



**SAVONIA**

# Utility of a Web Site

**Alexander Malysenko**

Bachelor's Thesis

---

**Valitse kohde.**

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Alexander Malysenko			
Title of Thesis Utility of a Web site			
Date	28 February 2017	Pages/Appendices	42
Supervisor(s) Mr Keijo Kuosmanen, Lecturer			
Client Organisation/Partners A&N Palvelut Ay			
<p>Abstract</p> <p>It is not a secret that web development and web design play the leading roles in all the network projects. The web site of a company is a business card that represents it and its quality can bring the customer in or frighten him off. Utility and usability are two concepts that help to build a web site that will be the best for the user.</p> <p>Because of that, the purpose of the thesis was to study utility, usability and all the parts of these principles. Also, how the research-based theoretical base can help to improve web design of a web site to make it more efficient and useful. To begin with, all the theory was collected and studied. Secondly, the existing web site was developed using the theoretical base, including methods, principles, and techniques that are used in web development process.</p> <p>Finally, some new functions were added to the web site during the practical part using the PHP programming as well as the analysis of user activity was done. Future plans of the client company were announced as well.</p>			

Keywords

Web, Usability, Utility, Readability, PHP, Database, Web Constructor

## CONTENTS

1	INTRODUCTION .....	5
2	PRODUCT WEBSITE SPECIFICATIONS .....	7
2.1	Requirements to a web-constructor .....	7
2.2	First results.....	<b>Error! Bookmark not defined.</b>
2.3	Website before work.....	9
2.4	Analysis and goals to achieve .....	14
3	THEORETICAL BASE .....	16
3.1	Website working process .....	16
3.2	Readability of text .....	18
3.3	Hierarchy in text.....	21
3.4	Contrast.....	22
3.5	Adding multimedia .....	23
3.6	Design of website with good usability .....	24
4	PRACTICAL PART .....	25
4.1	Changes in website outlook.....	25
4.2	Programming task.....	25
4.3	Making of MySQL databsfe .....	27
4.4	Calendar code .....	28
4.5	User interface .....	29
4.6	Future plans .....	30
5	CONCLUSION .....	32
	REFERENCES .....	33
	ATTACHMENTS .....	34

## 1 INTRODUCTION

Nowadays, web design and web development are very important components of all projects located inside or outside the network, oriented to a mass client and to a small group of clients. Often, thanks to successful development, the website of a company is a business card that represents it in the best light and from the first look brings the customer in. All the information that he needs to know about the product or service is right before his eyes but in the environment of growing consumer market and serious competition there – only successful marketing and an interesting, bright cover can make the alleged customers stop exactly on this product. Mostly, this applies to young and small companies that have just started their development and need to find their first clients but companies with an established client base also need to improve their services for getting more clients.

At present, web developers, no matter if they are professionals or beginners, can use a huge amount of services able to support the creation of a web page, website or even a group of websites quickly and with high-quality. Regardless of the professional level, knowledge or experience, almost every beginner can make a product that is not inferior to a product made by professionals who have worked as a web designer for years. Of course, there are some traditional methods of making websites and HTML pages as Aloha Editor or Dreamweaver CS5 for html5, Thimble or Notepad++ for HTML and CSS and many other great products. But now web constructors or web builders are more and more popular basically because it is not necessary to know how to work with them at all to start doing something.

While studying in Finland, a lot of webpages which were not oriented to sending to the web were created during university courses. Mostly they were made only for achieving certain lesson goals and there were no websites that would really be on the web and their goal would be selling some product or service and developing that brand. Also, a practical training was made in a local company in Kuopio that is focused on a client from abroad and working in a narrow segment – medicine. At the time of that practice, the duties for a web developer position were a constant control and troubleshooting, contacting clients, adding, editing information or removing the old content.

All the work there began with a simple website without any special features or impressing layouts but during the time of that practical training, a company team made a full reorganization, redesign and corrected many mistakes and defects. That was the first practice in web-development with a real product for a real customer who bought that

product afterwards. At the same time, a web constructor was used for the first time as a framework for the project.

After a while, the author's own small company was started and it was based on interests and things that each member of the family wanted to do. If in the previous project there was no such enthusiasm about the final product because it was not something each member of a team was fully involved in and that was only a practical training, now the project started as a private one and started from zero there was a motivation to work with 100% efficiency. So after the registration of a collective enterprise, one logical question had to be solved: how will the potential clients get information about the services that the company offers? A public group on Facebook was a good decision but what if somebody does not use social medias? The answer was suggesting itself because one's own website is something special and individual. It can present business and what the client can get from the company to an everyone who comes there.

## 2 PRODUCT WEBSITE SPECIFICATIONS

### 2.1 Requirements to a web-constructor

There were a lot of difficulties with the choosing of a method of building of a website but finally, wix.com platform was chosen because it has friendly design and convenient and simple interface. The main requirements to a web-constructor were:

- the rapidity of creating and editing a website
- a wide choose of templates
- simplicity of the interface
- a built-in system for contacting clients
- an ability to connect the international domain

The wix.com platform suited all of these requirements. It is also quite popular around the web and has a big database with instructions for many situations related to the decorating of the website or filling it up with the content (Wix team, 2006).

The .fi domain was chosen mostly because the client company is registered in Finland and all the work will be done in Finland.

The requirements to the website design were as follow:

- it should be a website with one main page and few secondary pages
- the website should be as short and informative as possible so the client will not get lost in useless information
- the website should be done in one main style without any edgy and disturbing elements or elements that would be hard to read or to look on
- the client should be able to see all the main information about the company and services on the main page. All the contacts and feedback forms should be also on the main page
- the secondary pages will reveal all services separately with a description, prices, and examples.

### 2.2 First results

During the working process, some of these points were corrected and some examples of this work were moved to an additional page "Gallery". Also, the website was connected to the Facebook group so users could easily go both ways: if they saw a company group they could go directly to the website and vice versa.

Decorating of a website was very important and the most important were how simple the website is for users and how easy it is for them to understand website structure because according to Gutenberg eye tracking research (Nielsen, 2006) work mostly people do not want to spend a lot of time searching for some information on the website. According to the diagram shown in Figure 1 the user will not read all the information but he will just quickly look through the page. That is the reason why all the information needs to be placed densely but be informative enough and also there should be some multimedia and animation so the user will come off the reading and relax (Bradley, 2010).

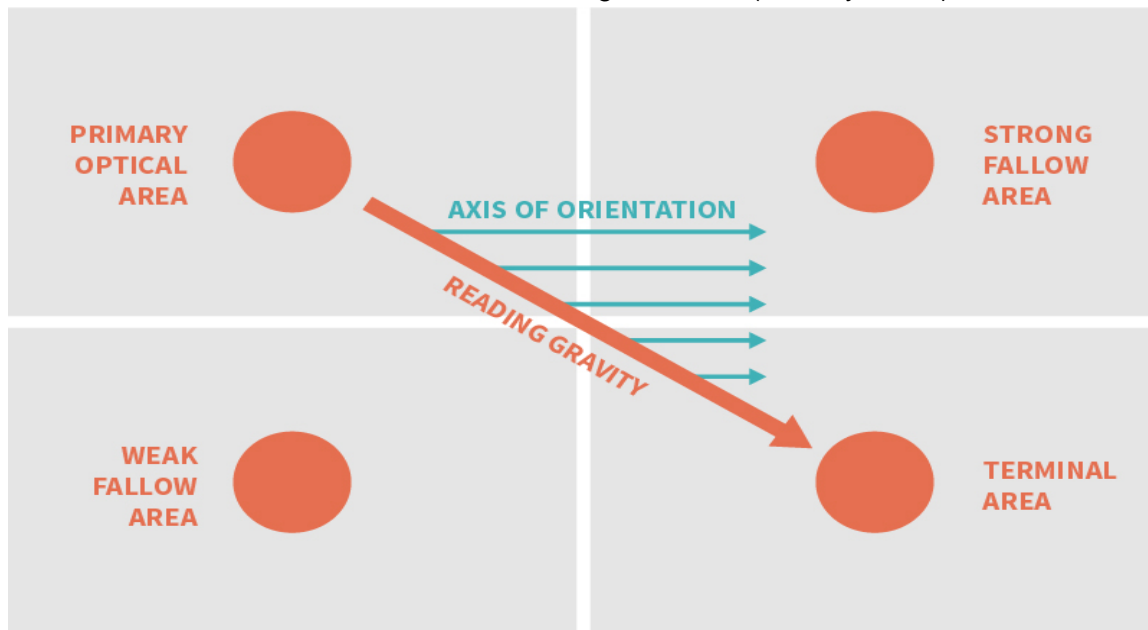


FIGURE 1. The Gutenberg Diagram also known as Diagonal Balance

At the time of writing thesis the website was already published and filled up with a content but the main goal of the work was a process of design optimization, the convenience of using and the website utility. Some analytic programs were used for user behaviour check and analysis, user preferences and device monitoring.

Within the first two weeks it has became clear that almost half of the clients use mobile devices so the website had to be optimized for them as well but due to the fact that the web-constructor has an automatic adaptation function it makes the whole process a lot easier.

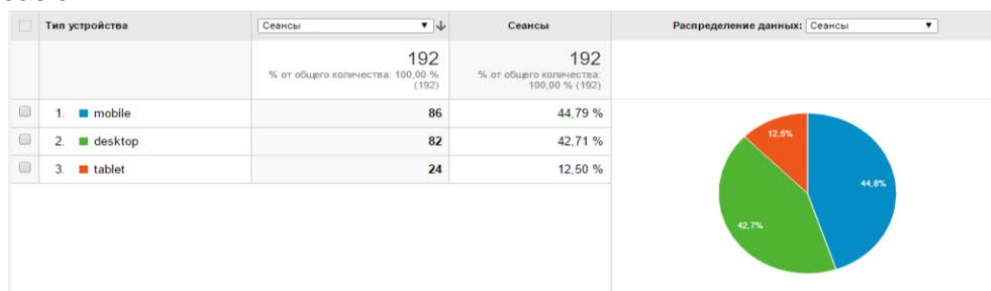


FIGURE 2. Analysis of devices used within first two weeks by website users



Figure 2 here represents the results of the first two weeks period when the website was published. 192 users have visited the website in total and 86 (44,79%) from them have used mobile phones, 24 (12,5%) have used tablets and 82 (42,71%) have used desktops.

