page) was caused by the same problem: the participant was looking to modify Home page directly while salon's photo is part of 'Our Story' custom post type.

- 2) Since the participant worked at tasks without looking into the admin user manual, it took longer to find the right custom post. In real life, the participant can always check the manual.
- 3) Tasks 12 and 13 was related to Offers/Tarjoukset page. The participant could complete the task with prompts only. Editing Offers page was not included in the training conducted earlier, the participant attempted such tasks for the first time.
- 4) Failure in task 1 was caused by not following the syntax required by the contact form plugin.

Most of the tasks were completed independently, and the test can be considered successful. Based on the test results the user manual was corrected and supplemented with additional instructions.

It is not possible to change admin logics from custom-post-based to page-based. Page-based approach would better comply with the heuristic principle "Match between system and the real world". In this case, real world is the website's user interface. Custom-post-based approach requires more efforts and memory resources from the admin user. Improved and detailed user manual with task-based step-by-step instructions is supposed to mitigate the problem.

Following the 'content-driven design" concept described in section 2.2 hereof, it must be noted that the content of the website does not require very frequent updates. Unlike a blog or media resource with continuously published posts and articles, a business website is only updated once in a while.

4.5 Evaluation of the selected development approach

Classic WordPress CMS with own theme developed from scratch was selected as a development approach in this project. Upon the project's completion the approach is evaluated against the criteria defined earlier in Table 1, Chapter 2.

Criterion 1: Performance speed

Speed of the ready website was analysed using PageSpeed Insights by Google. The tool makes it possible to check both mobile and desktop version page by page. Speed is scored from 1 to

100, where 1-49 is low, 50-89 is average, and 90-100 – fast. Results are presented in the table below.

Page URL	Speed (1-100)	
	Mobile	Desktop
http://alejandra.fi/	66	84
https://www.alejandra.fi/en/home/	72	88
https://www.alejandra.fi/ajanvaraus/	31	87
https://www.alejandra.fi/en/reservations/	43	92
https://www.alejandra.fi/ota-yhteytta/	70	97
https://www.alejandra.fi/en/contact-us/	71	97
https://www.alejandra.fi/galleria/	74	84
https://www.alejandra.fi/en/gallery/	94	87
https://www.alejandra.fi/master/valeriia/	62	94
https://www.alejandra.fi/en/master/valeriia-en/	56	95
https://www.alejandra.fi/master/elvira/	51	93
https://www.alejandra.fi/en/master/elvira-en/	56	84

Comments on the testing results:

1. In most cases images load slower than other elements.
2. Slow speed is shown for mobile version of Reservation page with the third-party reservation form.
3. The website works faster on desktop in 100% of cases.
4. No essential differences between language versions

PageSpeed Insights gives the following recommendation to increase speed:

1. Changing image format
2. Use images of suitable size.
3. Reduce number of WordPress plugins to avoid loading unneeded JavaScript code.
4. Improve cache policy of static assets by increasing their cache storing time.
5. Impact of third-party code should be reduced by using asynchronous loading, lazy-loading approach for embedded resources, or by avoiding third-party code. The problem is caused by use of third party scripts, such as Google Analytics script, Bootstrap CDN, embedded iFrame components.

It must be noted that during the usability testing, users did not complain about the speed. All successful tasks were completed relatively fast. For most pages, speed testing gives satisfactory results. Certain improvements are still possible within the selected development approach, such as image format correction.

Most of interactive functionality is a third-party provided, in particular: Phorest reservation form, contact form plugin, and embedded Google maps map. As test has shown, third-party code slows down page loading.

Criterion 2: Content management

WordPress Custom Posts with Advance Custom Fields (ACF) plugin are used to enable content editing. Since admin user cannot modify page directly, such functions as page live preview, page layout changes are not always available. Use of WYSIWYG (What You See IS What You GeT) editor turned out undesirable since it causes unwanted modifications in the theme mark-up. Still, admin user has freedom to use such WP admin functions as adding/deleting full pages and page components, to modify navigation menu, to create different content for separate language versions, to manipulate posts, their displaying order and status.

Custom-post-based content editing can hardly be called intuitive and user-friendly. This was confirmed by usability testing. Without sufficient practice and user manual, it takes time to find required post type.

On the other hand, custom posts with ACF fields are very convenient tool for a developer. With the native WP functions, data queries are made by custom field parameters, which ensures clear and readable code. ACF website provides good detailed documentation for PHP developers, which significantly accelerates development.

Despite the mentioned usability issues, content editing is enabled. Admin user can edit all content types and layout components as was initially planned.

Criterion 3: Availability of WP plugins

Some UI features were implemented fast due to available ready-made plugins. These include contact form (Contact Form 7 plugin) and multilanguage pages (Polylang plugin). Yoast SEO plugin simplifies SEO actions. WordPress Zero Spam plugin is blocking black list IP trying to send spam via the contact form. Post Types Order plugin allows admin user to shuffle the posts with easy drag-and-drop action. UpdraftPlus - Backup/Restore plugins creates backups is a variety of forms. ACF plugin enables creating of custom fields.

