OULU UNIVERSITY OF
APPLIED SCIENCES

Manish Maharjan

MAGENTO E-COMMERCE STORE WEB APPLICATION DEVELOPMENT

MAGENTO E-COMMERCE STORE WEB APPLICATION DEVELOPMENT

Manish Maharjan
Bachelor's Thesis
Spring 2017
Degree programme in
Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Degree programme in Information Technology

Author: Manish Maharjan
Title of the bachelor’s thesis: Magento E-commerce Store Web Application Development
Supervisor: Eino Niemi
Term and year of completion: Spring 2017 Number of pages: 61

The aim of this bachelor’s thesis was to provide information on current e-commerce trend and market domination and to understand the YourFace brand webstore project application implementation based on the Magento system. This project was implemented and documented under the strict supervision of the Vaimo Finland Oy. In this document, the background of the project implementation, required tools, methodologies and objectives are defined. Some short statistical big data has also been researched, analysed and the results are presented in the document.

This thesis also examines how it would be possible to get more customers and how to assure the customers to do safe business, how a merchant could bring a positive impact and build a right e-commerce store to get more users. Therefore, this thesis also explains how a merchant can build a powerful e-commerce store with the help of the Magneto environment and integrate it with the current latest secure technologies.

In the first few pages the analytics and research data have been presented and later continued with the implementation details. All the findings during the research are presented in detail with the real statistical data, such as browsers type, mobile screen resolution, screen colour, and operating system.

After completion of this project, the results were very fruitful. I gained a handful of experiences, in depth knowledge and tactics on building a powerful webstore that would attract a lot of visitors in any future webstore application development.

Keywords: PHP, PHTML, CSS3, SASS, XML, JavaScript, jQuery, Mercurial, Jira, Bitbucket, Version Control, MVC controller, Widgets

PREFACE

Eino Niemi supervised my Bachelor’s thesis. I got to know him as one of the profound teachers and very enthusiastic instructor. I would like to thank him for his marvelous assistance and being very helpful and encouraging all the time.

I have worked with many different and very interesting projects in Vaimo, YourFace being one of them. I worked with a team where each independent team member had a specific set of tasks. Among those tasks my sole responsibility was to implement the YourFace web store and deliver it successfully on time. Any difficulties or complicated problems encountered were easily solved with the guidance from other colleagues. During the project my main role included implementing the Frontend UI and creating the backend functionality wherever required. I would like to thank the entire Vaimo Finland for giving me an opportunity to work on my thesis with the YourFace project. I had collaborative team members and helpful senior developers. I would also like to thank all of them for helping me to overcome the obstacles and continue to have a good work flow with the project. This tremendous support and wonderful working environment helped me to enhance my knowledge and experiences. I learned not to give up any complicated tasks easily, but rather work and finish it by collaborating with other developers for optimum solutions.

My special thanks go to Marko Rytkönen who always supervised me throughout the project inside the company and communicated with the clients as a Project Manager. I was provided with all the necessary tools, equipment and a very calm working environment throughout the development process. It would have been very difficult to complete the work without his good management.

Oulu, 19 April 2017
Manish Maharjan

CONTENTS

VOCABULARY

WORD	DEFINITION
PHTML	Standard suffix of the HTML page consisting the PHP code. [6] [7]
CSS	Cascading Style Sheets is a style sheet for documenting different layout, fonts, and colors.
SCSS	It is the upgrading version of Cascading Style Sheet. [4]
DOM	Document Object Model is an application programming interface which builds all the object structure in any given valid HTML. [3]
CMS Editor	Content Management System Editor is one of the powerful ways of editing the content through a graphical interface without having needs of writing the code
owl carousel	It is the third party JavaScript library for creating a responsive touch enabled slider. [11]
XML	eXtensible Markup Language is a software tool for storing and transporting of data.
MVC	Model View Controller is a software architectural pattern for implementation of User interface separating the application into three main logical components. [10]

WORD	DEFINITION
ZF	Zend Framework is an open source object-oriented web application framework consisting of professional PHP based packages. [10]
UI	User Interface is a design interpreted by the information device through which a person may interact with the system.
API	Application Program Interface is routine protocols and tools which specify how software components would interact with each other securely. [22]

1 INTRODUCTION

Vaimo was founded in 2008 based on values, such as quality, speed and excellence through which Vaimo has become one of the world's most appreciated experts in eCommerce on the Magento platform. By creating web shops that attract visitors and turn them into customers, Vaimo has helped more than 400 merchants reach higher revenues and build stronger brands. Today, Vaimo is based in Scandinavia, even though Vaimo develops web sites for clients all over the world. With offices in Sweden, UK, Finland, Norway, Denmark, UAE, Japan, South Africa, Estonia, and Ukraine, Vaimo offers a global presence that makes it more experienced, stronger and gives an opportunity to work even closer with the clients. Vaimo is Magento's first Enterprise Partner in Scandinavia and Africa and amongst the top Magento Enterprise Partners in the UK. [20]

Vaimo is the global leader in delivering award-winning digital storefronts, omni-channel solutions and mobile apps on the Magento platform. Vaimo's focus is to accelerate B2B and B2C sales for different brand, retail and manufacturer clients. With 13 global offices across EMEA, APAC and Africa and over 300 employees, Vaimo provides an international presence that allows it to cultivate close, long-term relationships with the clients. Vaimo has delivered over 60 Magento Enterprise level eCommerce solutions and has a vast experience in implementing omni-channel solutions. Magento recognised Vaimo's leadership by selecting Vaimo as its EMEA Partner of the Year in 2015 and its Omnichannel Partner of the Year in 2016. With a focus on driving a digital innovation, Vaimo's services cover a full-spectrum of client's eCommerce needs ranging from strategy, design, development, optimisation, and hosting all the way to wherever eCommerce initiatives reach. [20]

Over the years, Vaimo has delivered more than 400 eCommerce solutions on the Magento platform using the best practices and providing a truly global service with a local expertise. Vaimo has over 250 dedicated in-house experts working in specialised departments through which Vaimo can provide a customer with a one-stop shop for all the eCommerce needs and ambitions. Vaimo also provides solutions to merchants all over the world. Vaimo's talented leadership team consists of like-minded professionals and eCommerce experts who are all dedicated to the success of the clients. [20]

The main aim of this thesis was to gain a handful of experience on building a Magento ecommerce webstore and research how to attract more visitors on an e-commerce store. The ecommerce trend has increased very rapidly nowadays and people love shopping mostly through hassle-free ways. Rather than spending a huge amount of time by visiting one store after another, people love to surf the store webpages online, compare with another competitor for example different quality and prices. They also want the goods to be delivered to home by just sitting in front of the computer. This time saving and flexible opportunity has been brought to customers by an online e-commerce system.

Considering these factors, one of the high-level Finnish fashion chain, YourFace, considered to make a new e-commerce webstore for presenting the women fashion in a beautiful way. They wanted to offer their services to their customers to build up their strong positive emotions, self-confidence, encouragement and own unique character. Therefore, they were confident that this webstore would bring many values to their customer.

In addition, the objectives were to make sure that the customers feel safe during their web visits and that the website attracts many visitors frequently. With these requirements and problems, the Big data statistical research was made and the work based on the optimisation was presented.

Another objective was to develop the whole application in the Magento system and deliver it to the client on time. After the application is fully developed, it will be verified through the Quality assurance department. After the strict quality check, then the application will be transferred for the customer testing and then it will be ready to be released for the final deployment to the production environment and it will be made available publicly.

2 ONLINE TRANSACTIONS

Many people do online transactions daily with or without being aware of the safety measures. There have also been a countless number of people being victimised every day due to surfing in the internet on unsecure and online phishing sites which are designed to steal the user's credentials. People who have a low knowledge of information security are the majority number of people who will be tricked by these kinds of scam websites. Considering these facts and in order to prove the web store that was built as legitimate, various safety and security methods were followed.

During the web development and testing phase, there will be several unused codes which developers generally comment or leave some kind of open section during the testing. This often becomes a security problem if some vulnerabilities will be discovered. Code clean-up and vulnerabilities testing were regularly made in this project. If there were some vulnerabilities discovered, then the system was patched with the latest security fixes.

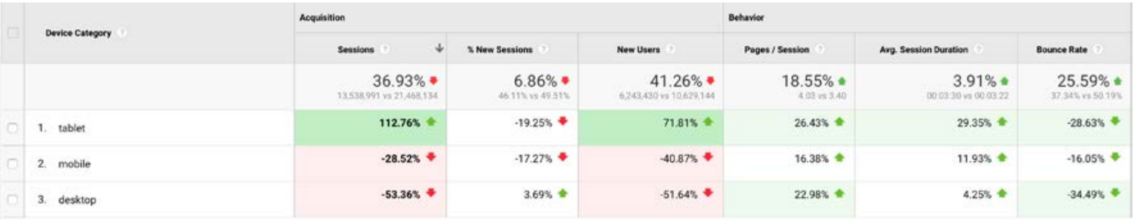
Various security measures were learned and the latest security technology news and updates were followed. Being an enthusiastic learner, the author happily learned many security loop holes which helped him write the clean code efficiently.

3 CURRENT E-COMMERCE TREND AND MARKET DOMINATION

In this research the live google analytics Big Data trend was used to do the research and to learn about the various users' online behaviour. The target was to get statics on users' operating system, devices that the majority of users use, browser type, mobile screen resolution and screen colour. By doing a thorough research of these facts, it was focused on creating a better responsive webpage for the visitors.

3.1 Device usages and Statistics

After researching Big Data, It was noticed that the maximum number of users frequently visits through tablet devices, which is then followed by mobile devices and lastly desktop devices. Checking the user statistics, It was also noticed that people visit through the same session again and again using tablet devices as it is usually much more convenient and new sessions are comparatively lower than desktop media. There were also higher statics on the number of people staying for a longer duration period from a tablet compared to a mobile and a desktop. Surprisingly, the number of people visiting and leaving the website without browsing any further (also termed a bounce rate) was higher from desktop media. Based on these statistics (shown in figure 1) Big Data, it was concluded that in these era, the tablet design needs to be highly responsive to give a good ex-perience on the web interface.