### 2.3 Website before work

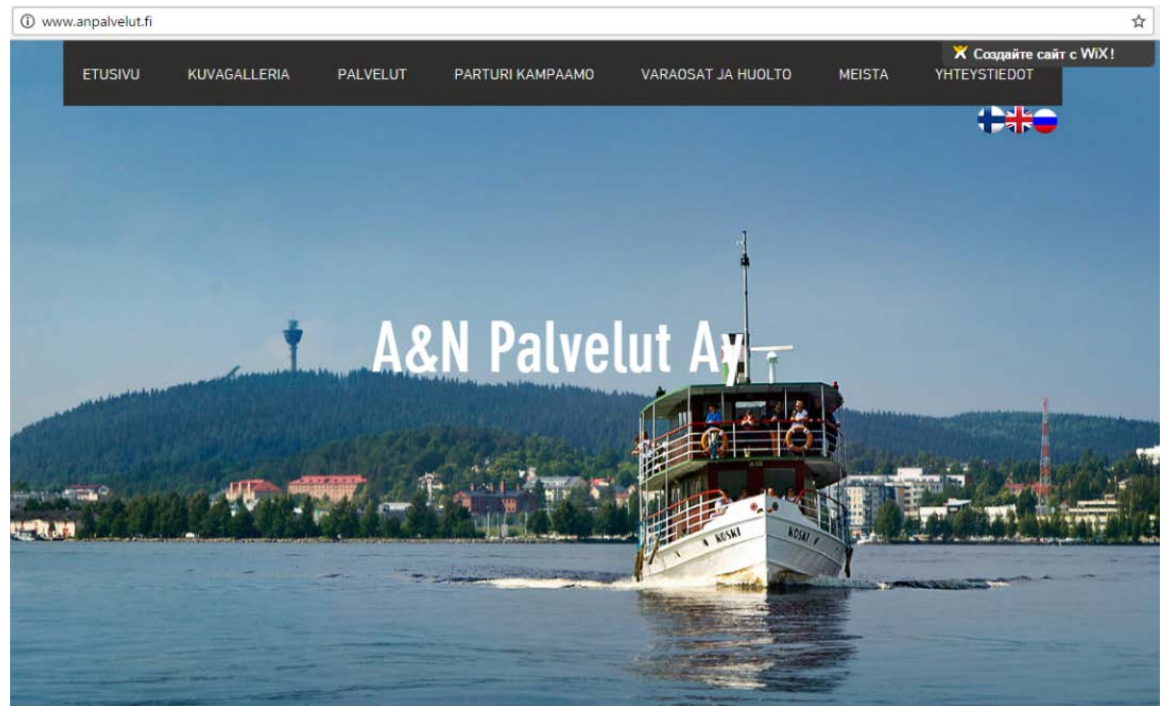


FIGURE 3. Main page of the website

The main page is the longest at the moment as shown in Figure 3. After entering the website, the user sees only a background picture, the header with a menu and the name of the company so the main attention is on this. If the client looks on the header he will be able to imagine a map of a website but it is not the main element of the page. These colors were chosen so there will be nothing that can disturb the vision. The main emphasize is on the name of the company as mentioned above but it is made with the help of the font size and central placement.

One more salient element is the language panel: it is easy to find it because it is colorful and placed in a more or less common part of the page.

Scrolling down the page the user sees a part of the page where all the services are shown with a short pop-up animation. This part of the page is shown in Figure 4. There are three elements chosen to describe services in a traditional way: scissors for the hairdressing studio, wrenches for the workshop and saw for furniture and interior elements production.

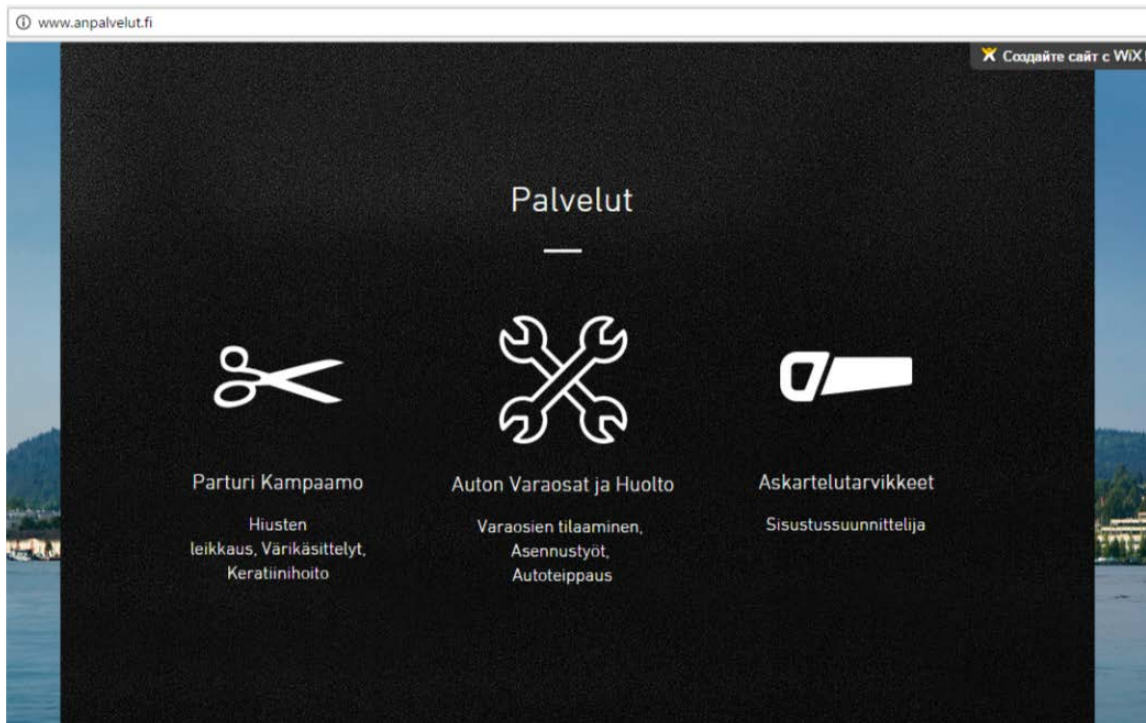


FIGURE 4. Service part of the page

That part of the page is the main one and this is the reason why the contrast between dark background color and white symbols and text is the most visible one. It has to show to the client what exactly he or she can get from the company in few words but at the same time they have to describe all services fully. Symbols here are the biggest elements on the page. For descriptions a small but well-readable font size is used and for headings - the size is a bit bigger so a client will see the difference between them. Besides, there are small spaces between headings and descriptions and it also helps to make these borders between the parts of the page more visible.

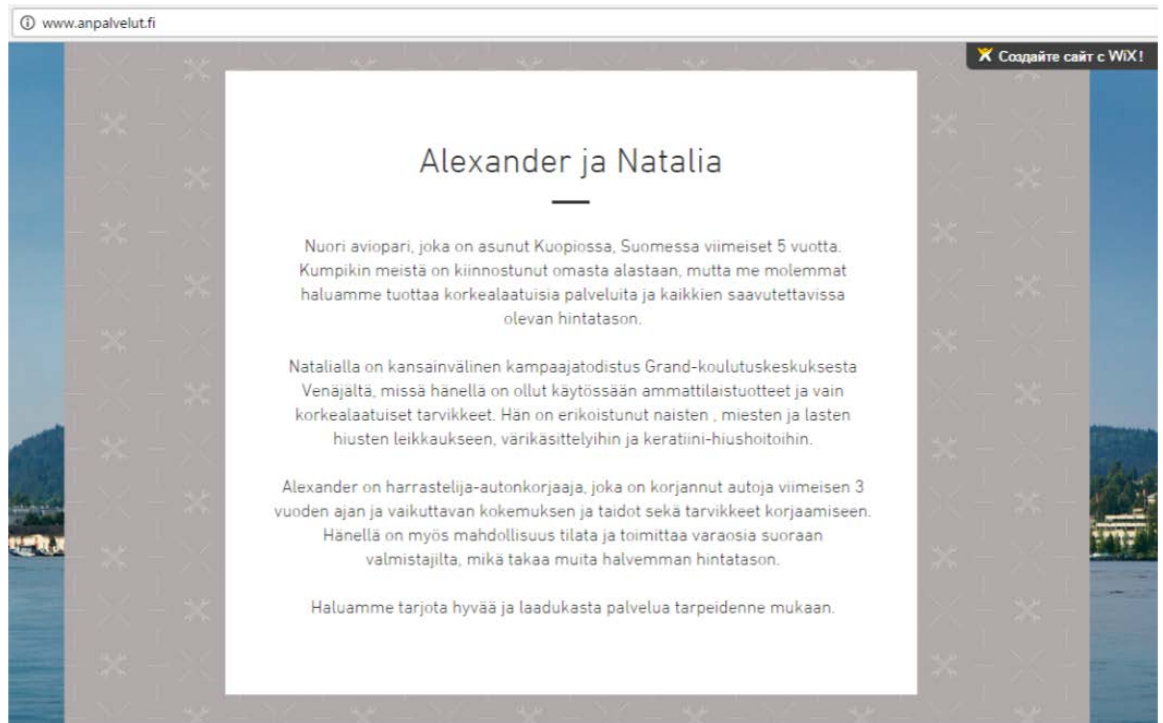


FIGURE 5. Block of text on the main page

Right after that, as shown in Figure 5, the user sees a block of the text with a small description of how that company was created. The text was maximally compressed so the block is as short as possible because due to Jakob Nielsen research (2003) people read closely only 16% of continuous, line text.