The following functional features were encoded in JavaScript as part of the theme: image gallery lightbox, image slider, mobile burger menu, read more/read less toggling.

Criterion 4: Security

A problem of increased spamming through the contact form was faced after website's launch. No other security problems have been revealed so far.

Criterion 5: SEO

Despite the out-of-the-box SEO and convenient Yoast SEO plugin, not all SEO problems could be solved. The problem of 404 error pages remains. The author had to use Google Search Console as an additional SEO tool.

Criterion 6: Content fetching

As already stated in Content management section, data querying is easy due to the custom fields and WordPress native query functions.

Criterion 7: Scalability

The current website can be scaled up in two ways. New page layouts and elements, color palettes and design patterns can be added by further developing of Alejandra2020 theme. A new theme version will need to be uploaded again to WordPress platform.

An alternative is to grow website using admin panel. Admin user can create new pages using existing page templates. It is possible to set any page as a Homepage with required settings. Admin user can manipulate navigation menus. Theme files can be directly modified in Theme Editor tab of admin panel. The latter, however, requires web development skills.

To enable further design customization from admin panel, developer would have to add Theme Options and Widgets to the theme.

Criterion 7: Freedom of styling

The chosen way to develop own theme from scratch provides full freedom to embody own design. The developer can create page layouts in form of dedicated template files and layout components in form of reusable template partials. There are no restrictions in applying CSS. With Laravel Mix library, it is possible to use convenient SASS syntax. The desirable design solutions were implemented without any technical obstacles.

In addition to the preliminary defined criteria, several more conditions should be noted.

Suitable hosting services

Alejandra website is using self-hosted WordPress.org platform in combination with a shared hosting including a WordPress installation. Switching to a React app would require changing a hosting plan or hosting provider, since the current plan does not support React.js applications.

Cost-efficiency of development tools

The new Alejandra website was created using free development tools only. The author utilized free software and managed to avoid payable plans and tiers. Costs of hosting and domain name are not taken into account. As a rule, cost efficient development approach is essential for small- and medium-scale business.

5 Discussion

The project's problem was in developing a dynamic updatable website with a new unique design. It was assumed that developing a new WordPress theme and using WordPress dashboard as admin panel would provide best possible results. The hypothesis implied that development process is easy, and all business requirements can be fulfilled.

Testing of the idea through the project implementation revealed pros and cons of the approach.

All design solutions were implemented as they were conceived. The new website has a brand-relevant look. All required features and components are added. User can navigate smoothly around the website following the navigation routes as initially planned. During the UI usability testing, participants successfully completed most of the tasks. The website pages demonstrate satisfactory performance speed.

The resulting WordPress theme is dynamic and allows for content editing. All required content types are updatable.

The development process is quite easy and fast due to detailed WordPress documentation, predefined PHP functions, availability of asset compilation tools, CSS pre-processor, ready-made plugins.

Cons of the developments approach include the following.

Customisation of admin panel based on custom post types is confusing for a novice admin user. As proved by the usability testing, it takes time to find required post type, i.e., admin interface is not intuitive.

To scale up the website, Alejandra2020 theme should be further developed. Design customisation without coding is limited.

Search Engine Optimisation of a WordPress website is not as straightforward as the author assumed based on the literature review. A SEO specialist is required to solve all SEO issues of the website.

As a conclusion, developing of new WordPress theme is suitable for already existing WordPress website. This is a good solution if the website must be preserved at WordPress platform, whereas being totally redesigned. For a brand-new website, other CMS platforms could be considered.

For example, API-first Contentful CMS presents a valuable alternative to WordPress as a modern product featuring a user-friendly interface with a fresh feeling.