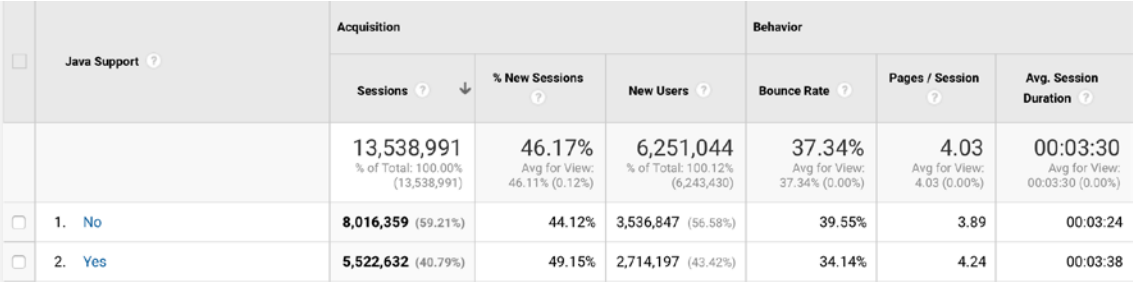


□	Device Category	Acquisition			Behavior		
		Sessions ↓	% New Sessions	New Users	Pages / Session	Avg. Session Duration	Bounce Rate
		36.93% ▼ 13,538,991 vs 21,468,134	6.86% ▼ 46.11% vs 49.51%	41.26% ▼ 6,243,430 vs 10,629,144	18.55% ▲ 4.03 vs 3.40	3.91% ▲ 00:03:30 vs 00:03:22	25.59% ▲ 37.34% vs 50.19%
□	1. tablet	112.76% ▲	-19.25% ▼	71.81% ▲	26.43% ▲	29.35% ▲	-28.63% ▼
□	2. mobile	-28.52% ▼	-17.27% ▼	-40.87% ▼	16.38% ▲	11.93% ▲	-16.05% ▼
□	3. desktop	-53.36% ▼	3.69% ▲	-51.64% ▼	22.98% ▲	4.25% ▲	-34.49% ▼

FIGURE 1. Device Categories [18]

Based on the further Big Data collection it was also noticed that the majority of devices do not have the Java Support. Therefore, creating some API or using

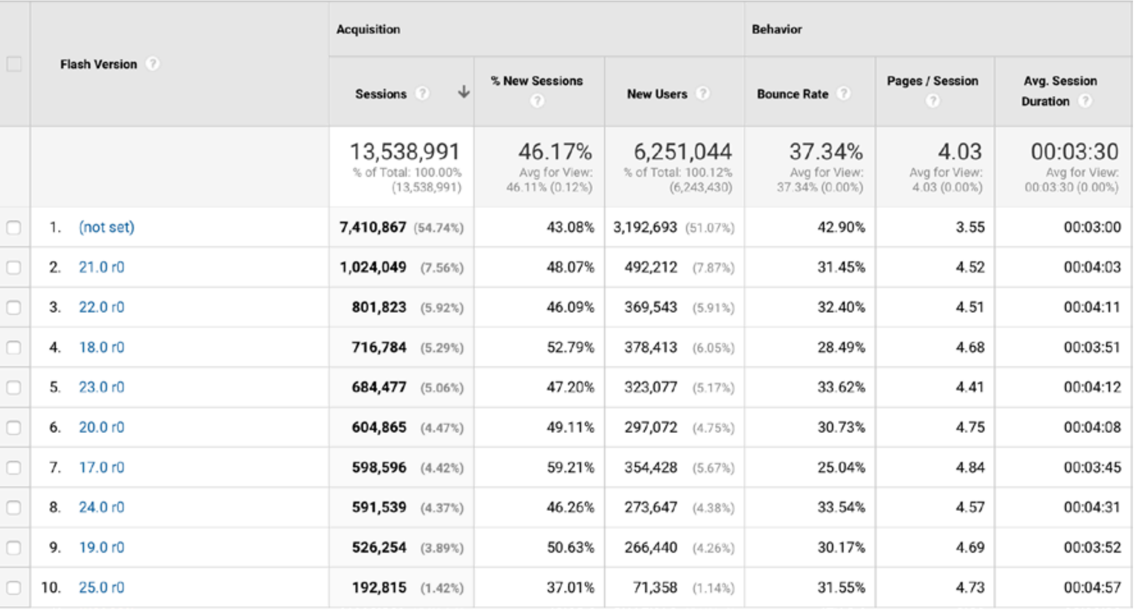
Java applets might be a problematic factor as per shown in figure 2 as statistical Big Data. [18]



	Java Support	Acquisition			Behavior		
□		Sessions ↓	% New Sessions	New Users	Bounce Rate	Pages / Session	Avg. Session Duration
		13,538,991 % of Total: 100.00% (13,538,991)	46.17% Avg for View: 46.11% (0.12%)	6,251,044 % of Total: 100.12% (6,243,430)	37.34% Avg for View: 37.34% (0.00%)	4.03 Avg for View: 4.03 (0.00%)	00:03:30 Avg for View: 00:03:30 (0.00%)
□	1. No	8,016,359 (59.21%)	44.12%	3,536,847 (56.58%)	39.55%	3.89	00:03:24
□	2. Yes	5,522,632 (40.79%)	49.15%	2,714,197 (43.42%)	34.14%	4.24	00:03:38

FIGURE 2. Java Support [18]

Flash is another dominant factor when it comes to a better dynamic content input in a web application. Many security vulnerabilities are revealed from time to time and they need to be updated due to a security reason. The statistics in figure 3, show that the majority of people keeps their devices up-to-date with the latest flash version updates, but also many of them do not take the security concern seriously. [18]



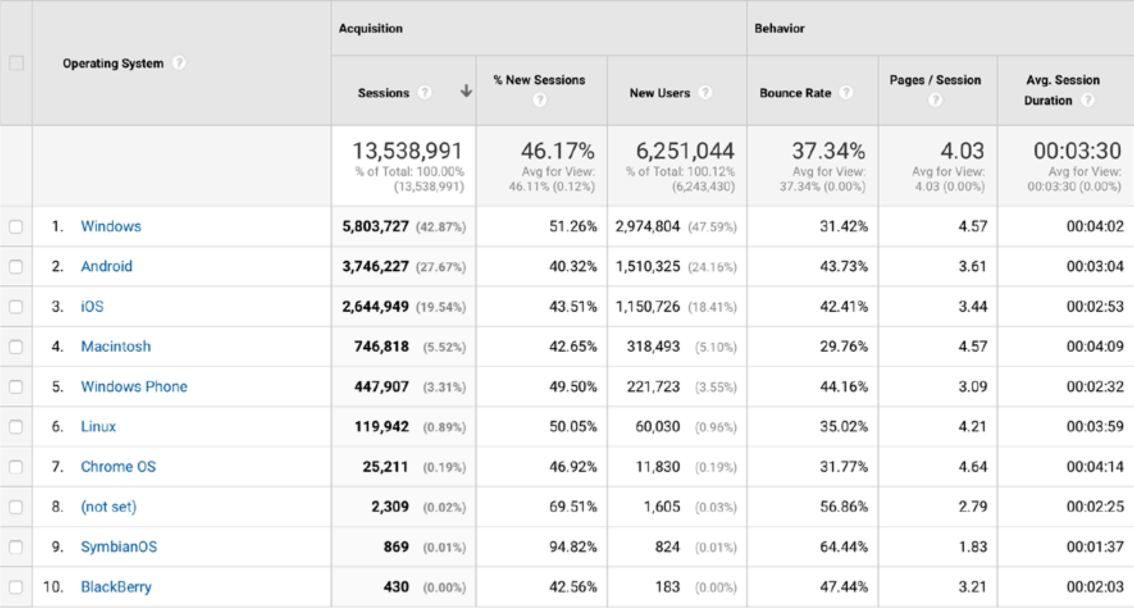
	Flash Version	Acquisition			Behavior		
□		Sessions ↓	% New Sessions	New Users	Bounce Rate	Pages / Session	Avg. Session Duration
		13,538,991 % of Total: 100.00% (13,538,991)	46.17% Avg for View: 46.11% (0.12%)	6,251,044 % of Total: 100.12% (6,243,430)	37.34% Avg for View: 37.34% (0.00%)	4.03 Avg for View: 4.03 (0.00%)	00:03:30 Avg for View: 00:03:30 (0.00%)
□	1. (not set)	7,410,867 (54.74%)	43.08%	3,192,693 (51.07%)	42.90%	3.55	00:03:00
□	2. 21.0 r0	1,024,049 (7.56%)	48.07%	492,212 (7.87%)	31.45%	4.52	00:04:03
□	3. 22.0 r0	801,823 (5.92%)	46.09%	369,543 (5.91%)	32.40%	4.51	00:04:11
□	4. 18.0 r0	716,784 (5.29%)	52.79%	378,413 (6.05%)	28.49%	4.68	00:03:51
□	5. 23.0 r0	684,477 (5.06%)	47.20%	323,077 (5.17%)	33.62%	4.41	00:04:12
□	6. 20.0 r0	604,865 (4.47%)	49.11%	297,072 (4.75%)	30.73%	4.75	00:04:08
□	7. 17.0 r0	598,596 (4.42%)	59.21%	354,428 (5.67%)	25.04%	4.84	00:03:45
□	8. 24.0 r0	591,539 (4.37%)	46.26%	273,647 (4.38%)	33.54%	4.57	00:04:31
□	9. 19.0 r0	526,254 (3.89%)	50.63%	266,440 (4.26%)	30.17%	4.69	00:03:52
□	10. 25.0 r0	192,815 (1.42%)	37.01%	71,358 (1.14%)	31.55%	4.73	00:04:57

FIGURE 1. Flash version in user’s device [18]

3.2 Optimization and Responsiveness Statistics

After researching all these Big Data, it was noticed that optimisation and responsiveness of the users’ website creates a greater value in order to attract many visitors. Obviously, being the developer, the website needs to be taken extra care of the media, which has the highest usage by visitors such as tablet. But the developer cannot ignore to put an effort in making a responsive website to other various media such as a mobile and a desktop.

Based on the operating system statistical data shown in figure 4, there is a higher number of Windows users, which is then followed by Android and then iOS. [18]