FIGURE 6. Facebook microblog on the main page

Scrolling down the page the user sees a photo of the owners of the company and after that there is a Facebook microblog, as in Figure 6, where a client can see what is happening in the company group, he can go directly to this group on Facebook or mark it as his favourite. Probably, this is not the perfect place for a microblog and it has to be moved somewhere else but it will be decided during the working process.

www.anpalvelut.fi

Создайте сайт с WIX!

Kirjoittaa tai soittaa ja kysyy lisää:

alexander.malyshenko@an.palvelut.fi  
+358-456447619

natalia.malyshenko@an.palvelut.fi  
+358-469449456

Nimi

Email

Aihe

Viesti

Lähetä

FIGURE 7. Feedback form in the end of the page

And as a conclusion the client goes to the feedback block, as can be seen in Figure 7, and it is the last part of the main page. If a user has some questions he can send them immediately and they will be redirected to the Wix website page for the manager reply. If some mailboxes are connected to this management service - everything will be sent also there. Also, the user can see email addresses and contact phone numbers of the owners of the company.

This is where the main page ends but there are few minor pages to be seen and the user can go to these pages by clicking one of three elements on the main page in the “Services” block or by choosing them from a header.



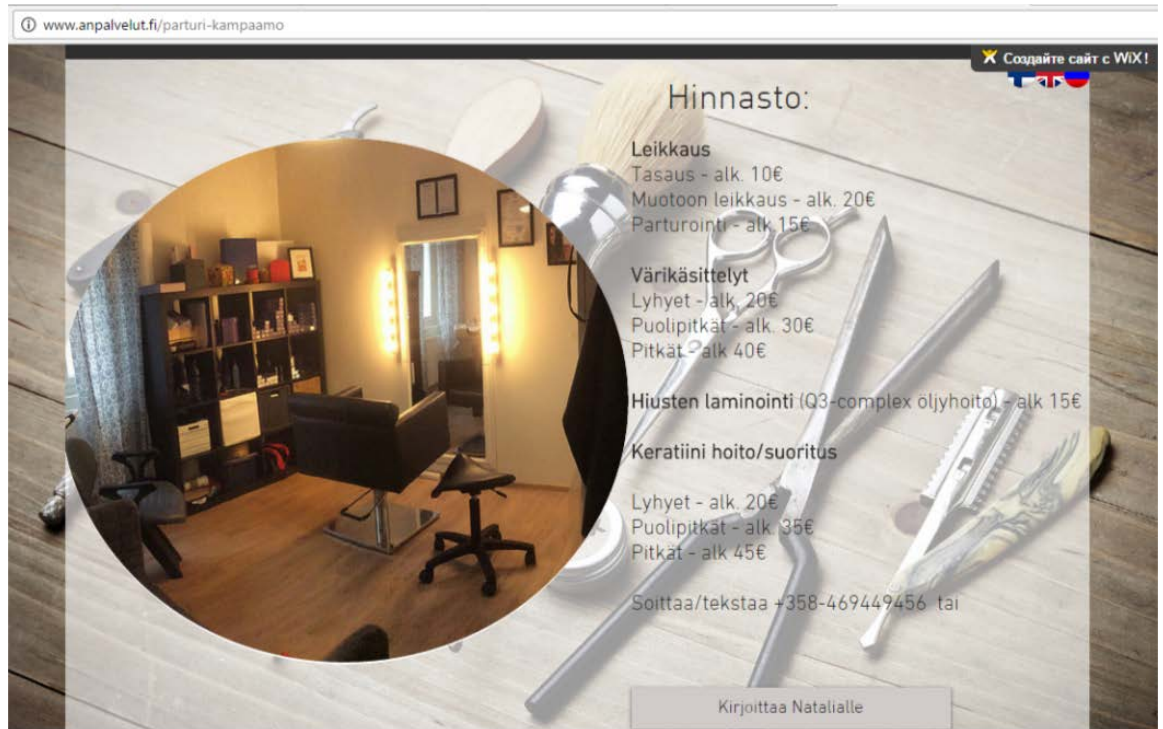


FIGURE 8. Secondary page of the hairdressing studio

Hairdressing page is one of them and Figure 8 represents it. Here the client can get all the information about this service and all the prices as well. Here is also a button for a quick email service access where the outbox message with “Hairdressing” title and recipient address (second owner) opens. There is also a multimedia element – a photo that goes to the full-format after clicking on it.

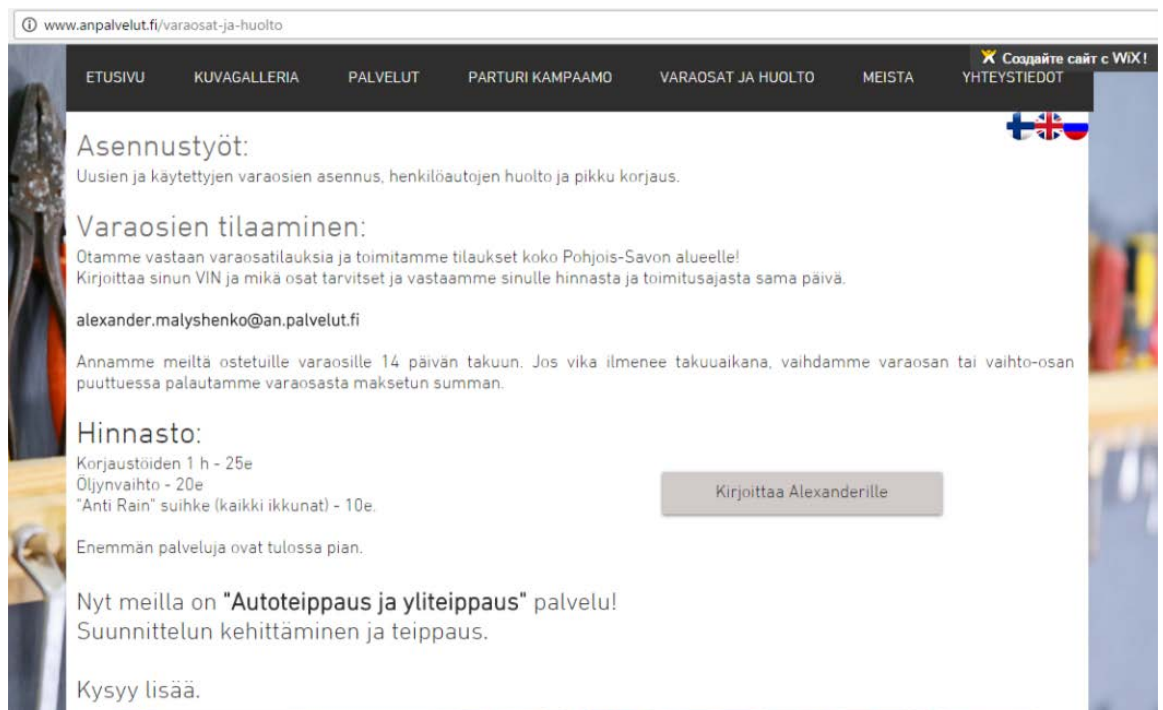


FIGURE 9. Workshop secondary page

Another secondary page is shown in Figure 9 and it is the workshop service and vinyl wrapping description. There is also a pricelist for some services. Of course this page also needs to be improved because there is not enough hierarchy so it is a bit hard to follow all page parts. It is necessary to think about design of the page, maybe it is needed to make it longer and put there a slide show with works that have been done or prices so client can compare these services to other companies providing the same ones.

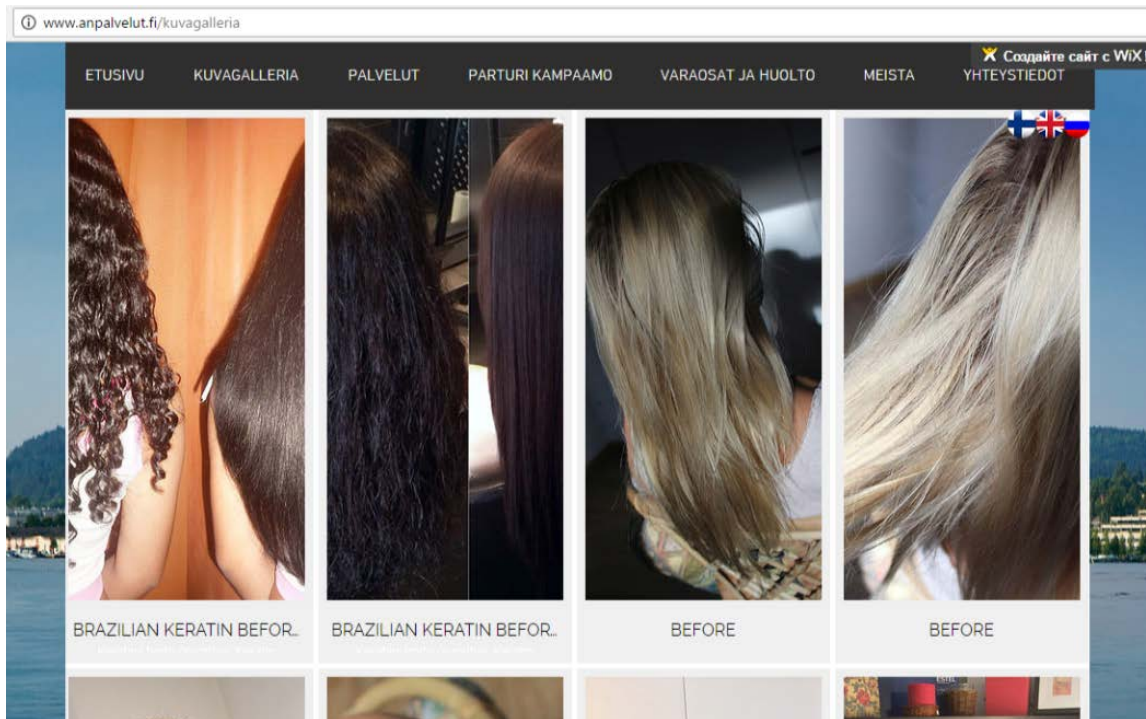


FIGURE 10. Hairdresser gallery secondary page

The last page is the gallery which is represented in Figure 10 with works of a hairdresser where photos “before” and “after” are placed. Again after making this page and asking around ten people for the feedback it was clear that design is not that good because to open the next photo the client has to close the one that is already opened and find next one on the page. Also, all the images were cut from both sides that makes the page look incomplete and muzzy.