6 References

- Andersson, Ö. (2012) *Experiment!: Planning, Implementing and Interpreting*. Wiley.
- Aronson, D. (2006) *Evidence-Based Technical Analysis: Applying the Scientific Method and Statistical Inference to Trading Signals*. Wiley.
- Barker, D. (2016) *Web Content Management*. O'Reilly Media, Inc.
- Barnum, C. M. (2010) *Usability Testing Essentials*. Morgan Kaufmann.
- Budiu, R. (2017) *Quantitative vs. Qualitative Usability Testing*.
- Casabona, J. (2012) *Building WordPress Themes from Scratch*. Rockable Press.
- Cole, A. et al. (2010) *Build Your Own Wicked Wordpress Themes*. SitePoint Pty. Ltd.
- figma.com (2020). Available at: <https://www.figma.com/> (Accessed: 4 December 2020).
- Gooding, S. (2015) *State of the Word 2015: JavaScript and API-driven Interfaces Are the Future of WordPress*. Available at: <https://wptavern.com/state-of-the-word-2015-javascript-and-api-driven-interfaces-are-the-future-of-wordpress> (Accessed: 18 November 2020).
- Helander, A. (2015) *Building a WordPress site for Sports Academy of Kouvola*. Haaga-Helia UAS. Available at: https://www.theseus.fi/bitstream/handle/10024/93999/Thesis_Helander_FINAL.pdf?sequence=1&isAllowed=y.
- Heslop, B. (2018) *History of Content Management Systems and Rise of Headless CMS*. Available at: <https://www.contentstack.com/blog/all-about-headless/content-management-systems-history-and-headless-cms/> (Accessed: 18 November 2020).
- Krishnaswamy, K. N., Sivakumar, A. I. and Mathirajan, M. (2006) *Management Research Methodology: Integration of Principles, Methods and Techniques*. Pearson India.
- Krofegha, B. (2020) 'Building A Web App With Headless CMS And React', *smashingmagazine.com*.
- Laravel Mix (2020) *Laravel Mix. Basic Example*. Available at: <https://laravel-mix.com/docs/5.0/basic-example> (Accessed: 19 October 2020).
- Laubheimer, P. (2020) *3 Persona Types: Lightweight, Qualitative, and Statistical*. Available at: <https://www.nngroup.com/articles/persona-types/> (Accessed: 1 December 2020).
- MacDonald, M. (2020) *WordPress: The Missing Manual, 3rd Edition*. O'Reilly Media, Inc.
- Maioli, L. (2018) *Fixing Bad UX Designs*. Packt Publishing.
- Messenlehner, B. and Coleman, J. (2019) *Building Web Apps with WordPress, 2nd Edition*. O'Reilly Media, Inc.
- Mioni, P. (2019) 'Vue.js And SEO: How To Optimize Reactive Websites For Search Engines And Bots', *smashingmagazine.com*. Available at: <https://www.smashingmagazine.com/2019/05/vue-js-seo-reactive-websites-search-engines-bots/>.
- Moran, K. (2020) *Benchmarking UX: Tracking Metrics*.
- Nielsen, J. (2020) *10 Usability Heuristics for User Interface Design*. Available at:

<https://www.nngroup.com/articles/ten-usability-heuristics/> (Accessed: 29 November 2020).

Nielsen, J. and Loranger, H. (2006) *Prioritizing Web Usability*. New Riders.

PHP Documentation (2020).

Prabhu, A. (2015) *Beginning CSS Preprocessors: With SASS, Compass.js, and Less.js*. Apress.

Ramaswamy, V. and Kerimcan, O. (2014) *The Co-Creation Paradigm*. Stanford University Press.

React Documentation (2020). Available at: reactjs.org (Accessed: 18 November 2020).

Rohrer, C. (2014) *When to Use Which User-Experience Research Methods*.

Russel, A. (2020) *What is HTTPS?*

Schade, A. (2013) *Competitive Usability Evaluations: Learning from Your Competition*. Available at: <https://www.nngroup.com/articles/competitive-usability-evaluations/> (Accessed: 30 November 2020).

Scott Jr., E. A. (2020) *SPA Design and Architecture: Understanding single-page web applications*. Manning Publications.

Toh, W. S. (2020) *What is WAMP LAMP MAMP XAMPP (Plus Their Differences)*. Available at: <https://code-boxx.com/difference-wamp-lamp-mamp-xampp/> (Accessed: 19 October 2020).

W3Techs (2020). Available at: <https://w3techs.com> (Accessed: 18 November 2020).

Walliman, N. (2011) *Research Methods. The Basics*.

Webpack (no date) *GitHub Repository*. Available at: <https://github.com/webpack/webpack> (Accessed: 19 October 2020).

Winkels, N. (2019) *The Definitive Guide to CMS Architecture*. Available at: <https://developers.bloomreach.com/blog/2019/cms-architecture.html#coupled> (Accessed: 18 November 2020).

WordPress.org (2020) *Theme Handbook*. Available at: <https://developer.wordpress.org/themes/getting-started/what-is-a-theme/#required-files> (Accessed: 19 October 2020).