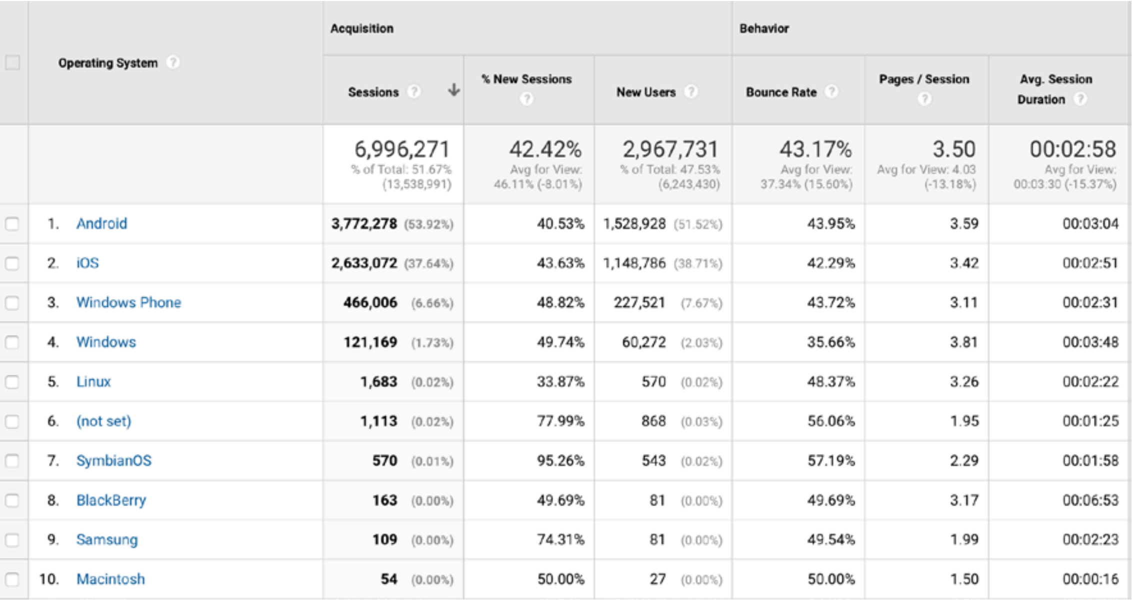


□	Operating System	Acquisition			Behavior		
		Sessions ↓	% New Sessions	New Users	Bounce Rate	Pages / Session	Avg. Session Duration
		13,538,991 % of Total: 100.00% (13,538,991)	46.17% Avg for View: 46.11% (0.12%)	6,251,044 % of Total: 100.12% (6,243,430)	37.34% Avg for View: 37.34% (0.00%)	4.03 Avg for View: 4.03 (0.00%)	00:03:30 Avg for View: 00:03:30 (0.00%)
□	1. Windows	5,803,727 (42.87%)	51.26%	2,974,804 (47.59%)	31.42%	4.57	00:04:02
□	2. Android	3,746,227 (27.67%)	40.32%	1,510,325 (24.16%)	43.73%	3.61	00:03:04
□	3. iOS	2,644,949 (19.54%)	43.51%	1,150,726 (18.41%)	42.41%	3.44	00:02:53
□	4. Macintosh	746,818 (5.52%)	42.65%	318,493 (5.10%)	29.76%	4.57	00:04:09
□	5. Windows Phone	447,907 (3.31%)	49.50%	221,723 (3.55%)	44.16%	3.09	00:02:32
□	6. Linux	119,942 (0.89%)	50.05%	60,030 (0.96%)	35.02%	4.21	00:03:59
□	7. Chrome OS	25,211 (0.19%)	46.92%	11,830 (0.19%)	31.77%	4.64	00:04:14
□	8. (not set)	2,309 (0.02%)	69.51%	1,605 (0.03%)	56.86%	2.79	00:02:25
□	9. SymbianOS	869 (0.01%)	94.82%	824 (0.01%)	64.44%	1.83	00:01:37
□	10. BlackBerry	430 (0.00%)	42.56%	183 (0.00%)	47.44%	3.21	00:02:03

FIGURE 4. Users’ Operating system usages [18]

Looking at the mobile operating system statistical data usages shown in figure 5, Android and iOS take the first and second place and windows Phone has the third place. A different operating system has a different system default UI and it renders the website differently. For example: The search box or drop down menu in Android devices looks different than that of the iOS and other operating systems.

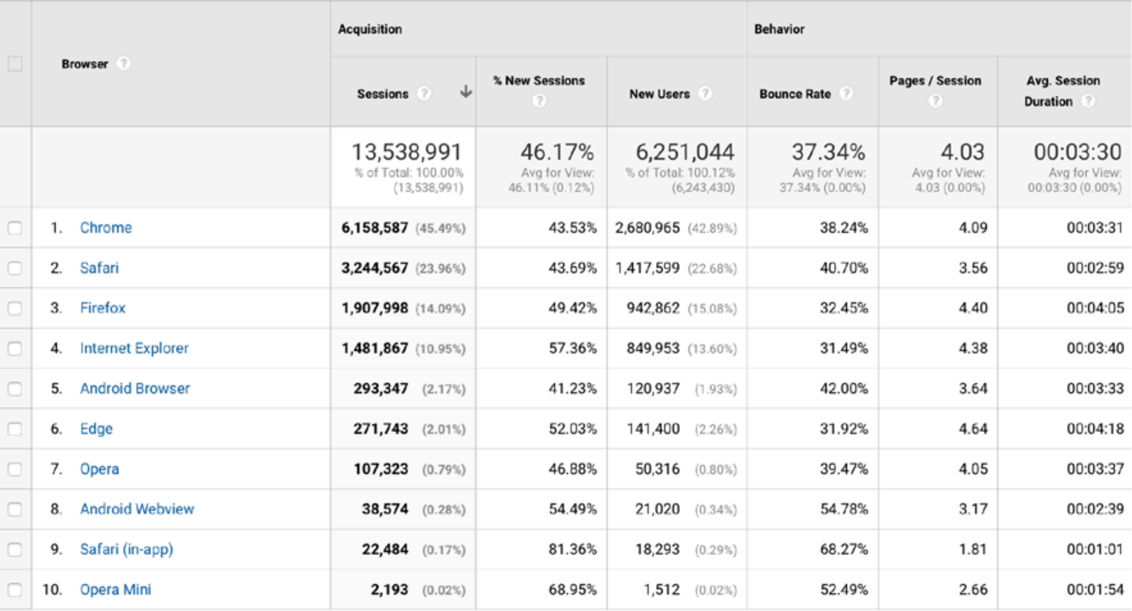
It is hard to make a consistent layout to look similar throughout all devices. Therefore, in order to give the best user experience, the operating system that is used also plays a vital role for the developers to take into consideration. [18]



□	Operating System	Acquisition			Behavior		
		Sessions	% New Sessions	New Users	Bounce Rate	Pages / Session	Avg. Session Duration
		6,996,271 % of Total: 51.67% (13,538,991)	42.42% Avg for View: 46.11% (-8.01%)	2,967,731 % of Total: 47.53% (6,243,430)	43.17% Avg for View: 37.34% (15.60%)	3.50 Avg for View: 4.03 (-13.18%)	00:02:58 Avg for View: 00:03:30 (-15.37%)
□	1. Android	3,772,278 (53.92%)	40.53%	1,528,928 (51.52%)	43.95%	3.59	00:03:04
□	2. iOS	2,633,072 (37.64%)	43.63%	1,148,786 (38.71%)	42.29%	3.42	00:02:51
□	3. Windows Phone	466,006 (6.66%)	48.82%	227,521 (7.67%)	43.72%	3.11	00:02:31
□	4. Windows	121,169 (1.73%)	49.74%	60,272 (2.03%)	35.66%	3.81	00:03:48
□	5. Linux	1,683 (0.02%)	33.87%	570 (0.02%)	48.37%	3.26	00:02:22
□	6. (not set)	1,113 (0.02%)	77.99%	868 (0.03%)	56.06%	1.95	00:01:25
□	7. SymbianOS	570 (0.01%)	95.26%	543 (0.02%)	57.19%	2.29	00:01:58
□	8. BlackBerry	163 (0.00%)	49.69%	81 (0.00%)	49.69%	3.17	00:06:53
□	9. Samsung	109 (0.00%)	74.31%	81 (0.00%)	49.54%	1.99	00:02:23
□	10. Macintosh	54 (0.00%)	50.00%	27 (0.00%)	50.00%	1.50	00:00:16

FIGURE 5. Mobile operating system statistics [18]

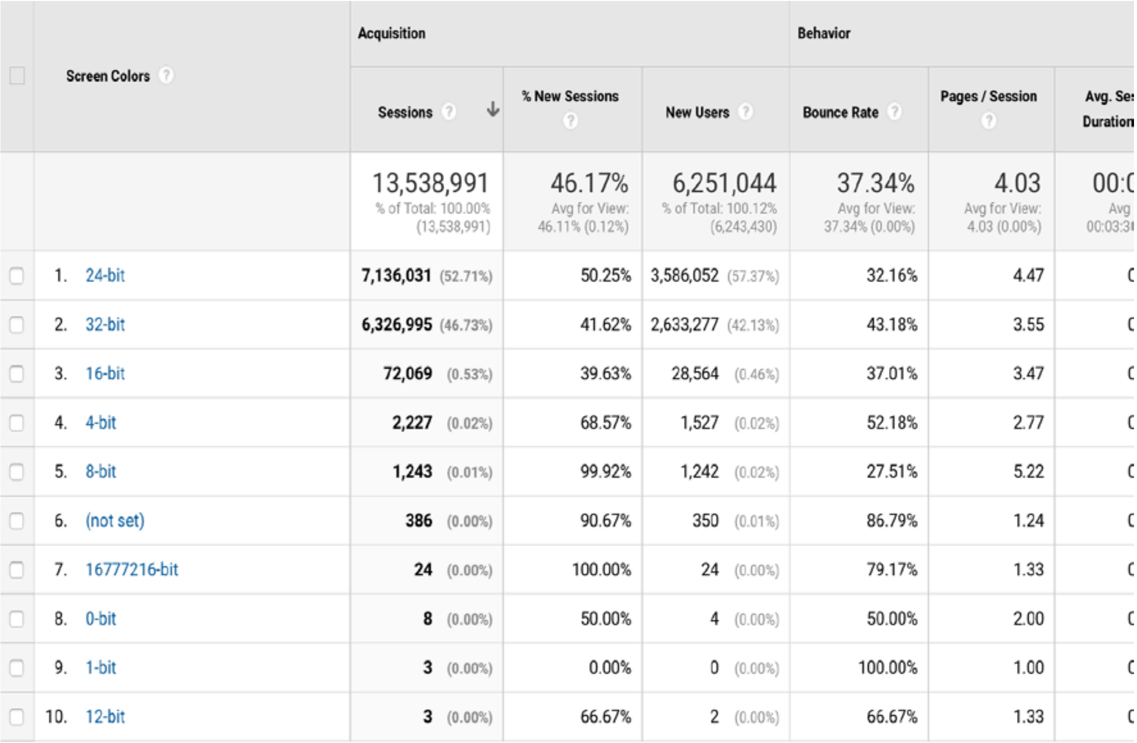
As discussed above, another factor that comes in the rendering is a web browser. During the project this application was tested with various browsers in order to avoid a rendering problem and to keep up the consistency. There are numerous browsers available in the Internet, but the statistical Big Data shown in figure 6, shows the fact that many users visit through the Chrome, Safari, Firefox and Internet explorer. It is also the matter of fact that it is not possible to go through every device and test the content for the developers. Therefore, it was mostly tested how the codes get rendered by these top four browsers. Surprisingly, Internet Explorer was not able to render some animations correctly. Therefore, in the code had to be fixed, to render the animations content correctly also for the people visiting through Internet Explorer. [18]



□	Browser	Acquisition			Behavior		
		Sessions	% New Sessions	New Users	Bounce Rate	Pages / Session	Avg. Session Duration
		13,538,991 % of Total: 100.00% (13,538,991)	46.17% Avg for View: 46.11% (0.12%)	6,251,044 % of Total: 100.12% (6,243,430)	37.34% Avg for View: 37.34% (0.00%)	4.03 Avg for View: 4.03 (0.00%)	00:03:30 Avg for View: 00:03:30 (0.00%)
□	1. Chrome	6,158,587 (45.49%)	43.53%	2,680,965 (42.89%)	38.24%	4.09	00:03:31
□	2. Safari	3,244,567 (23.96%)	43.69%	1,417,599 (22.68%)	40.70%	3.56	00:02:59
□	3. Firefox	1,907,998 (14.09%)	49.42%	942,862 (15.08%)	32.45%	4.40	00:04:05
□	4. Internet Explorer	1,481,867 (10.95%)	57.36%	849,953 (13.60%)	31.49%	4.38	00:03:40
□	5. Android Browser	293,347 (2.17%)	41.23%	120,937 (1.93%)	42.00%	3.64	00:03:33
□	6. Edge	271,743 (2.01%)	52.03%	141,400 (2.26%)	31.92%	4.64	00:04:18
□	7. Opera	107,323 (0.79%)	46.88%	50,316 (0.80%)	39.47%	4.05	00:03:37
□	8. Android Webview	38,574 (0.28%)	54.49%	21,020 (0.34%)	54.78%	3.17	00:02:39
□	9. Safari (in-app)	22,484 (0.17%)	81.36%	18,293 (0.29%)	68.27%	1.81	00:01:01
□	10. Opera Mini	2,193 (0.02%)	68.95%	1,512 (0.02%)	52.49%	2.66	00:01:54

FIGURE 6. Web browsers used by users [18]

Screen colours are another major factor that affect the quality of the webpage display after rendering. As shown in the data in FIGURE 7, it seems that many people have 16-bit, 24-bit and 32-bit screen colours. These statistics also prove that the bounce rate is higher to those people who have a lower screen depth colour. This means that the people visiting the website from a lower screen colour, website colours do not render smoothly. Therefore, people tend to leave the website soon.