## 2.4 Analysis and goals to achieve

Going through the checking and detailed analysis of the website following ideas were proposed for discussion:

- Work on some of the elements on the front page, for instance, add an animation to three symbols in the block of services, so that their icons are expanding a bit, when the

customer hovers the mouse on it. That will help to understand from the first sight that these elements are the links to the services options.

- Create the page for the "interior crafts", because it does not exist yet
- Change the design of the "workshop" page, recompose once again, add a portfolio in the form of a slide-show
- Organize all pages in the same pattern, so that there is no contrast in the design
- Study the visibility and legibility of the text
- The need to elaborate usability of using the web-site in general. The final version has to be easy and convenient in all terms of use.

### 3 THEORETICAL BASE

#### 3.1 Working process with the website

The main problem that was encountered in the process of building and filling in the content, was, of course, the Finnish language page. While pages in Russian and English languages are more or less clear, here with a Finnish translation it had an issue to work on, because it is quite hard to immediately construct complex sentences and process technical text without language background.

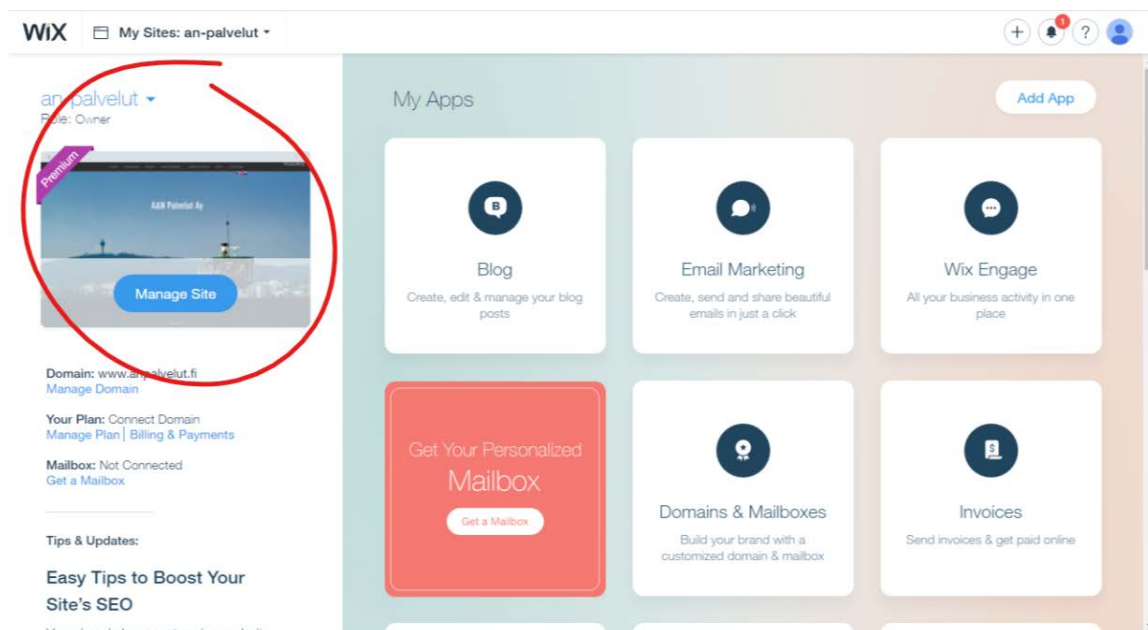


FIGURE 11. Wix.com profile manager

However, as it is shown in Figure 11 from the very first step the user enters the web-creator itself, after that his own profile and selects the web-site on which the work will be done.

On the main menu of the user profile, the main features can be chosen from the list that is presented for the site owner during the work on the builder. Here the owner can configure email marketing, send invoices to customers, or go into a feedback tool, which is imbedded to the builder. This greatly simplifies debugging of communication with customers, allowing customizing message forwarding with feedback forms on the website, directly to your email or you can communicate directly within this utility. But in that work the main goal - website management.



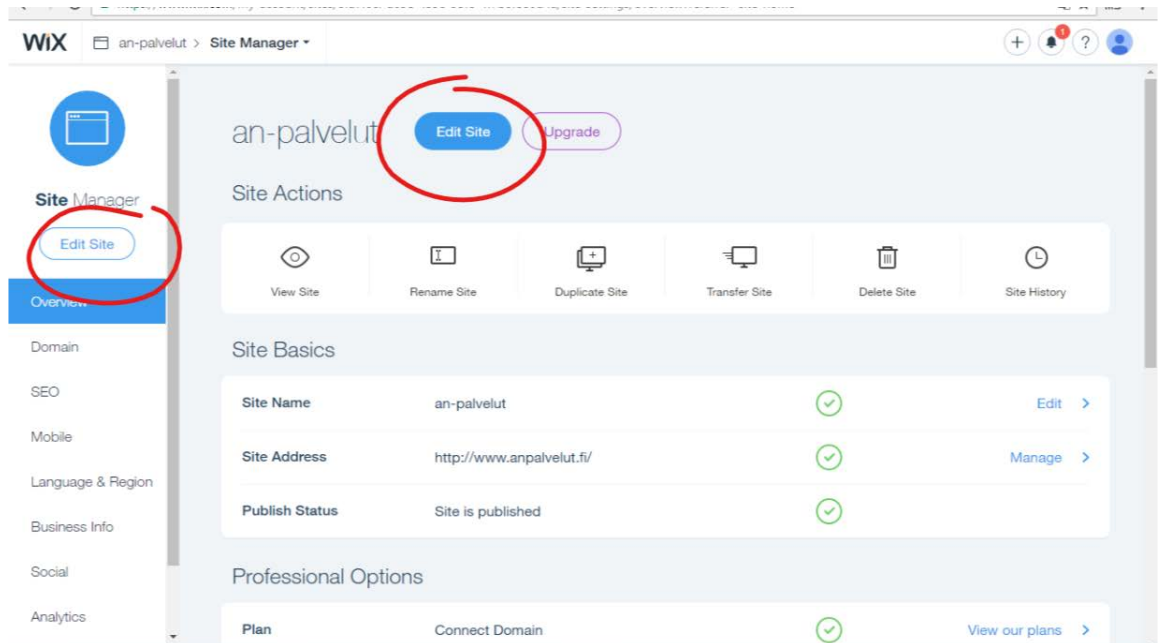


FIGURE 12. Website manager

In the web content management window represented in Figure 12, the user can see all the information about the website, here the domain can be configured, the site can be renamed, user can set up website analysis and much more. In this case, the "Edit site" button has to be chosen to start the main work.

Figure 13 represents very builders' window where design templates can be directly selected, content added or edited, the website binded to social networks and more. All menus are quite simple to understand and the editing process by itself is not too difficult.

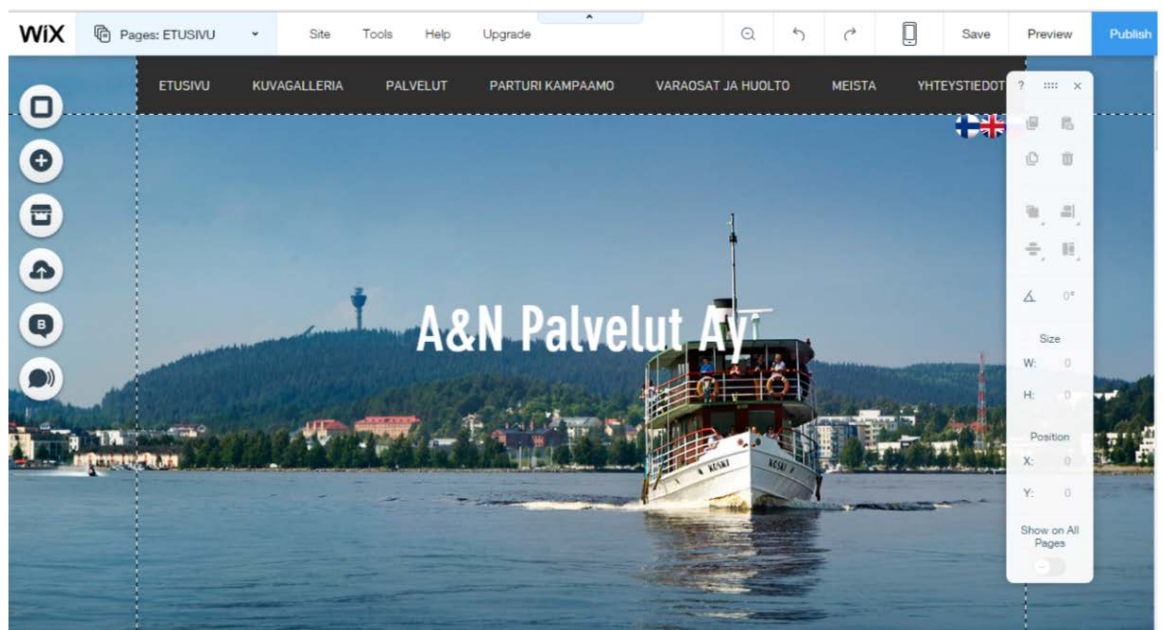


FIGURE 13. Website editor screen

Exactly from this point, some theoretical knowledge can be used for the website improvement, such as theory about the readability of the text and the entire site as a whole object. From here the tasks which have been set out in this work can be started to solve.