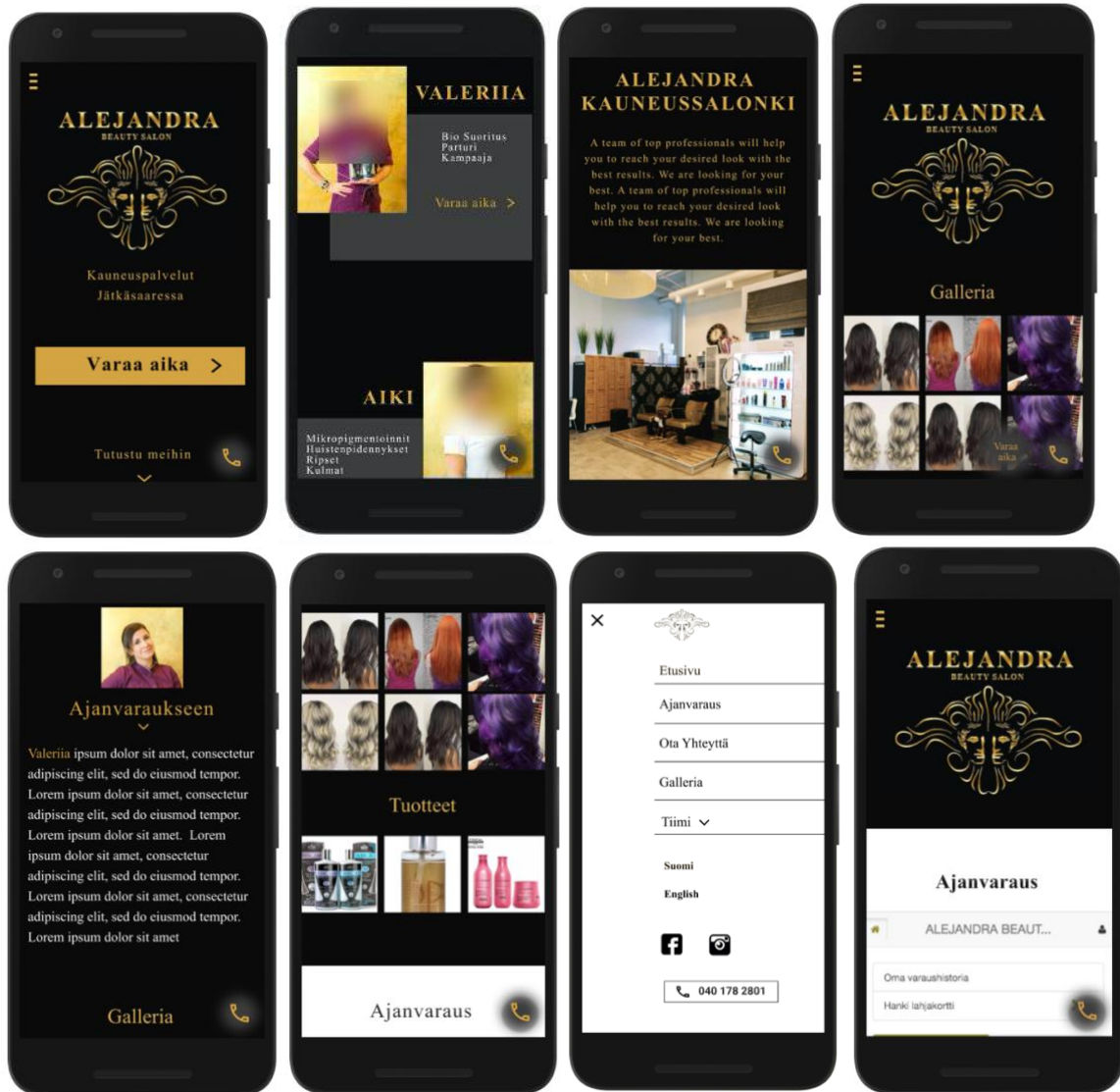
WordPress Codex (2020) *WordPress Codex. Theme Development*. Available at: http://codex.wordpress.org/Theme_Development (Accessed: 19 October 2020).

WordPress Documentation (2020). Available at: <https://wordpress.org/> (Accessed: 18 November 2020).

Appendices

Appendix 1. Semi-functional prototype, mobile version

[View in Figma](#)

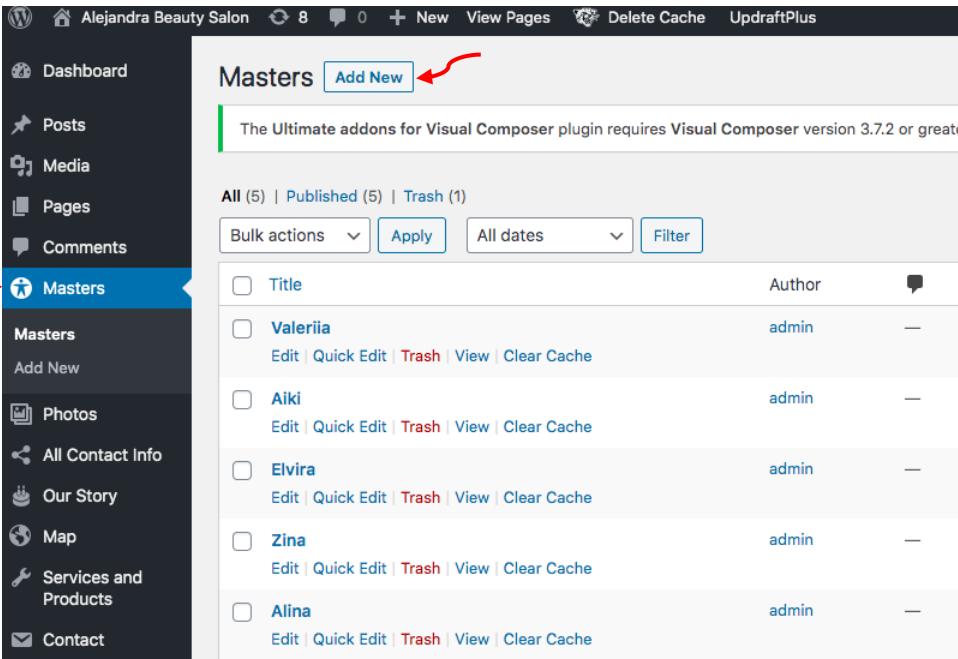


Appendix 2. Admin User Manual

Admin User Manual for alejandra.fi

Task 1: Adding a master to the front page and add a master page

- 1) Go to **Masters** tab, press **Add New**



The screenshot shows the WordPress admin interface for 'Alejandra Beauty Salon'. The top navigation bar includes 'New', 'View Pages', 'Delete Cache', and 'UpdraftPlus'. The left sidebar menu is visible, with 'Masters' highlighted in blue and a red arrow pointing to it. The main content area shows the 'Masters' tab with an 'Add New' button highlighted by a red arrow. Below the button, there is a notification: 'The Ultimate addons for Visual Composer plugin requires Visual Composer version 3.7.2 or greater'. The main content area displays a list of 5 published Masters posts. Each post has a checkbox, a title, an author name, and a trash icon. The posts are: Valeriia, Aiki, Elvira, Zina, and Alina, all authored by 'admin'. Each post has links for 'Edit', 'Quick Edit', 'Trash', 'View', and 'Clear Cache'.