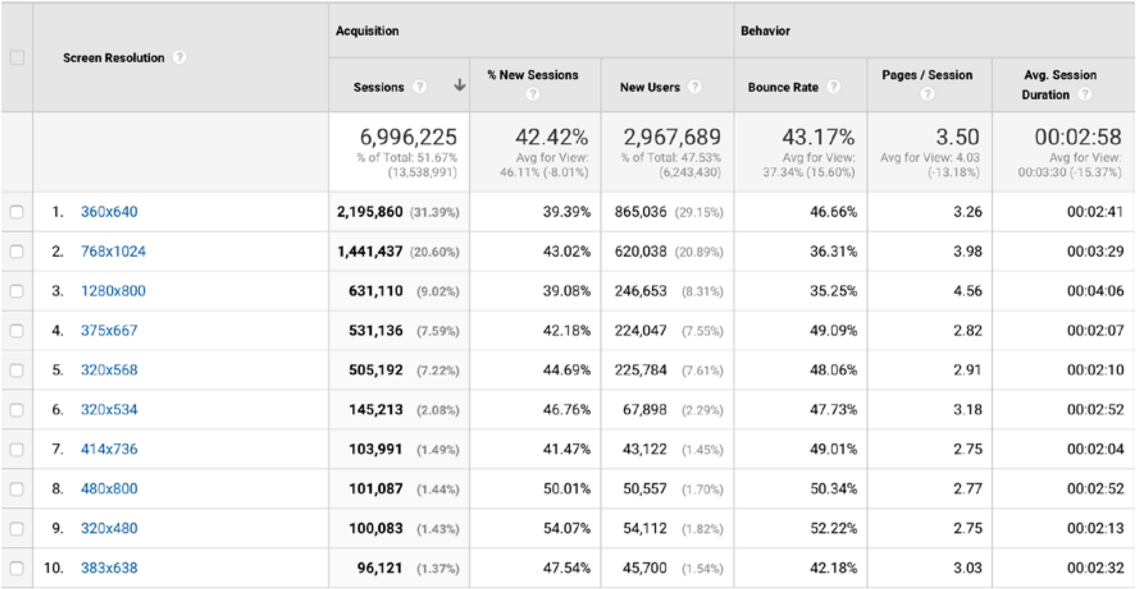


□	Screen Colors	Acquisition			Behavior		
		Sessions	% New Sessions	New Users	Bounce Rate	Pages / Session	Avg. Ses Duration
		13,538,991 % of Total: 100.00% (13,538,991)	46.17% Avg for View: 46.11% (0.12%)	6,251,044 % of Total: 100.12% (6,243,430)	37.34% Avg for View: 37.34% (0.00%)	4.03 Avg for View: 4.03 (0.00%)	00:0 Avg 00:03:3
□	1. 24-bit	7,136,031 (52.71%)	50.25%	3,586,052 (57.37%)	32.16%	4.47	0
□	2. 32-bit	6,326,995 (46.73%)	41.62%	2,633,277 (42.13%)	43.18%	3.55	0
□	3. 16-bit	72,069 (0.53%)	39.63%	28,564 (0.46%)	37.01%	3.47	0
□	4. 4-bit	2,227 (0.02%)	68.57%	1,527 (0.02%)	52.18%	2.77	0
□	5. 8-bit	1,243 (0.01%)	99.92%	1,242 (0.02%)	27.51%	5.22	0
□	6. (not set)	386 (0.00%)	90.67%	350 (0.01%)	86.79%	1.24	0
□	7. 16777216-bit	24 (0.00%)	100.00%	24 (0.00%)	79.17%	1.33	0
□	8. 0-bit	8 (0.00%)	50.00%	4 (0.00%)	50.00%	2.00	0
□	9. 1-bit	3 (0.00%)	0.00%	0 (0.00%)	100.00%	1.00	0
□	10. 12-bit	3 (0.00%)	66.67%	2 (0.00%)	66.67%	1.33	0

FIGURE 7. Screen Colours [18]

Another researched statistical data shown in figure 8 was the screen resolution of the users. It is a fact that webpages will be scaled differently with a different view port and screen size from where the users are browsing. It is not possible to create or scale the same desktop size webpage to mobile users because of the different screen size and view port. For instance, if the webpage was developed with the 1920 x 1024 resolution and if there is no mobile version designed for that website, then if the users visit from the iPhone having a view port of 320px width, then the webpage would scale down to that website. It would look extra small and users could hardly follow up any content without zooming and spanning.

Therefore, it would be very critical to optimise the web content to the users with a screen resolution of below 768px. [18] [17]



□	Screen Resolution	Acquisition			Behavior		
		Sessions	% New Sessions	New Users	Bounce Rate	Pages / Session	Avg. Session Duration
		6,996,225 % of Total: 51.67% (13,538,991)	42.42% Avg for View: 46.11% (-8.01%)	2,967,689 % of Total: 47.53% (6,243,430)	43.17% Avg for View: 37.34% (15.60%)	3.50 Avg for View: 4.03 (-13.18%)	00:02:58 Avg for View: 00:03:30 (-15.37%)
□	1. 360x640	2,195,860 (31.39%)	39.39%	865,036 (29.15%)	46.66%	3.26	00:02:41
□	2. 768x1024	1,441,437 (20.60%)	43.02%	620,038 (20.89%)	36.31%	3.98	00:03:29
□	3. 1280x800	631,110 (9.02%)	39.08%	246,653 (8.31%)	35.25%	4.56	00:04:06
□	4. 375x667	531,136 (7.59%)	42.18%	224,047 (7.55%)	49.09%	2.82	00:02:07
□	5. 320x568	505,192 (7.22%)	44.69%	225,784 (7.61%)	48.06%	2.91	00:02:10
□	6. 320x534	145,213 (2.08%)	46.76%	67,898 (2.29%)	47.73%	3.18	00:02:52
□	7. 414x736	103,991 (1.49%)	41.47%	43,122 (1.45%)	49.01%	2.75	00:02:04
□	8. 480x800	101,087 (1.44%)	50.01%	50,557 (1.70%)	50.34%	2.77	00:02:52
□	9. 320x480	100,083 (1.43%)	54.07%	54,112 (1.82%)	52.22%	2.75	00:02:13
□	10. 383x638	96,121 (1.37%)	47.54%	45,700 (1.54%)	42.18%	3.03	00:02:32

FIGURE 8. Screen resolution of users [18]

Based on these Big Data research, it was concluded that in order to make a powerful webstore and attract more visitors, it would be very essential for the developers to take this data into account. Developers need to develop any web application content to give users the best web experience so that many users would feel comfortable when using the website and it would attract a bigger flow of visitors. With all these learned tactics, the author managed to work with this flow in the YourFace project very efficiently.

4 ECOMMERCE WEB APPLICATION SYSTEM

There are many open source and enterprise edition e-commerce web applications available currently in the market. Magento, Prestashop, WooCommerce, Opencart, ZenCart, Shopify, and VirtueMart are a few examples which are used by millions of people around the globe. Among these various e-commerce systems, Magento has been dominant towards the e-commerce industries. Magento has released both community and enterprise editions based on the consumer needs. The Magento community edition was used in this project because the project itself does not require handling many products compared to the Magento enterprise edition. The Magento community edition is also open source and freely available with no yearly license. It also provides flexible and powerful API integrations with different databases and other stores as per necessity.

4.1 Magento Ecommerce Architecture

Magento complies with the Model View Controller system architecture for handling all the user requests. When the user sends the request by inputting the URL into the browser, then the user is provided with the default view page as per the URL. During this process, Magento View sends the data to the controller. Then the controller communicates with the Model, which also communicates further with the database and finally the view page gets the data to show in the user's requested browser as shown in figure 9. [1]