### 3.2 Readability of text

To create a website which will become maximally convenient for reading, you need to follow few rules and take into account some features of the pages` alignment.

It`s not enough just to create the text for the website. After all, if the text will be useful and interesting, but the design is terrible, really few people will really want to strain to read it. Therefore it is necessary sometimes to put yourself in the user's place and to wonder "if I want to read it? Am I ready to spend on this text 15 minutes of my life? " (Few 2006, 115-120).

The first role is played by the font, in which the text is written because if it is too sharp or too floating - it will be difficult to read and it is unlikely that users will master the entire page. Take for example one of the most famous fonts "Times New Roman" that is shown in Figure 14, such a serif typeface font used in printing, because it was originally designed for better readability of text on cheap newsprint and save space on the sheets or in literature topography (Efimov & Tarbeev, 1994). Intersections in a font are helping your eyes to lead the line and clearly distinguish the letters (Butterick, 2016).

Times New Roman  
*Times New Roman Italic*  
**Times New Roman Bold**  
***Times New Roman Bold Italic***

FIGURE 14. Times New Roman font

Despite the fact that the font is good for newspapers and other publications, for web design, it is not suitable, because serifs make letters smaller and harder to read.

Another thing - the fonts such as "Arial" shown in Figure 15, which refer to the type of sans serif fonts and much more suitable for reading from the monitor of one's computer, because it was created exactly for this purpose. Sans serif fonts are significantly better

accepted by your eyes and brains and they get bigger when compared with a serif font (Carter & McDonald. 2011).

Arial Regular  
Arial Narrow  
*Arial Italic*  
**Arial Bold**  
***Arial Bold Italic***  
**Arial Black**

FIGURE 15. Arial font

For the visual comparison, the same length of the text, written in three different styles can be taken and then it becomes clear which one is the best for reading. Taking into account that the font size is the same, the distance between rows is made according to standards, etc. Results are represented in Figures 16-18.

Young married couple living in Kuopio, Finland for past 5 years.  
Each of us is interested in his own sphere, but we both want to create  
service of high quality and available price for everyone.

Natalia got an international hairdresser diploma from training center  
"Grand"(Russia), working with professional certified products and  
only high-quality equipment. She specializes on woman, men and  
children haircuts, coloring and keratin hair treatment.

Alexander is an enthusiast who repairs cars for past 3 years and got  
an impressive amount of knowledges and equipment for repairing.  
Also, he has an opportunity to order and deliver spare parts straight  
from manufacturers that makes the price lower than elsewhere.

We want our ambitions work for you.

FIGURE 16. Block of text written by Times New Roman

Young married couple living in Kuopio, Finland for past 5  
years. Each of us is interested in his own sphere, but we both  
want to create service of high quality and available price for  
everyone.

Natalia got an international hairdresser diploma from training  
center "Grand"(Russia), working with professional certified  
products and only high-quality equipment. She specializes on  
woman, men and children haircuts, coloring and keratin hair  
treatment.

Alexander is an enthusiast who repairs cars for past 3 years  
and got an impressive amount of knowledges and equipment  
for repairing. Also, he has an opportunity to order and  
deliver spare parts straight from manufacturers that makes the  
price lower than elsewhere.

We want our ambitions work for you.

FIGURE 17. Block of text written by Arial

*Young married couple living in Kuopio, Finland for past 5 years. Each of us is interested in his own sphere, but we both want to create service of high quality and available price for everyone.*

*Natalia got an international hairdresser diploma from training center "Grand" (Russia), working with professional certified products and only high-quality equipment. She specializes on woman, men and children haircuts, coloring and keratin hair treatment.*

*Alexander is an enthusiast who repairs cars for past 3 years and got an impressive amount of knowledges and equipment for repairing. Also, he has an opportunity to order and deliver spare parts straight from manufacturers that makes the price lower than elsewhere.*

*We want our ambitious work for you.*

FIGURE 18. Block of text written by Gabriola

Needless to say, the curly fonts, such as in the Figure 18 are practically unreadable on a web page. The user is unlikely be delving into the text, which is written in this font. If these fonts are used, they are only applicable for individual titles or slogans (Coady, 2014), but the constant use of these kind of fonts will definitely be a mistake: too much effort should be made by the user to master a website designed in this style.

### 3.3 Hierarchy in text

The next step towards to the optimization, better readability and visibility of the web- page is the use of hierarchies and divisions into particular sections. This supports better orientation among the text and better understanding how to optimize it so that it is better to read it. More often, such an hierarchy is visual and achieved by using different font sizes.

Reading a text with subtitles and captions, the user understands easier what will be discussed in the paragraphs and on the top of the whole text divided into paragraphs psychologically easier to read because it seems for your brain to be shorter.

Again, do not get too much involved in the selection of font sizes and prevent the situation that their appearance on the page is too large, and vice versa too small. Sexton says

(2016) that if you work with pixels, then the scale looks like a set of the font sizes: 12, 14, 16, 18, 21, 24, 36, 48, 60, 72. It will easily define the hierarchy of typography. Always use the font size from the standard scale (The Yahoo! & Barr 2010, 255).

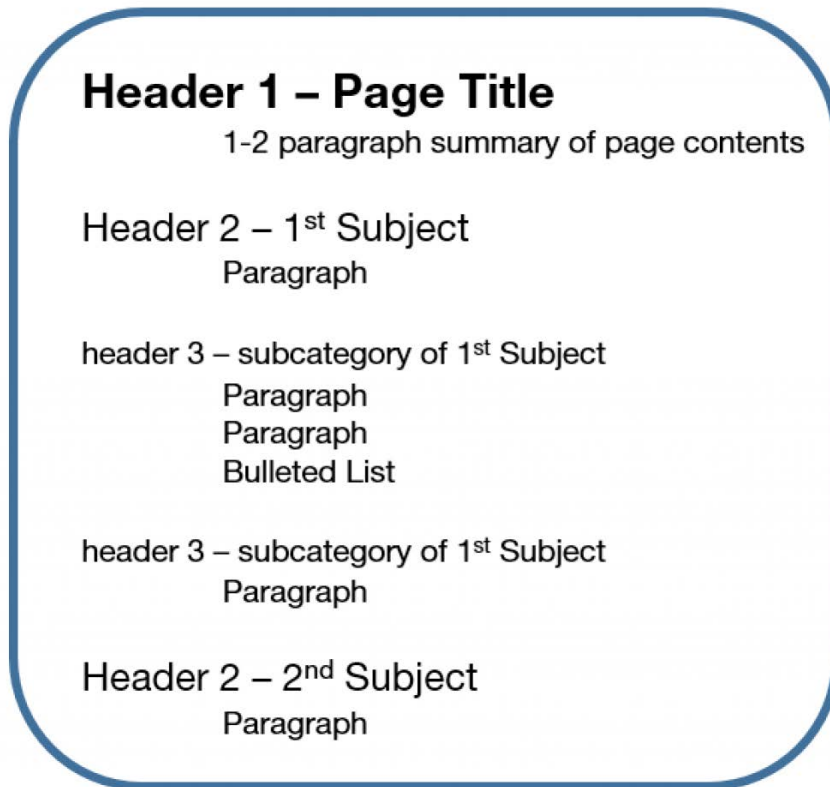


FIGURE 19. Example of hierarchy in text

### 3.4 Contrast

In one list with the previous factors as most important issues can be highlighted the contrast between the text and background. Browsing and reading a site with poorly chosen contrast can be a real challenge for the user, even if he has no vision problems. For a person who is not able to distinguish between colors, or who is color blind, reading this site may become even impossible at all (Schaffer 2004, 43).

The best and most usable contrast – dark text on bright background. Also, it is a good contrast in opposite situation with a dark background and bright text but this text is hard to read for a long time because it strains users vision (Coady, 2014). However, this kind of contrast is good for some small pieces of text. Colors from one color pattern have the worst contrast with each other. In such cases, speed and comfort of reading are dramatically reduced and accordingly reduces the desire to read this website.



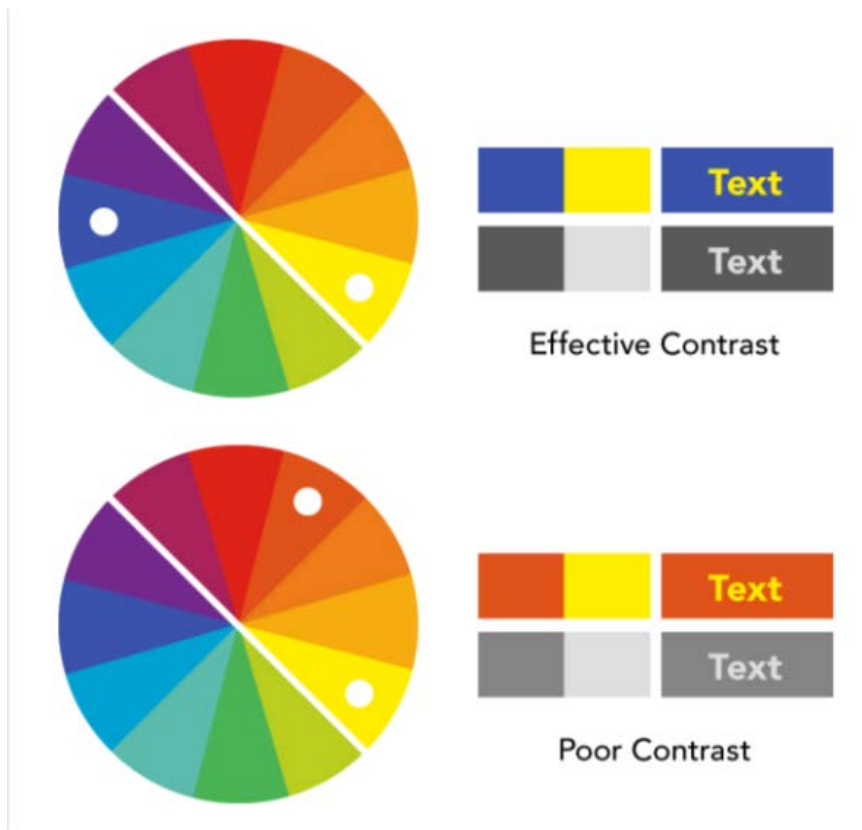


FIGURE 20. Good and bad contrasts on color patterns

### 3.5 Adding Multimedia

One more solution in web design is to add some multimedia content to a website so the user can distract for watching some images or animation between reading different parts of a page (The Yahoo! & Barr 2010, 108).