<input type="checkbox"/>	Title	Author	
<input type="checkbox"/>	Valeriia	admin	—
<input type="checkbox"/>	Aiki	admin	—
<input type="checkbox"/>	Elvira	admin	—
<input type="checkbox"/>	Zina	admin	—
<input type="checkbox"/>	Alina	admin	—

- 2) Fill in all fields. **About** field is optional. Check existing Masters posts for an example.

MANDATORY!!! The same as the master name !

Name of the master, one word

Each service from new line !

Press to add image from media files

Short summary about the master for the grey container, maximum 25 words ! Use span tag to highlight a word in gold: `Valeriia`

Press to add link to the master's page on Phorest site. Without the link a master's page will have no booking section.

Information shown on the master's page right under the grey container

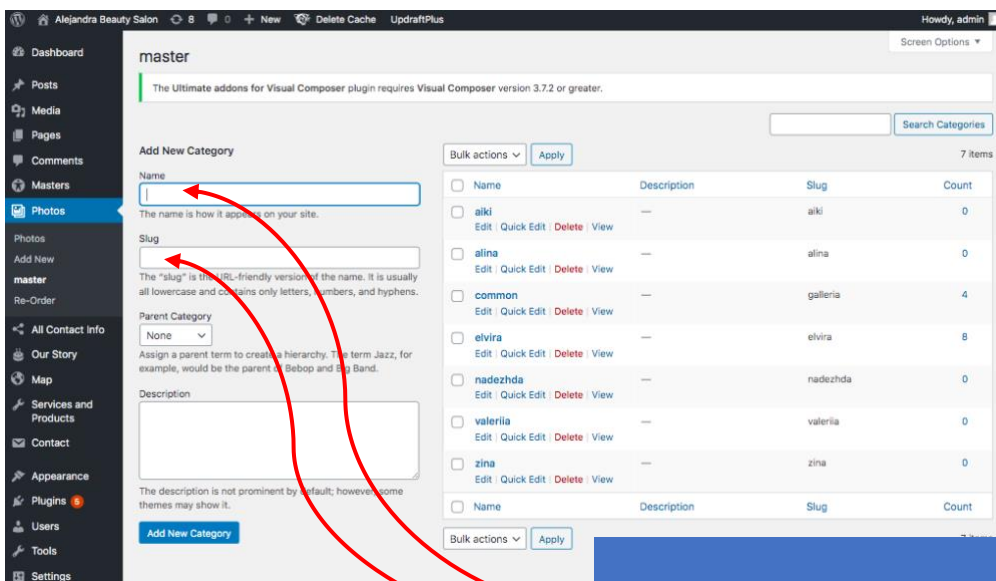
- 3) You can press **Preview** button to see the changes on the Master's individual page to check: summary, booking section, about section. The changes on the Front page are not shown in the Preview.
- 4) Press **Publish** button.
- 5) If needed, shuffle the masters to be shown in the required order on front page. Shuffling is done by dragging entries one by one up and down within the list.

The order of entries in admin panel reflects the order of the masters on front page

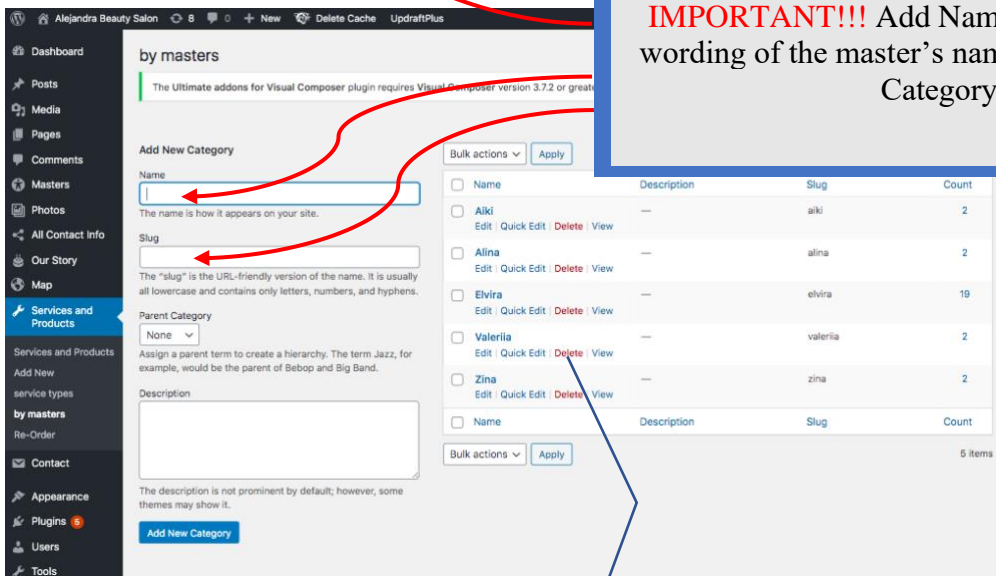
To delete a master, press **Trash** from the masters' list.