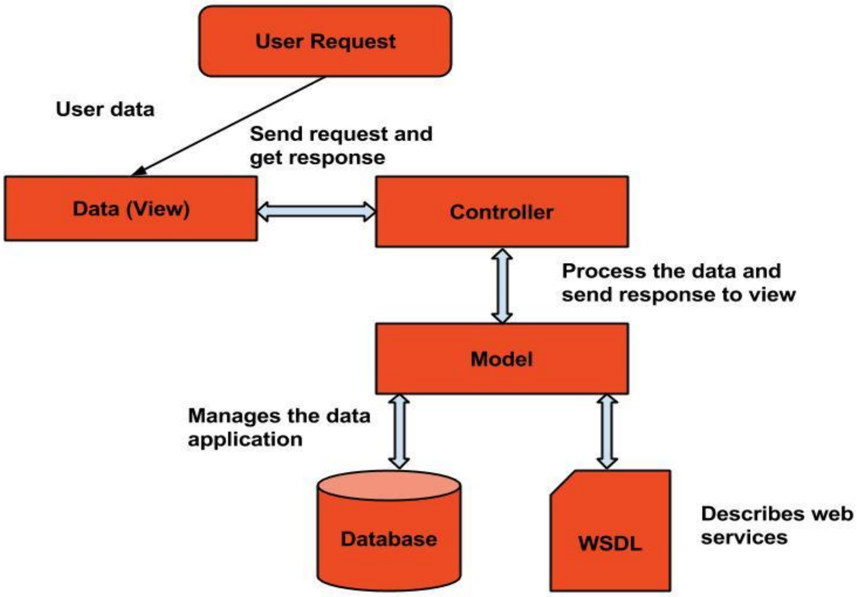
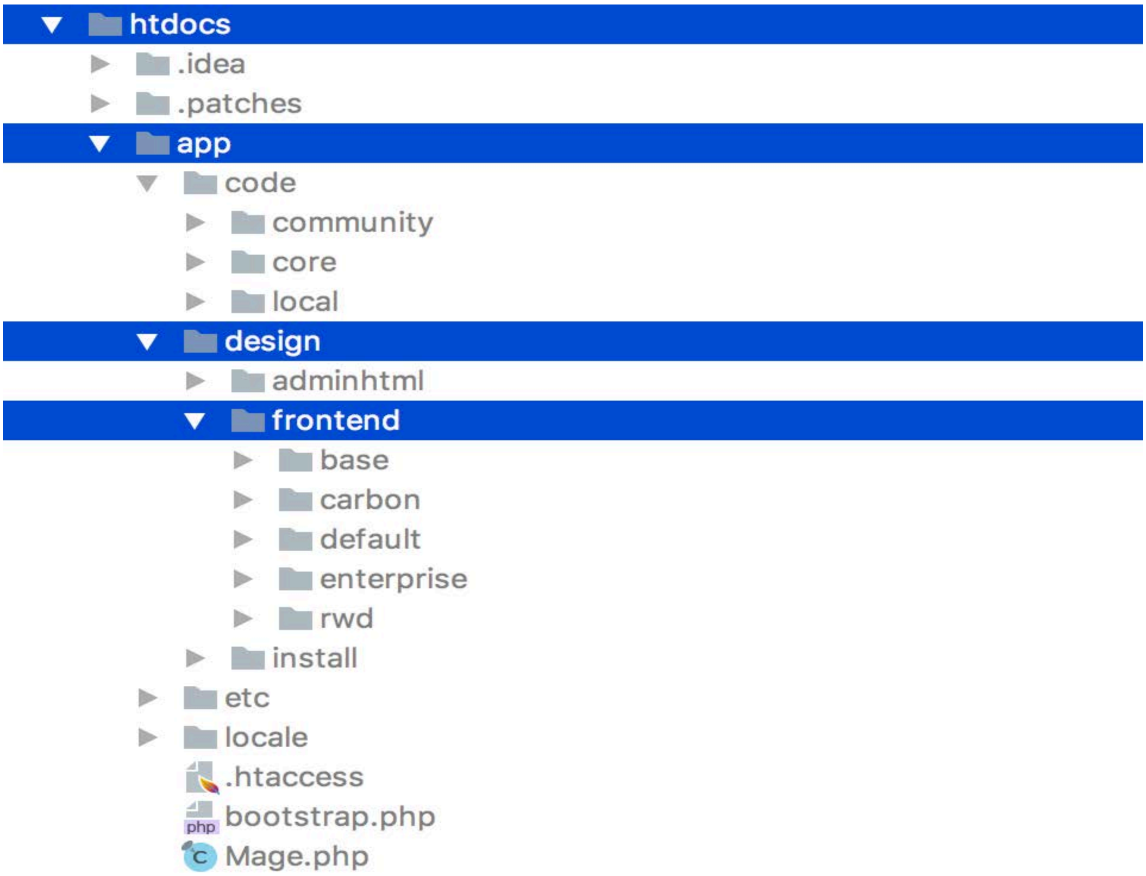


FIGURE 9. Magento Application architecture [10]

4.1.1 Magento Fall Back Structure

Magento follows a specific fall back structure with respect to the user's directory. For example: If Magento needs to access the theme directory when the user requests the data then it first searches through the custom directory created under the *app* > *code* > *design* > *frontend* > *then custom theme package* > *custom theme*. In this case, carbon is the custom theme package. Therefore, this gives the flexibility to extending or overriding the default Magento theme files with the custom codes and modifying them wherever necessary without touching the default core files. The developer created the theme files under a theme package name "carbon" and modified them as per the design to match the user interface which can be seen in figure 10. [1]

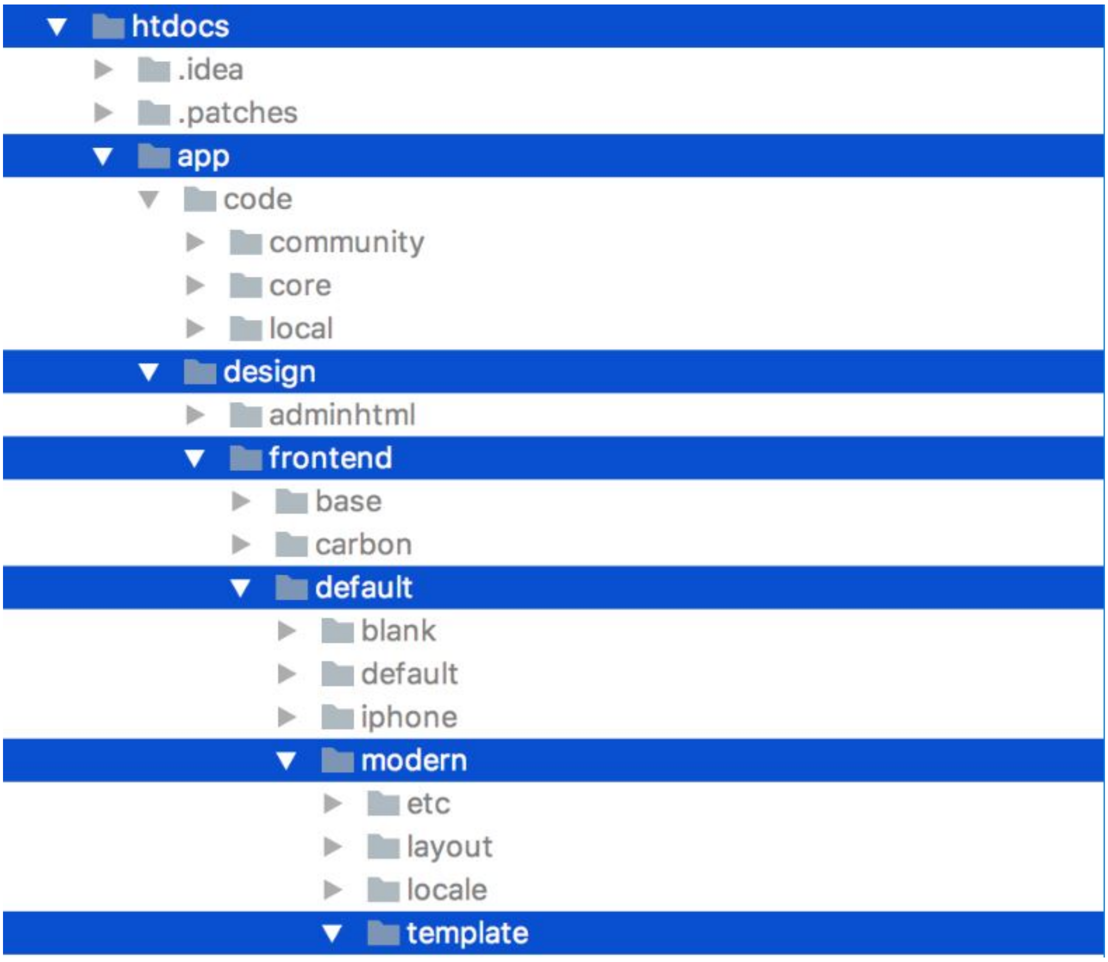


```
▼ htdocs
    ▶ .idea
    ▶ .patches
    ▼ app
        ▼ code
            ▶ community
            ▶ core
            ▶ local
        ▼ design
            ▶ adminhtml
            ▼ frontend
                ▶ base
                ▶ carbon
                ▶ default
                ▶ enterprise
                ▶ rwd
            ▶ install
        ▶ etc
        ▶ locale
          .htaccess
          bootstrap.php
          Mage.php
```

FIGURE 10. Magento Fall back structure for theme

4.1.2 Magento Core File Override

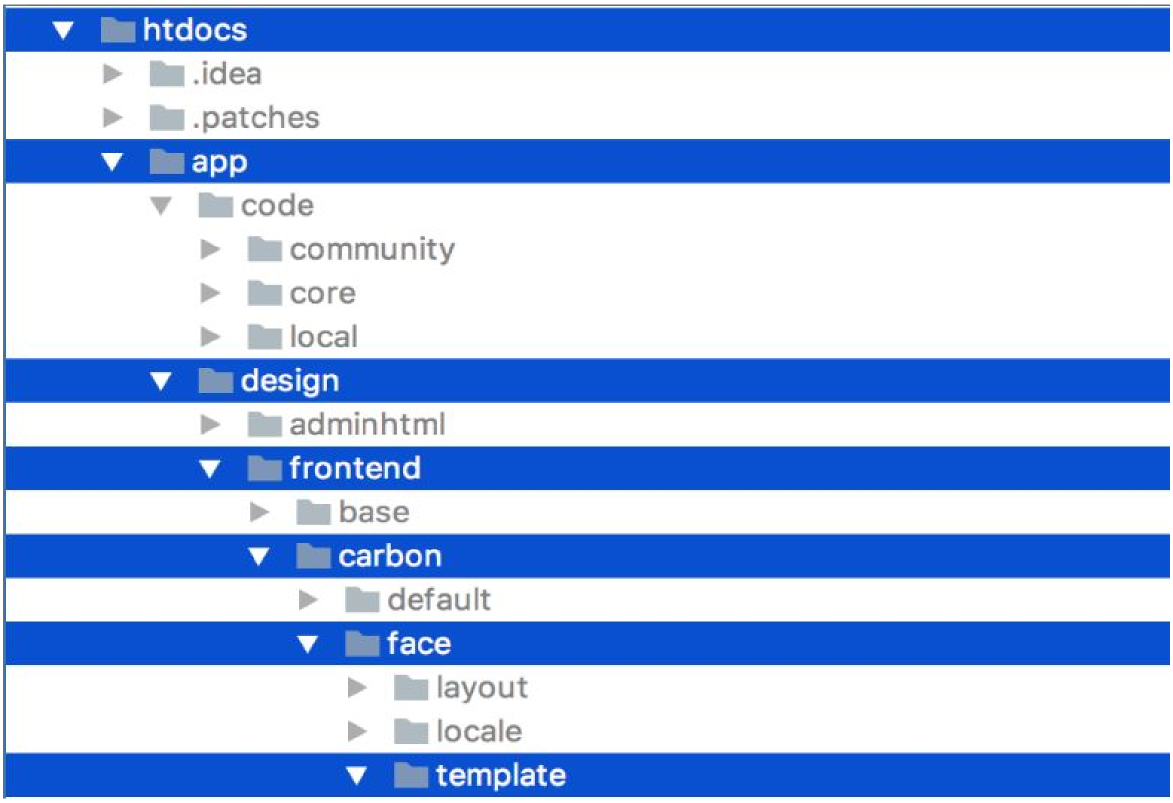
Magento provides a generic web layout and functionality which are suitable for many users. In addition, Magento also provides the possibility to extend the functionality wherever necessary. Due to this flexibility, the Magento core files could be safely overrides without the need of changing the generic default files.



```
▼ htdocs
  ▶ .idea
  ▶ .patches
  ▼ app
    ▼ code
      ▶ community
      ▶ core
      ▶ local
    ▼ design
      ▶ adminhtml
      ▼ frontend
        ▶ base
        ▶ carbon
        ▼ default
          ▶ blank
          ▶ default
          ▶ iphone
          ▼ modern
            ▶ etc
            ▶ layout
            ▶ locale
            ▼ template
```

FIGURE 11. Magento default theme folder

As shown in figure 11, Magento contains a few theme files, such as blank, default, iPhone, and modern under default package. Modern is one of the Magento theme files, which a user could use by default and without having to modify the basic theme layout and structure.



```
▼ htdocs
    ▶ .idea
    ▶ .patches
    ▼ app
        ▼ code
            ▶ community
            ▶ core
            ▶ local
        ▼ design
            ▶ adminhtml
            ▼ frontend
                ▶ base
                ▼ carbon
                    ▶ default
                    ▼ face
                        ▶ layout
                        ▶ locale
                        ▼ template
```

FIGURE 12. Magento default theme override with custom theme

In this project called YourFace, the theme name face was created under the theme package “carbon” to override the default generic Magento theme files which can be seen in figure 12. This enabled to modify and create a custom user interface without modifying the default files using less code. Therefore, in future if the Magento version and files need to be upgraded then this also enables to safely upgrade the Magento core system without any need to code again with these custom files.