On the example of two newspaper articles in Figure 21 with and without multimedia content (in this case – images) it is clear that the article with images looks more attractive because of them. Needless to say that there is a huge difference between newspaper and website, their goals and target audience so it is more difficult to imagine a website without multimedia because the user does not want only to read a text (Few 2006, 95). The main point here is to give the user a visualization of what he is reading.



FIGURE 21. Example of two newspaper articles with and without images

### 3.6 Design of website with good usability

So some interim conclusion can be made about how should a website look like so it will be the most convenient and readable in use for user or potential client. To reach that goal all the content should be as short as possible, spelled clearly, divided into parts using hierarchy and paragraphs. Also, everything should be presented in a most comfortable way for the user: font, background color or image, contrast between them, page layout. There should be multimedia content as well so user can relax and distract from reading.

In this way website will be the most readable and attractive for client from visual point of view and it is the first step to engage the customer which is so important for growing business. Properly speaking, the idea of building and editing of the website was to follow these requirements so it would be as simple and comfortable as possible for watching and reading.



## 4 PRACTICAL PART

### 4.1 Changes in the website outlook

In order to make all the pages in one style the web editor was used where all the changes can be made quickly. In general there is not so much specific work but as a result web pages look nicer and more attractive than in the beginning and it is presented in Figure 22.

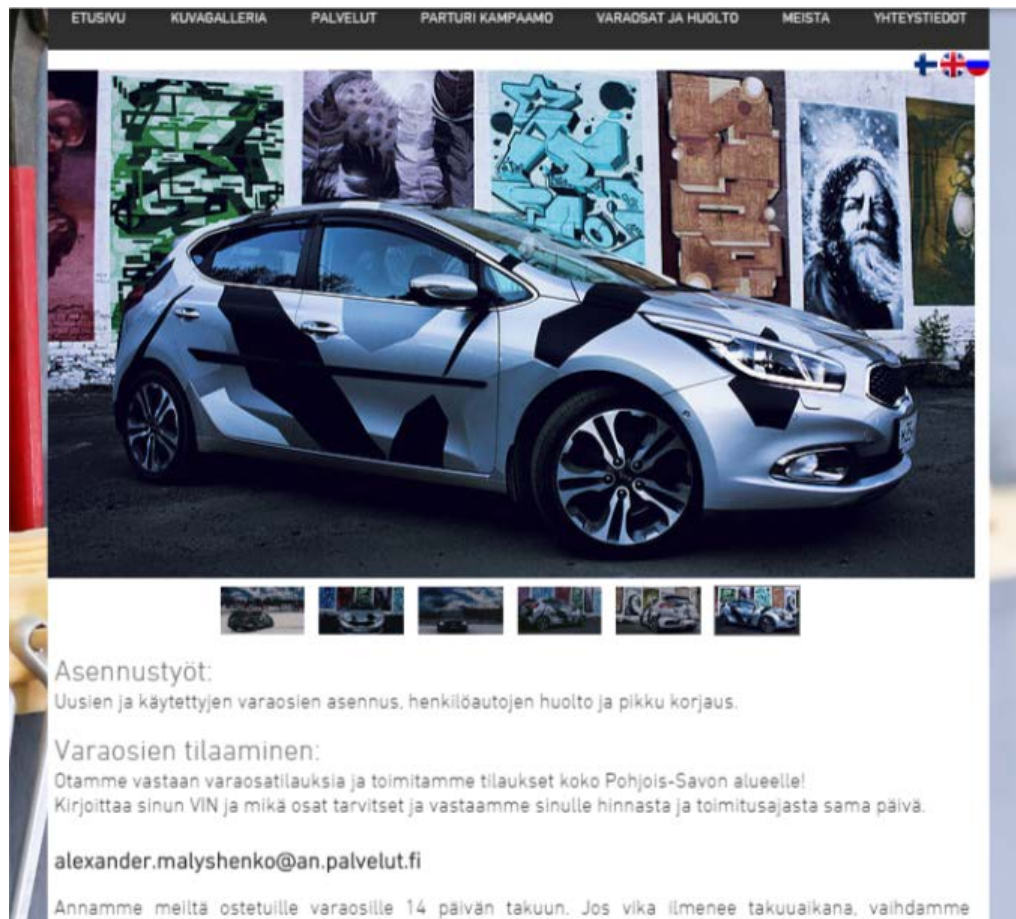


FIGURE 22. Workshop page after rebuilding and all the corrections.

Moreover, the new page has been created for a service that was just started: wooden interior crafts. The idea was almost the same as with the workshop page: to create a small slideshow with works that have already been done and put there some information about the service. With the minimum of information, but the customer should see everything that he needs and he should be able to contact the owners or representatives if interested.

### 4.2 Programming task

After all the changes had been done one more idea had to be realized and that is something that was really connected to programming by itself. The calendar for

hairdressing part of a company where clients could see what time is already booked or free for an ongoing month or any period of time he or she is interested in.

Some requirements have to be estimated for this kind of task:

- the calendar should give a certain information to the customer (booking date and time)
- as there will be a lot of customers - calendar has to have a database so all the information about them can be added and edited
- calendar code should be written in PHP so it will be easier to take information from the database

What problems were faced with this kind of task:

- Wix.com does not give any source code of the website to a user and the only way to add something is through the HTML-application but it can be only something simple (if PHP code is added there it will be shown as a text block). It is represented in Figure 23.

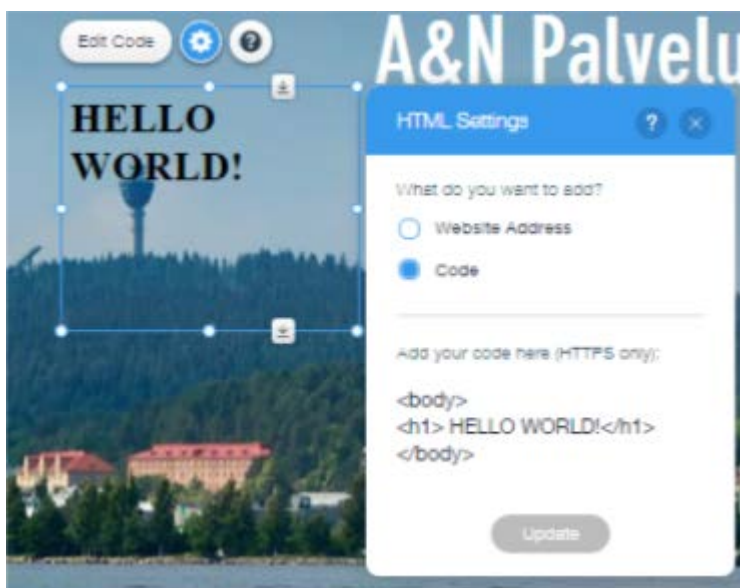


FIGURE 23. Example of code that works in HTML-application

- There is an opportunity to create a calendar on Javascript and it will be shown on a Wix page but to make a request for the database anyway it will be needed to use some PHP code.

The only logical solution for this kind of task was to use HTML-application just for showing another website's address where it will be allowed to use PHP code and work with the database normally.

### 4.3 Making of MySQL database

Unfortunately, it was not possible to use the Wix interface for solving that problem because there is no opportunity to work with a code that was needed. So the new page was made and registered on a new domain so on the main Wix page there will be just a link to the calendar.

But first of all the MySQL database has to be made with the phpMyAdmin tool as shown in Figure 24.