After adding or deleting a master, you should add/delete corresponding master category field to:

- Photos posts -> master;
- Services and Products -> by masters



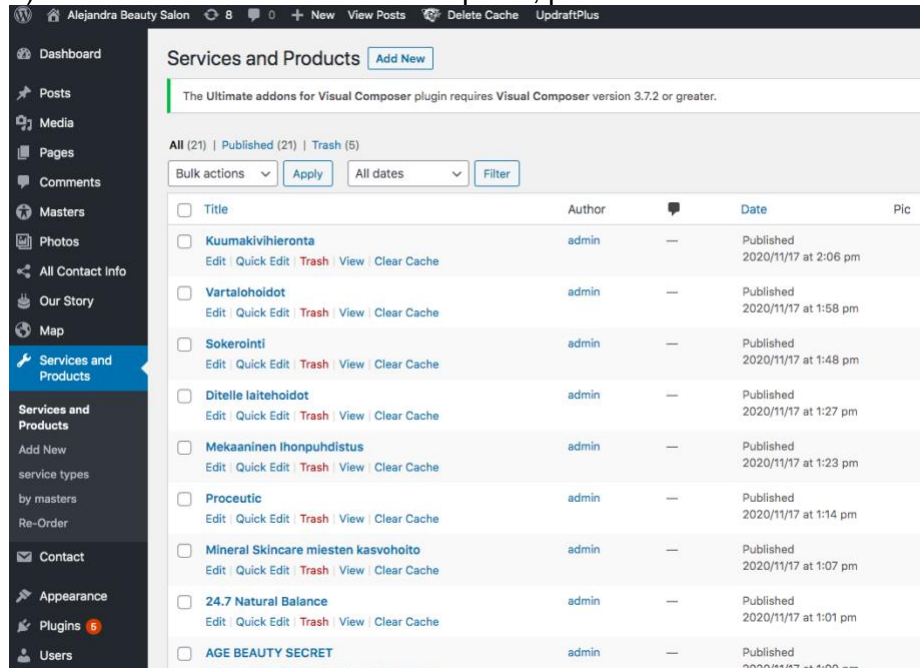
IMPORTANT!!! Add Name and Slug as exact wording of the master's name! Press "Add New Category"



Press **Delete** to remove master's category. Remove a master's category only if the corresponding master is deleted from the website.

Task 2: Add/edit information about products and services for each master

1) Go to **Services and Products** posts, press **Add New**



2) Fill in the fields

Title can be any, it is only shown inside the admin panel

Name of the service, the names will be automatically transformed into capital letters of gold color. You can give several names one under the other

MANDATORY!!! Check out the box for the right section type (service for Palvelut, product for Tuotteet), and a box with master name for the right master's page. You can check out several boxes. If no box is marked, the post will not be published anywhere.

The text shown on page before the "Lue lisää" cut

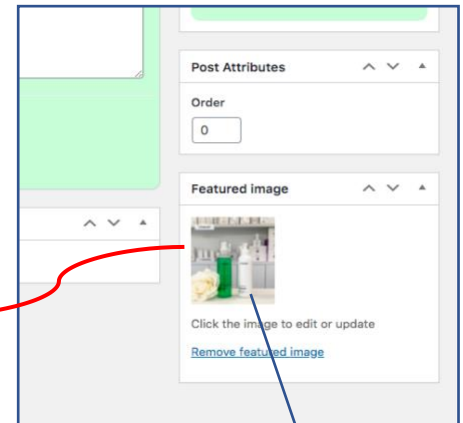
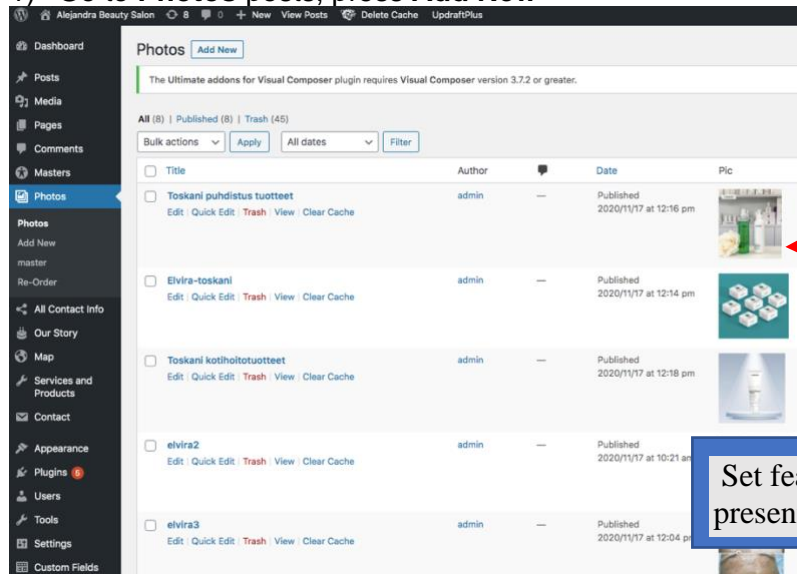
You can give a link to a third-party website, e.g. a product's manufacture. This is optional

The text that opens on page when "Lue lisää" is clicked. This is optional.