5 WEB DEVELOPMENT TOOLS AND TECHNOLOGIES

Web development tools play a vital role in the web application development. They allow any web developer to test their code and debug the codes during any phase of the web development cycle. Due to the technologies, advance web development tools already come integrated in almost all popular browsers, such as Safari, Google Chrome, Firefox, Internet Explorer and Opera. These integrated web development tools are used throughout the Magento webstore application development phase.

Default web browsers in this work are Safari and Chrome. Therefore, in addition to these web development tools, there are also integrated various Chrome browser specific plugins, such as “Pesticides and Xdebug Helper” and they are used to ease the web development process. [16]

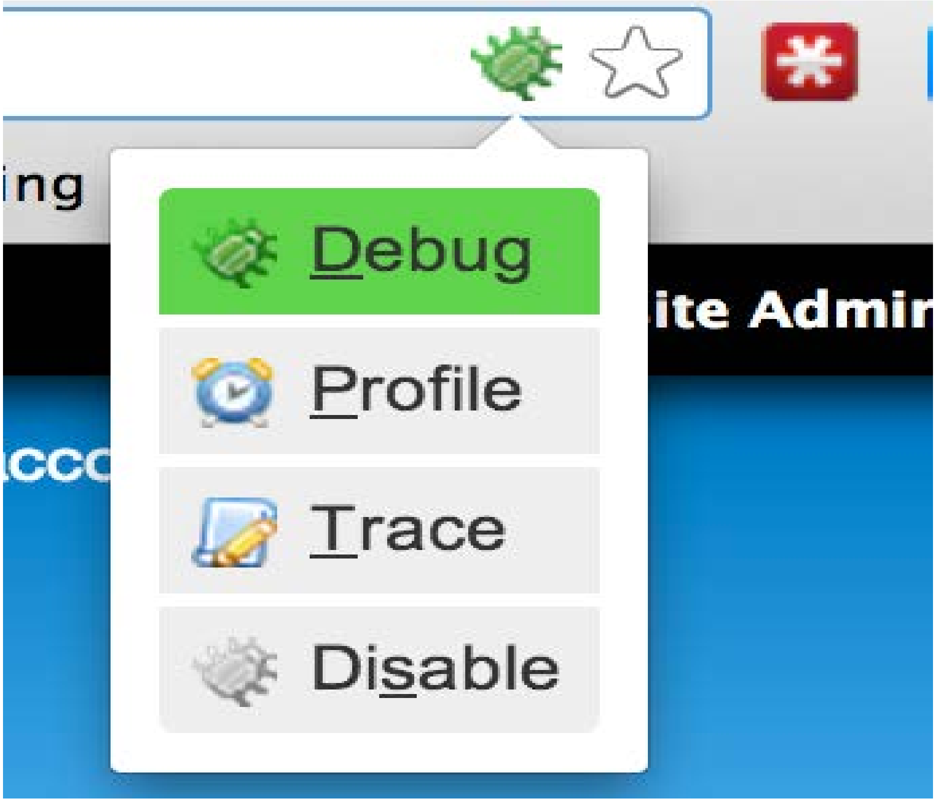
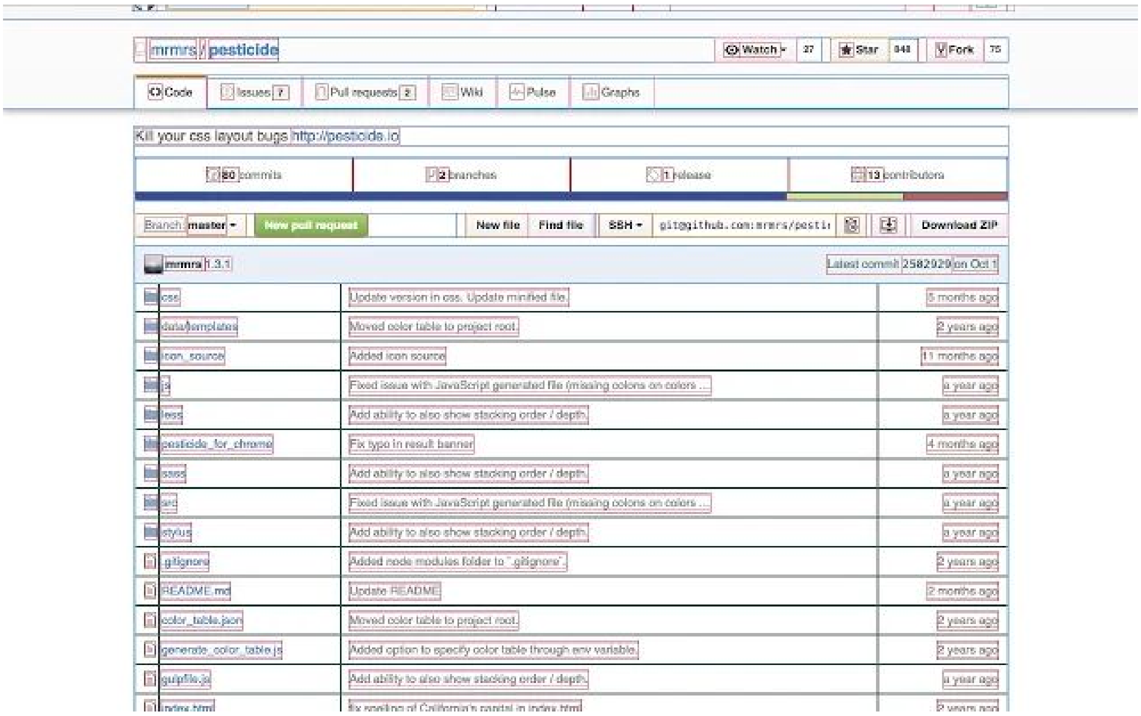


FIGURE 13. Xdebug in Chrome

The Xdebug Helper extension (shown in figure 13) provides easy debugging, profiling, and tracing capabilities. This plugin enabled to debug the PHP through PhpStorm IDE in an extensively easy manner. [16]

Likewise, Pesticides is also another extension which inserts the outline border around different html elements into the current page to better see the placement on the page. This plugin also enabled to detect all overflow elements having specific extra margins on a different layout. It creates the border around the html DOM element with different colours which help quickly to find the problematic element as shown in the figure 14. [15]



Kill your css layout bugs http://pesticide.io

git@github.com:mrmrs/pesti:

mrmrs 1.3.1

Latest commit 2582929 on Oct 1

css	Update version in css. Update minified file.	5 months ago
data/templates	Moved color table to project root.	2 years ago
icon_source	Added icon source	11 months ago
js	Fixed issue with JavaScript generated file (missing colons on colors ...	a year ago
less	Add ability to also show stacking order / depth.	a year ago
pesticide_for_chrome	Fix typo in result banner	4 months ago
sass	Add ability to also show stacking order / depth.	a year ago
src	Fixed issue with JavaScript generated file (missing colons on colors ...	a year ago
stylus	Add ability to also show stacking order / depth.	a year ago
.gitignore	Added node modules folder to ".gitignore".	2 years ago
README.md	Update README	2 months ago
color_table.json	Moved color table to project root.	2 years ago
generate_color_table.js	Added option to specify color table through env variable.	2 years ago
gulpfile.js	Add ability to also show stacking order / depth.	a year ago
index.html	fix spelling of California's capital in index.html	2 years ago