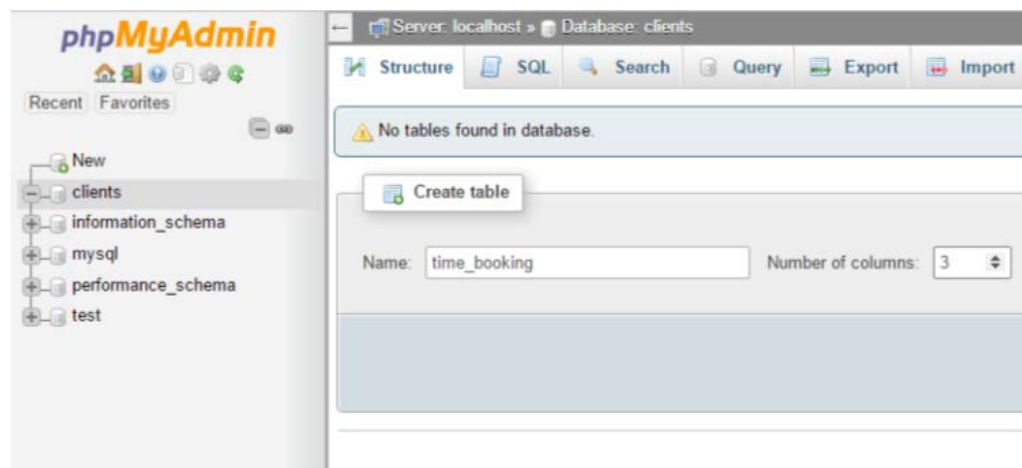


FIGURE 24. Creating of new database

There is no need in more than one table “time\_booking” in “clients” database which is represented in Figure 25 and in this table should be four rows:

- id (INT field, also marked as “unsigned” so there will not be any negative values)
- name (name of client, VARCHAR field where 50 symbols will be enough)
- date (DATE field)
- time (TIME field)

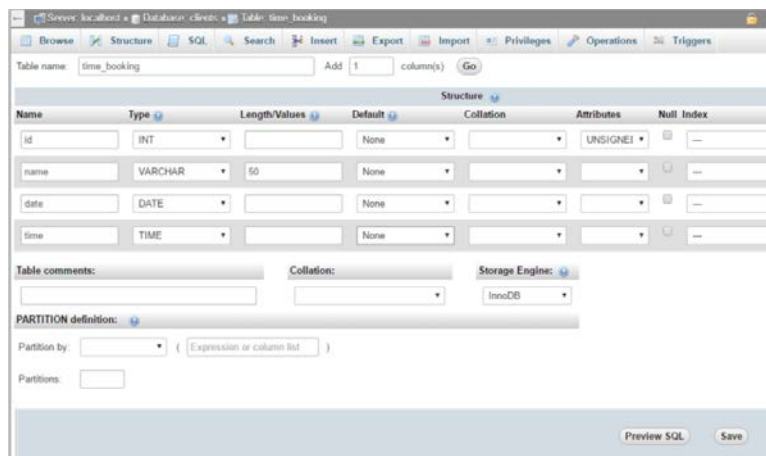


FIGURE 25. Rows we have in database table

Next step is just to save what is done and add clients to the table. For the beginning only six clients were added to check how code will work. Figure 26 represents these process.



id	name	time	date
1	Tarutin Nikita	13:00:00	2016-12-01
2	Marikka Virtanen	16:00:00	2016-12-01
3	Stanislav Belov	17:00:00	2016-12-05
4	Jay Decai	19:00:00	2016-12-07
5	Reeta Timonen	13:00:00	2016-12-19
6	Isaak Graham	16:00:00	2016-12-26

FIGURE 26. Clients in the database

#### 4.4 Calendar code

First of all, all the work should be planned, estimation how this calendar should look like is needed and what goals should be achieved:

- this should be the page with one month showed but with arrows to choose the one that customer needs
- cells should be big enough so it will be easy to point at them with a cursor and numbers inside should be well-visible
- there are two solutions for showing up the information from a database (it can be stored in cells or it can be a pop-up window when the cursor is on the right cell)
- dates with events should be marked somehow (cell background color or thick border lines)
- weekends and holidays should be also marked in the most common way (red color)

And after discussing these details writing of the code can be started and “index.php” (ATTACHMENT 1) file has been created. Few more PHP pages were required for normal functionality of service. “registration.php” (ATTACHMENT 2) was written for the secondary page from which client can register for hairdressing, “conn.php” (ATTACHMENT 3) is responsible for connecting database that was created earlier to a calendar. There is one more PHP script “schedulemodel.php” (ATTACHMENT 4) that generates the calendar for

a month that is chosen with all the dates that are booked and saves all the changes to a database.

#### 4.5 User Interface

Then the user comes to the calendar page that is shown in Figure 27. He sees the ongoing month and he can choose next or previous one if needed. Inside of the day cells there is the information about clients who have already booked their time for hairdressing for that day and time of booking. There is no need in some additional information that the user needs to see when he comes to that page. After choosing the date and time that is needed he can go to the “Register” page by clicking the button below and there will be several fields that he will need to fill up for booking.

<December / 2016>

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
			Tarutin Nikita- 13:00:00 Marikka Virtanen- 16:00:00			
			1	2	3	4
Stanislav Belov- 17:00:00	5	6 Jay Decai-19:00:00	7	8	9	10
Fedor Mikhalemm- 11:11:11	12	13	14	15	16	17
	19	20	21	22	23	24
	26	27 Violetta Raunen- 14:00:00	28 klassikkid- 17:00:26	29	30	31

[Register](#)

FIGURE 27. Calendar page

On the registration page represented in Figure 28 there are few fields to be filled in for a successful booking:

- name
- phone number
- email address
- date of booking
- time of booking

Fields “name”, “date” and “time” are necessary to fill, otherwise there will be an error message near the field that is empty. If registration was made successfully the user automatically comes to a calendar page and he can see himself on a schedule.

The main goal was to create a functionality that will be suitable for this kind of task and there was no accent on a design. That is the reason why everything looks that raw. During the process of coding, a lot of new ideas had come and there is a plenty of time for realization.

**Fill in all the fields to make a reservation:**

Your name\*

Phone number:

Email address:

Date of booking:\*

Time of booking:\*

FIGURE 28. Registration page

#### 4.6 Future plans

There are a lot of functions that can be added to the calendar page to make it work properly and look more attractive to the user. Of course, the deeper knowledge of PHP and databases are required to do that, but in this case that amount of functions in use is an overstock. First of all because the business is not that big nowadays. There are not so many clients so it is unreasonable to make such a difficult realization of functionality.

What can be done in the nearest future is some additional databases that will be connected to the existing one and some more client data can be added there as well. It can be used to create the login page to make the process of registration more protected from bots or indifferent users.

The registration of the user will be made within an email verification when the user has to use some existing email to register in the system and enter it. In this case, the email can be also used as a login and one of the additional databases can have 3 fields:

- email as a login
- password
- identification (ID)
- status (unregistered, waits for registration, registered, deleted)

If the client is new on the website his status is “unregistered”. After filling up all the data and pressing “Register” button he will receive a message from the server with a specifically generated link that will activate the script to change his status to the

“Registered” one. After entering the data and before the registration his status is “waits for registration”. If the user has been deleted, his email will not be used again and he will not receive anything from the company anymore.

Next database will have all the information about the user and will have several fields:

- name
- date of birth
- phone number
- email address
- ID

Here are the same fields as in the first database that is already created. It will be necessary to delete some fields there and the first database will consist of three fields only:

- id
- date of booking
- time of booking

As a result, these databases are used for different purposes but all of them are connected between each other. For example, if the user is registered and he enters the system it shows a message “hello, User!” there “user” changes for a name of a person logging in. The calendar will show all the times booked without names so this information will be private and nobody will know names of other clients.

There will be also some marketing features, so one of the scripts that will be written will go through the database with dates of birth. If the ongoing date is the same as date of birth, message will be sent to this client that he has a discount for company services if he will show that message after coming to hairdressing studio.

If the company will have some news for clients, of course, everything will be sent through the mail client from server to email address from database. Besides, the design should be made on a proper level.

## 5 CONCLUSION

In case of working with web pages, imagination is the only limit for creating additional features for them. The main problem is that there will be no possibility to actualize it using only simple basic PHP code. Knowledge of Object Oriented Programming is required and for sure some special frameworks should be used because many functions they have as a default and there will be no need to write them. This makes it easier to code but only an experienced programmer can do that.

Also as the client company is small, there is no need to create such a complicated system that will be overstocked for a small group of clients. A lot of resources will be spent at least for creating such a system. For better optimization, there should be two programmers: a front-end and a back-end programmer that will keep that system running and add new functions if needed.

This is just a tip of the iceberg and this project can be developed and it can have a lot of secondary projects but a lot of time and knowledge is needed for that. So there is still a lot to learn about PHP programming, OOP programming and frameworks that will be used in the future for projects of this company.



## REFERENCES

- Bradley, S. 2010. 3 *Design Layouts: Gutenberg Diagram, Z-Pattern, And F-Pattern*. [article]. Boulder: VanseoDesign [accessed 21 November 2016]. Available from: <http://vanseodesign.com/web-design/3-design-layouts/>
- Butterick. M. *Practical Typography: A Brief hystory of Times New Roman*. [web document]. [accessed 06 December 2016] Available at: <Http://practicaltypography.com/times-new-roman.html>
- Carter, M & McDonald, R. 2011. Some history about Arial. *Blue pencil* [e-journal]. 2011, No. 18 [accessed 12 December 2016]. Available from: <Http://www.paulshawletterdesign.com/2011/09/blue-pencil-no-18%E2%80%94some-history-about-arial/>
- Coady, G. *Integrating Contrast Checks in your web workflow*. 24 Ways [blog]. 22 December 2014 [accessed 18 December 2016]. Available from: <https://24ways.org/2014/integrating-contrast-checks-in-your-web-workflow/>
- Efimov, V. & Tarbeev, A. 1994. *Text antique of transition style*. [web document]. [accessed 09 December 2016]. Available at: <http://www.paratype.ru/arts/TimesNewRoman.pdf>
- Few, S. 2006. *Information Dashboard Design*. Sebastopol: O'Reilly Media.
- Nielsen, J. 2003. *Top 10 Web Design Mistakes of 2003*. [report]. Fremont: Nielsen Norman Group [accessed 12 December 2016]. Available from: <https://www.nngroup.com/articles/top-10-web-design-mistakes-of-2003/>
- Nielsen, J. 2006. *F-shaped pattern for reading Web content*. [report]. Fremont: Nielsen Norman Group [accessed 22 November 2016]. Available from: <https://www.nngroup.com/articles/f-shaped-pattern-reading-web-content/>
- Neil, T. 2014. *Mobile Design Pattern Gallery, 2<sup>nd</sup> Edition*. Sebastopol: O'Reilly Media.
- Schaffer., E. 2004. *Institutionalization of Usability: A Step-by-Step guide*. Boston: Addison-Wesley Professional.
- Sexton, P. *Use legible font sizes*. Varvy SEO tool [blog]. 5 March 2016 [accessed 17 December 2016]. Available from: <https://varvy.com/mobile/legible-font-size.html>
- The Yahoo! & Barr, C. 2010. *Style Guide: The Ultimate Sourcebook for Writing, Editing, and Creating Content for the Digital World*. New York: St. Martin's Press.
- Wix team. *Web Design*. Wix blog [blog]. 2006 [accessed 27 August 2016]. Available from: <http://www.wix.com/blog/category/web-design/>