- 3) All the posts about all services and products for all masters are stored in one list. Posts can be shuffled the same way as Masters posts.
- 4) Each master's page can have many posts on services and products.
- 5) Click **Publish** to publish the post, or **Update** if the post was edited.
- 6) To delete a post, press **Trash** from the posts list.

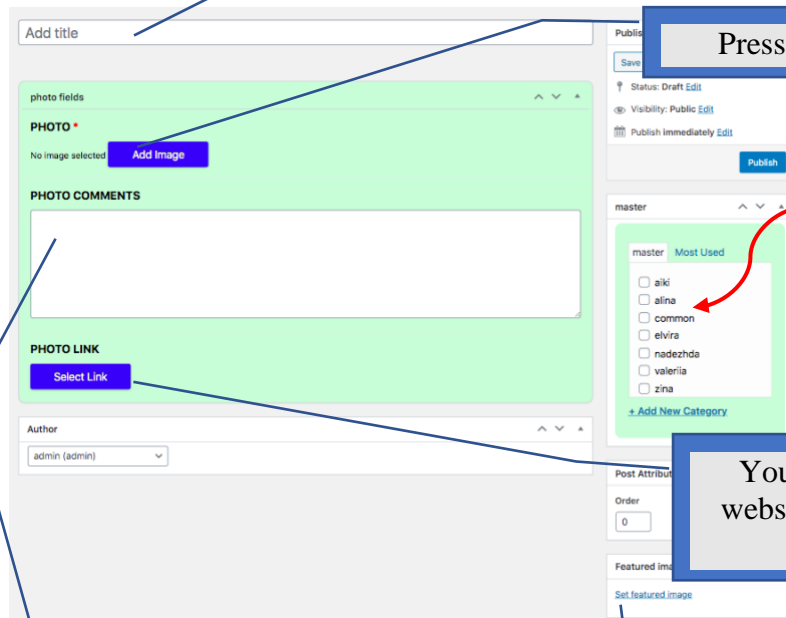
Task 3: Adding/editing a photo to a gallery.

1) Go to **Photos posts, press Add New**



Set featured image for clearer presentation of the entry in the post list

2) Fill in the fields



Title can be any, it is only shown inside the admin panel

Press to add image from media files

MANDATORY!!! Check out the box for the right gallery type (master name for corresponding master's gallery, 'common' for the Galleria page). You can check out several boxes. If no box is marked, the photo post

You can give a link to a third-party website, e.g. a product's manufacture. This is optional

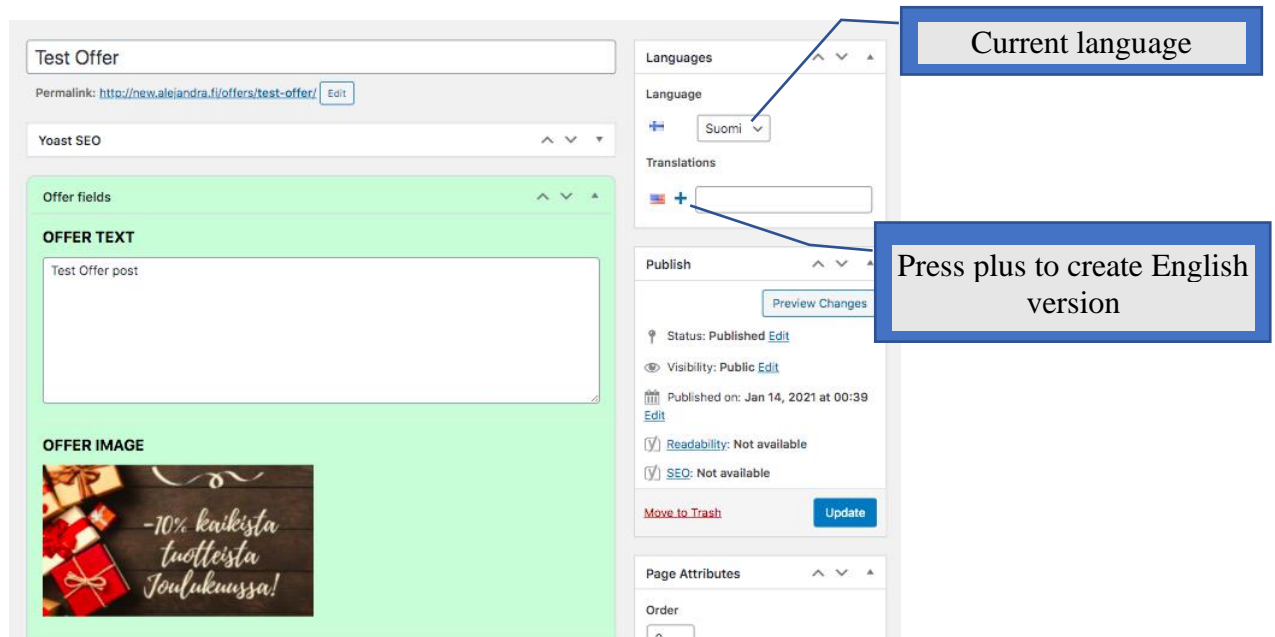
Comments to the photo are displayed in the lightbox, comments can be of any length, the scrollbar is switched on in case of overflow