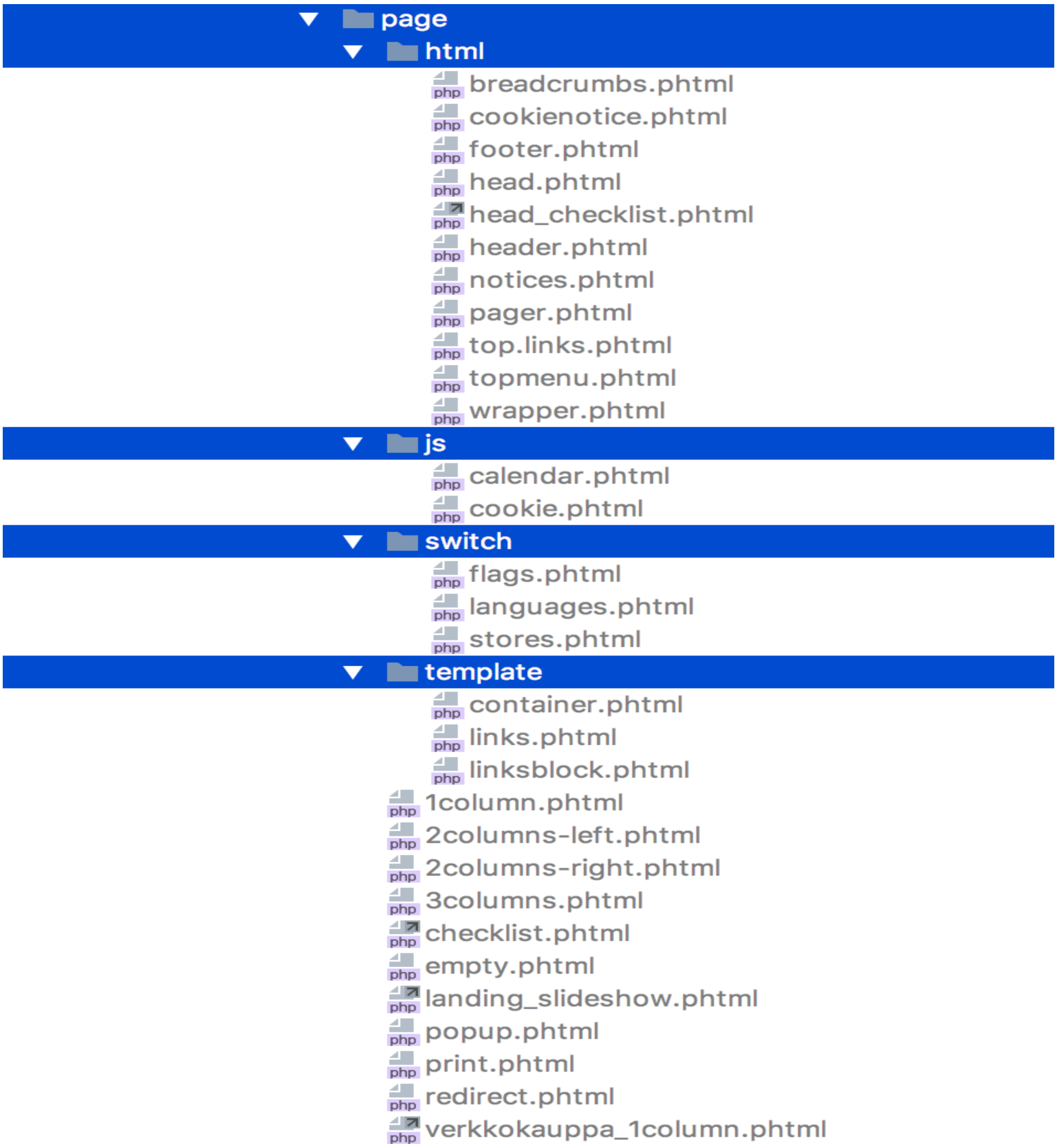
FIGURE 14. Pesticides in Chrome

It is the fact that different browsers have the capability of rendering the web application page differently. Some codes and web styles do not reflect the same on all browsers with their respective version. During the development phase the developer had to make sure that this web application was consistently rendered properly by all the browsers used by various users. Therefore, one of my responsibilities was also to test the web pages using various popular browsers and if a problem persisted, then to fix the styles and other rendering issues using these web development tools. These web development tools allowed to work with popular Magento web technologies, such as PHTML, CSS and JavaScript.

5.1 PHTML

PHTML is the suffix of the standard HTML page which also consists of the PHP code. It is widely used in the Magento web application system to render the webpage for its user. Multiple required template ***.phtml** files were created to combine and write the PHTML + PHP code. It is one of the most common and easy way of Magento system which executes the PHP code on the server side and renders the HTML code on the users' browser.

Every HTML DOM objects layout was separated and the PHP + HTML Code was written inside these files containing the ".phtml" suffix shown in the figure 15. [6][7][10]



```
▼ page
    ▼ html
        breadcrumbs.phtml
        cookienotice.phtml
        footer.phtml
        head.phtml
        head_checklist.phtml
        header.phtml
        notices.phtml
        pager.phtml
        top.links.phtml
        topmenu.phtml
        wrapper.phtml
    ▼ js
        calendar.phtml
        cookie.phtml
    ▼ switch
        flags.phtml
        languages.phtml
        stores.phtml
    ▼ template
        container.phtml
        links.phtml
        linksblock.phtml
    1column.phtml
    2columns-left.phtml
    2columns-right.phtml
    3columns.phtml
    checklist.phtml
    empty.phtml
    landing_slideshow.phtml
    popup.phtml
    print.phtml
    redirect.phtml
    verkkokauppa_1column.phtml
```

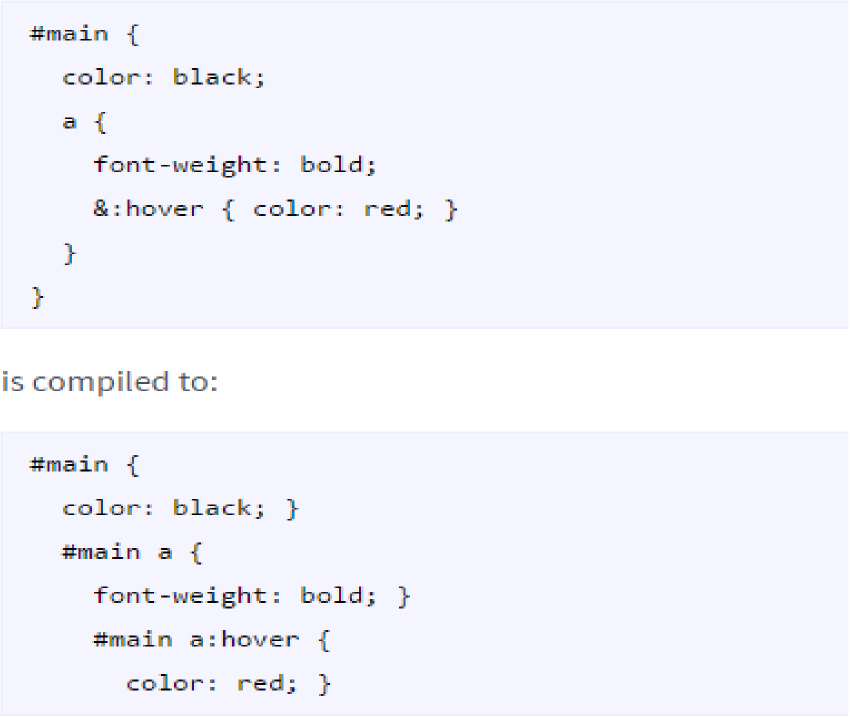
FIGURE 15. phtml template file example

5.2 CSS

Cascading Style Sheets (CSS) is a style sheet containing all the styles, colours and fonts required to render the webpage as per the design. It has played a vital role during the web application styling phase. All the necessary styles code that is needed by the HTML DOM section such as Header, body content and footer were written in multiple CSS files. Due to this separation of multiple CSS files, it has helped to load only the necessary code files quickly to the user browsers. It has also enabled to refer to the correct style sheet efficiently.

5.2.1 SCSS

During this project style implementation, the superset of CSS3's syntax extensively was used throughout the development process. The main reason behind using this advanced version of CSS3 was to have access to variables, manipulate colours, nesting CSS properties, an advanced syntax and a proper format of the indentation. SCSS is well compatible with the Magento system as after compiling the SCSS, it converts into regular CSS style sheets without any problem. [4] figure 16 shows a short SCSS snippet, which is then compiled back to a normal CSS file.



```
#main {
  color: black;
  a {
    font-weight: bold;
    &:hover { color: red; }
  }
}
```

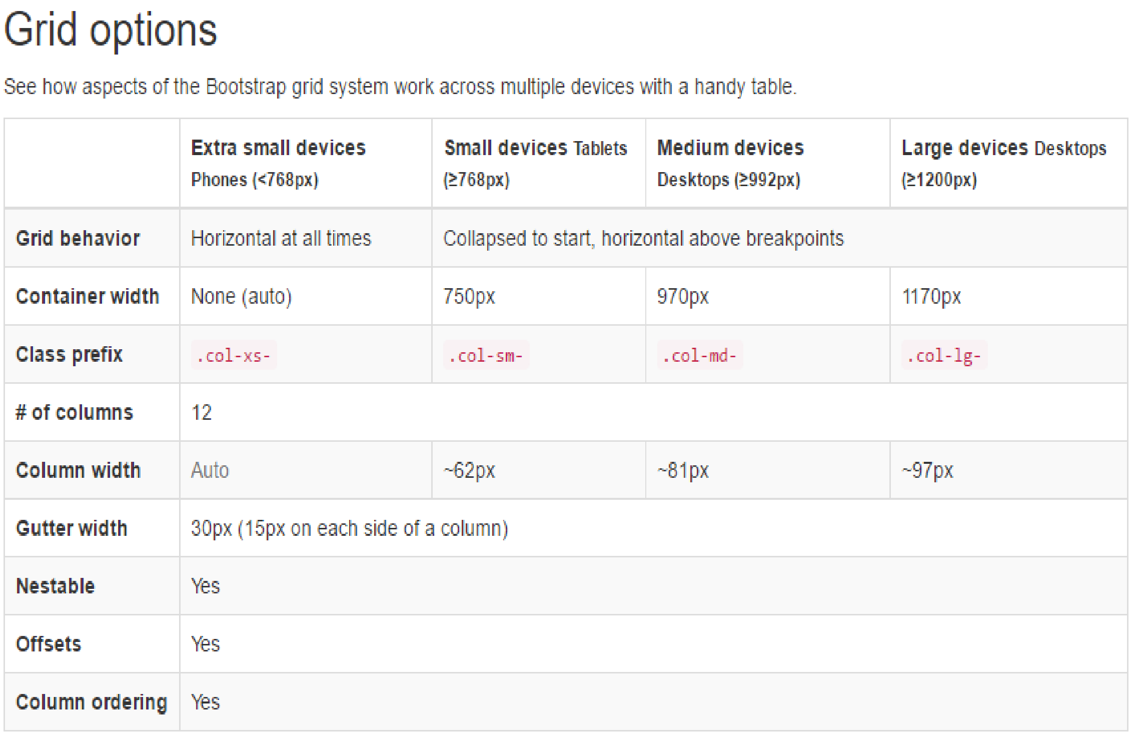
is compiled to:

```
#main {
  color: black; }
  #main a {
    font-weight: bold; }
    #main a:hover {
      color: red; }
```

FIGURE 16. SCSS compilation to CSS

5.3 Bootstrap

Bootstrap is another tool which was used throughout this project. It is also the most widely used front-end framework, which consists of HTML, CSS, and JavaScript for making the website responsive. Bootstrap and its grids and columns system were used to control the mobile and desktop shape easily. Using the Bootstrap saved time and reduced the work load because of its ability to scale between every device by utilizing a single code base. Being fully compatible with CSS pre-processors such as SASS, Bootstrap also helped to style the layout quickly. Also many predefined Bootstrap classes were used to ease the development process. [5]



Grid options

See how aspects of the Bootstrap grid system work across multiple devices with a handy table.

	Extra small devices Phones (<768px)	**Small devices** Tablets (≥768px)	**Medium devices** Desktops (≥992px)	**Large devices** Desktops (≥1200px)
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints		
Container width	None (auto)	750px	970px	1170px
Class prefix	`.col-xs-`	`.col-sm-`	`.col-md-`	`.col-lg-`
# of columns	12			
Column width	Auto	~62px	~81px	~97px
Gutter width	30px (15px on each side of a column)			
Nestable	Yes			
Offsets	Yes			
Column ordering	Yes			

FIGURE 17. Bootstrap grid layout to scale between every device [5]

FIGURE 17 shows the pixel amounts used between extra small devices and large devices desktops. These Bootstrap properties were used to control the content between various media.

5.4 JavaScript

JavaScript is a high-level programming language which is compatible with all the major and popular browsers. With the help of JavaScript, it was possible to do various activities quickly and efficiently directly in the user's browser with or without user interactions.

To make the development process more convenient and to manipulate the HTML DOM, the famous JavaScript library, such as jQuery, SelectorDie, OwlCarousel, were used.

jQuery was used extensively for all kind of event handling and ajax request. For instance, using the jQuery JavaScript library it was possible to display and configure the search page result for the users. Therefore, when users click on the search Icon shown in figure18, then the search bar gets populated as shown in figure 19. From the user interface point of view, the search icon will be replaced with a close icon, which can also be seen in figure19. jQuery helped in creating these functionality in a simpler and easier manner.