## ATTACHMENT 1 (index.php)

```

<?php
include_once "../Conn.php";
include_once "../ScheduleModel.php";

echo '
<style>

/* calendar */
table.calendar { border-left:1px solid #999; }
tr.calendar-row { }
td.calendar-day { min-height:80px; font-size:11px; position:relative; } * html div.calendar-day { height:80px; }
td.calendar-day:hover { background:#eceff5; }
td.calendar-day-np { background:#eee; min-height:80px; } * html div.calendar-day-np { height:80px; }
td.calendar-day-head { background:#ccc; font-weight:bold; text-align:center; width:120px; padding:5px; border-bottom:1px solid #999; border-top:1px solid #999; border-right:1px solid #999; }
div.day-number { background:#999; padding:5px; color:#fff; font-weight:bold; float:right; margin:-5px -5px 0 0; width:20px; text-align:center; }

/* shared */
td.calendar-day, td.calendar-day-np { width:120px; padding:5px; border-bottom:1px solid #999; border-right:1px solid #999; }
</style>
';

/* Generating the calendar */
function draw_calendar($month, $year)
{
    $conn      = new Conn();
    $scheduleModel = new ScheduleModel($conn);
    $schedule   = $scheduleModel->getScheduleByMonthAndYear($month, $year);
    /* Beginning of the table */
    $calendar = '<table cellpadding="0" cellspacing="0" class="calendar">';

    /* Headings */
    $headings = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'];
    $calendar .= '<tr class="calendar-row"><td class="calendar-day-head">' . implode(
        '</td><td class="calendar-day-head">', $headings
    ) . '</td></tr>';

    /* Variables that we need */
    $running_day    = date('N', mktime(0, 0, 0, $month, 1, $year));
    $running_day    = $running_day - 1;
    $days_in_month  = date('t', mktime(0, 0, 0, $month, 1, $year));
    $days_in_this_week = 1;
    $day_counter     = 0;

    /* first line */
    $calendar .= '<tr class="calendar-row">';

```

```

/* blank sells if day does not exist */
for ($x = 0; $x < $running_day; $x++):
    $calendar .= '<td class="calendar-day-np">&nbsp;</td>';
    $days_in_this_week++;
endfor;

/* here we are going to write days */
for ($list_day = 1; $list_day <= $days_in_month; $list_day++):
    $calendar .= '<td class="calendar-day">';
    /* day number to sell */
    $calendar .= '<div class="day-number">' . $list_day . '</div>';
    if (!empty($schedule[$list_day])) {
        foreach ($schedule[$list_day] as $item) {
            $calendar .= '<p>' . $item['name'] . '-' . $item['time'] . '</p>';
        }
    }

    $calendar .= '</td>';
    if ($running_day == 6):
        $calendar .= '</tr>';
        if (($day_counter + 1) != $days_in_month):
            $calendar .= '<tr class="calendar-row">';
        endif;
        $running_day = -1;
        $days_in_this_week = 0;
    endif;
    $days_in_this_week++;
    $running_day++;
    $day_counter++;
endfor;

/* if there are blank sells in the end of the month */
if ($days_in_this_week < 8):
    for ($x = 1; $x <= (8 - $days_in_this_week); $x++):
        $calendar .= '<td class="calendar-day-np">&nbsp;</td>';
    endfor;
endif;

/* closing last line */
$calendar .= '</tr>';

/* closing the calendar */
$calendar .= '</table>';

return $calendar;
}

/* for showing ongoing month and date */
if (!empty($_GET["month"])) {
    $month = $_GET["month"];
} else {
    $month = date("m");
}

if (!empty($_GET["year"])) {
    $year = $_GET["year"];
} else {

```

```
$year = date("Y");  
}  
  
$date = mktime(0, 0, 0, $month, 1, $year);  
$prevDate = mktime(0, 0, 0, $month - 1, 1, $year);  
$nextDate = mktime(0, 0, 0, $month + 1, 1, $year);  
echo '<a href="index.php?month='. date("n", $prevDate) .'&year='. date("Y", $prevDate)  
.'"><</a>';  
echo(date("F / Y", $date));  
echo '<a href="index.php?month='. date("n", $nextDate) .'&year='. date("Y", $nextDate)  
.'">></a>';  
echo draw_calendar(date("n", $date), date("Y", $date));  
echo '<a href="registration.php">Register</a>';
```

## ATTACHMENT 2 (registration.php)

```

<?php
include_once "../Conn.php";
include_once "../ScheduleModel.php";
// define variables and initialize with empty values
$name = $email = $phone = $date = $time = "";
$hasError = false;
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $hasError = true;
        $nameErr = "Name is required";
    } else {
        $name = $_POST["name"];
    }

    if (empty($_POST["date"])) {
        $hasError = true;
        $dateErr = "Date is required";
    } else {
        $date = $_POST["date"];
    }

    if (empty($_POST["time"])) {
        $hasError = true;
        $timeErr = "Time is required";
    } else {
        $time = $_POST["time"];
    }

    if (!$hasError) {
        $conn = new Conn();
        $scheduleModel = new ScheduleModel($conn);
        try {
            $scheduleModel->saveSchedule($_POST);
        } catch (\Exception $ex) {
            exit;
        }

        header( 'Location: /index.php' );
    }
}
?>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Time booking</title>
</head>
<body bgcolor=#EED5A6>
<center>
    <h2>Fill in all the fields to make a reservation:</h2>
    <form method='post' action='registration.php'>
        <p>Your name<font color="#FF0000">*</font></p>
        <span class="error"><?php echo $nameErr;?></span>

```

```

<input type='text' name='name' size='30'/>
<p>&nbsp;</p>
<p>Phone number:</p>
<input type='text' name='phone' size='30'/>
<p>&nbsp;</p>
<p>Email address:</p>
<input type='text' name='email' size='30'/>
<p>&nbsp;</p>
<p>Date of booking:<font color="#FF0000">*</font></p>
<span class="error"><?php echo $dateErr;?></span>
<input type='text' name='date' id="datetimepicker" size="30"/>
<p>&nbsp;</p>
<p>Time of booking:<font color="#FF0000">*</font></p>
<span class="error"><?php echo $timeErr;?></span>
<input type='text' name='time' id='datetimepicker2' size='30'/>
<p>&nbsp;</p>

<input type='submit' value='Submit'/>
</form>
</center>
</body>
<link rel="stylesheet" type="text/css" href="https://cdnjs.cloudflare.com/ajax/libs/jquery-
datetimepicker/2.5.4/jquery.datetimepicker.css"/ >
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.1/jquery.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-
datetimepicker/2.5.4/build/jquery.datetimepicker.full.min.js"></script>
<script>
    jQuery('#datetimepicker').datetimepicker({
        timepicker:false,
        format:'y-m-d'
    });
    jQuery('#datetimepicker2').datetimepicker({
        datepicker:false,
        format:'H:i:s'
    });
</script>
</html>

```

## ATTACHMENT 3 (conn.php)

```

<?php

/**
 * DB connection
 */
class Conn extends mysqli
{
    const HOST    = 'localhost';
    const USER    = 'id360118_anpalvelut';
    const PASSWORD = 'q1w2e3r4t5';
    const SCHEMA   = 'id360118_clients';

    public function __construct()
    {
        parent::__construct(self::HOST, self::USER, self::PASSWORD, self::SCHEMA);
        if (mysqli_connect_error()) {
            die('Connection error (' . mysqli_connect_errno() . ') '
                . mysqli_connect_error());
        }
    }
}

```

## ATTACHMENT 4 (ScheduleModel.php)

```
<?php
```

```
/**
 * DB connection
 */
class ScheduleModel
{
    /**
     * @var mysqli
     */
    private $conn;

    public function __construct(mysqli $conn)
    {
        $this->conn = $conn;
    }

    /**
     * @param int $month
     * @param int $year
     *
     * @return array
     */
    public function getScheduleByMonthAndYear($month = null, $year = null)
    {
        if (empty($month)) {
            $month = date('n');
        }

        if (empty($year)) {
            $year = date('y');
        }
        $result = [];

        $query =
            'SELECT `name`, `time`, DAY(`date`)
            FROM `time_booking`
            WHERE MONTH(`date`) = ? AND YEAR(`date`) = ?
            ORDER BY `date` ASC, `time` ASC';
        if ($stmt = $this->conn->prepare($query)) {
            $stmt->bind_param('ii', $month, $year);
            $stmt->execute();
            $stmt->bind_result($name, $time, $day);
            while ($stmt->fetch()) {
                if (empty($result[$day])) {
                    $result[$day] = [['name' => $name, 'time' => $time]];
                } else {
                    array_push($result[$day], ['name' => $name, 'time' => $time]);
                }
            }
            $stmt->close();
        }
    }
}
```



```

        return $result;
    }

    /**
     * @param array $params
     */
    public function saveSchedule(array $params)
    {
        $query = "INSERT INTO `time_booking`(`name`, `email`, `phone`, `time`, `date`)
VALUES (?, ?, ?, ?, ?)";
        if ($stmt = $this->conn->prepare($query)) {
            $stmt->bind_param(
                'sssss', $params['name'], $params['email'], $params['phone'], $params['time'],
                $params['date']
            );
            $stmt->execute();
            $stmt->close();
        }
    }
}

```