Set featured image for clearer presentation of the entry in the post list

- 3) All the photos for all masters and for the Galleria page are stored in one list. Posts can be shuffled the same way as Masters posts.
- 4) Click **Publish** to publish the post, or **Update** if the post was edited.
- 5) To delete a photo post, press **Trash** from the posts list.

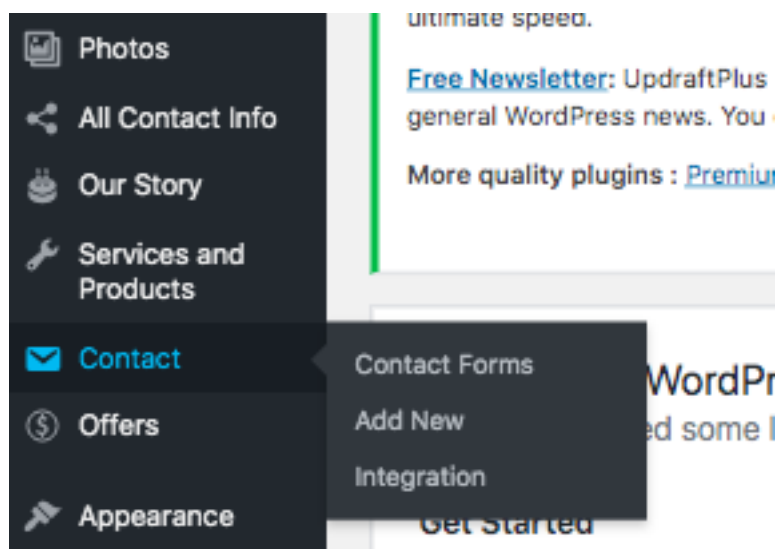
Task 4: Creating multilanguage posts and pages

- 1) When a post of any type or page is created, it belongs by default to the Finnish version of the website. To create its English version, press plus sign near the US flag icon.



Task 5: Change email in the contact form

- 1) Go to contact form by opening Contact section



- 2) Select English or Finnish contact form, then open **Mail** tab. Change the email preserving the existing syntax (do not delete the angle brackets!!!!)

yhveys lomake fi

Copy this shortcode and paste it into your post, page, or text widget content:

```
[contact-form-7 id="1496" title="yhveys lomake fi"]
```

◀ keys switch panels

Form **Mail** Messages Additional Settings

Mail

You can edit the mail template here. For details, see [Setting up mail](#).
In the following fields, you can use these mail-tags:
[your-name] [your-email] [your-message]

To:

From: [your-name] <[redacted]>

Subject: Alejandra Beauty Salon [your-name]

Task 5: Add Offer/Tarjous post and include Offers/Tarjoukset page to the navigation menus.

- 1) Go to **Offers** post, press **Add new**

Dashboard

Posts

Media

Pages

Comments

Masters

Photos

All Contact Info

Our Story

Services and Products

Contact

Offers

Offers

Add New

Offers [Add New](#)

The Ultimate addons for Visual Composer plugin requires Visual Composer vers

All (0) | Cornerstone content (0)

All SEO Scores All Readability Scores [Filter](#)

<input type="checkbox"/>	Title	Author	Date
No pages found.			
<input type="checkbox"/>	Title	Author	Date

- 2) Fill in the post fields, pay attention that title of the posts will be displayed as Offer's title at the Offers page.
- 3) Go to **Appearance** => **Menus**, select the menu to change, check out required page in Pages drop down section, press **Add to Menu**, then **Save Menu**.

Select a menu to edit: or [create a new menu](#). Don't forget to save your changes!

Add menu items

Language switcher

Pages

Most Recent

- Contact us
- Galleria
- Gallery
- Home — Front Page
- Offers
- Ota Yhteyttä
- Reservations
- Tarjoukset

Select All

Posts

Masters

Photos

All Contact Info

Our Story

Menu structure

Menu Name

Drag the items into the order you prefer. Click the arrow on the right of the item to reveal additional configuration options.

-
-
-
-

Menu Settings

Auto add pages Automatically add new top-level pages to this menu

Display location Header Menu Suomi

Header Menu English (Currently set to: header menu en)

Footer Menu Suomi (Currently set to: footer menu fi)

Footer Menu English (Currently set to: footer menu en)

[Delete Menu](#)

- 4) Finnish version of the pages called Tarjoukset belongs to Finnish menu types (header and footer), English version called Offers belongs to English menus.