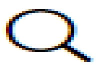


FIGURE 18. Search Icon and search bar before click

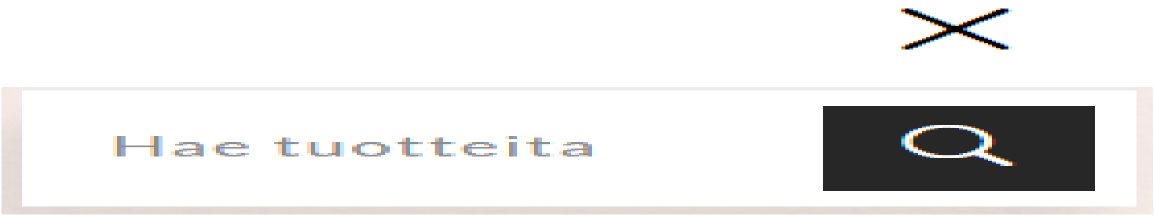
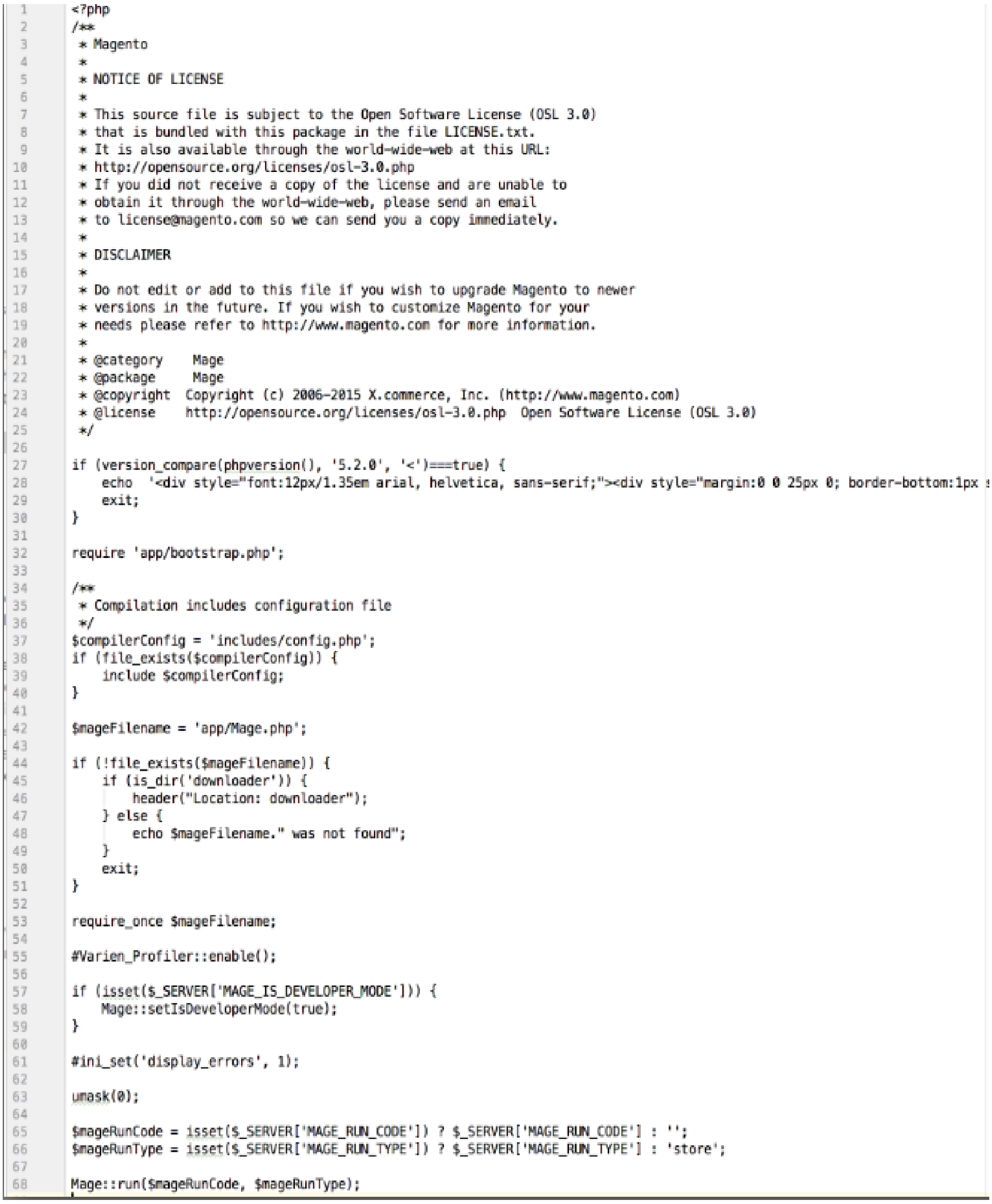


FIGURE 19. Search Icon and search bar after click

5.5 PHP

Hypertext Preprocessor(PHP) is also one of the high-level scripting language it is open source and widely used during the web application development. PHP was used in the project as a backbone of the application to handle and control all the backend logic and functionality. [6][7]

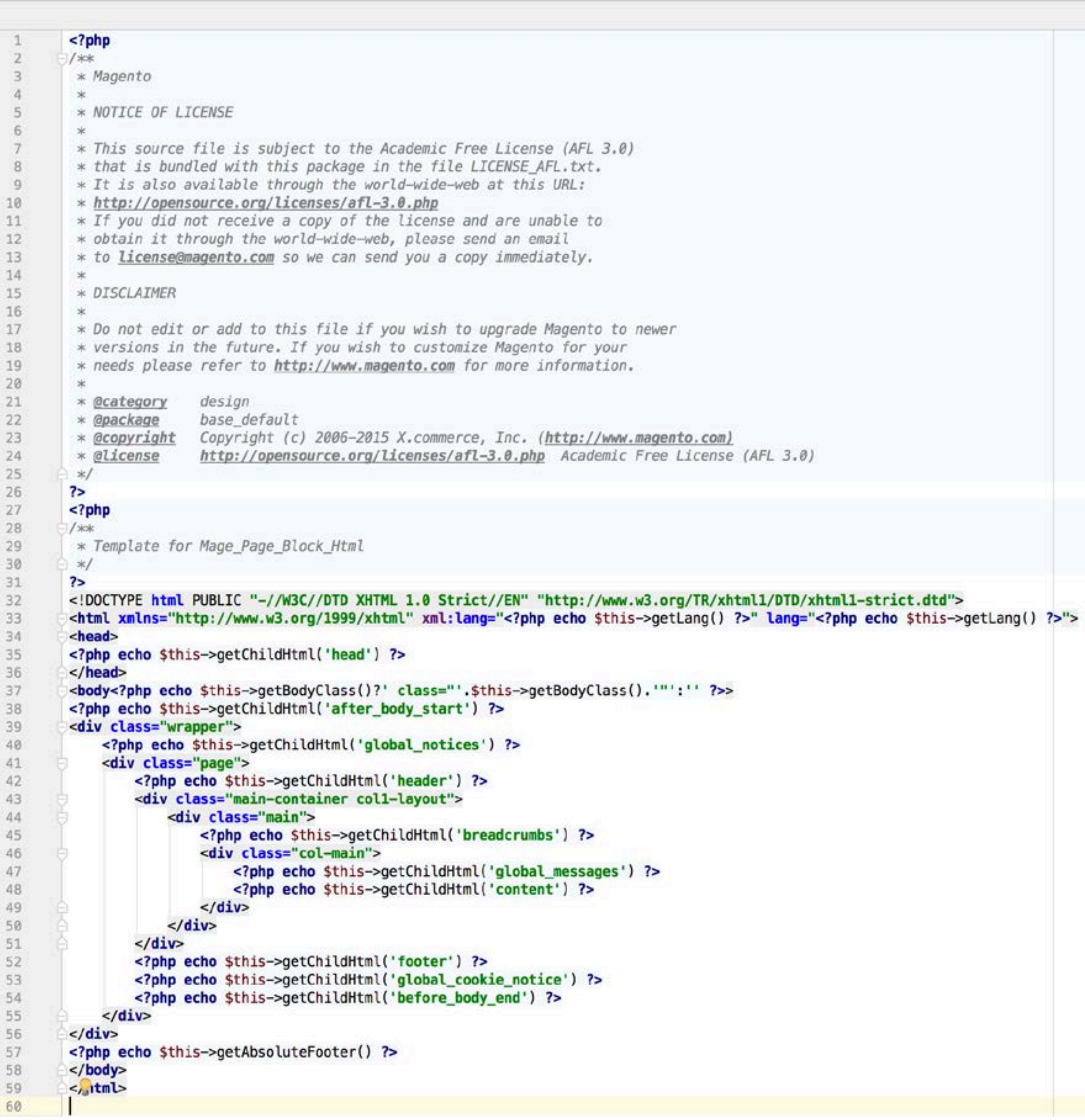


```php
<?php
/**
 * Magento
 *
 * NOTICE OF LICENSE
 *
 * This source file is subject to the Open Software License (OSL 3.0)
 * that is bundled with this package in the file LICENSE.txt.
 * It is also available through the world-wide-web at this URL:
 * http://opensource.org/licenses/osl-3.0.php
 * If you did not receive a copy of the license and are unable to
 * obtain it through the world-wide-web, please send an email
 * to license@magento.com so we can send you a copy immediately.
 *
 * DISCLAIMER
 *
 * Do not edit or add to this file if you wish to upgrade Magento to newer
 * versions in the future. If you wish to customize Magento for your
 * needs please refer to http://www.magento.com for more information.
 *
 * @category    Mage
 * @package     Mage
 * @copyright  Copyright (c) 2006-2015 X.commerce, Inc. (http://www.magento.com)
 * @license    http://opensource.org/licenses/osl-3.0.php  Open Software License (OSL 3.0)
 */

if (version_compare(phpversion(), '5.2.0', '<')===true) {
    echo  '<div style="font:12px/1.35em arial, helvetica, sans-serif;"><div style="margin:0 0 25px 0; border-bottom:1px
    exit;
}

require 'app/bootstrap.php';

/**
 * Compilation includes configuration file
 */
$compilerConfig = 'includes/config.php';
if (file_exists($compilerConfig)) {
    include $compilerConfig;
}

$mageFilename = 'app/Mage.php';

if (!file_exists($mageFilename)) {
    if (is_dir('downloader')) {
        header("Location: downloader");
    } else {
        echo $mageFilename." was not found";
    }
    exit;
}

require_once $mageFilename;

#Varien_Profiler::enable();

if (isset($_SERVER['MAGE_IS_DEVELOPER_MODE'])) {
    Mage::setIsDeveloperMode(true);
}

#ini_set('display_errors', 1);

umask(0);

$mageRunCode = isset($_SERVER['MAGE_RUN_CODE']) ? $_SERVER['MAGE_RUN_CODE'] : '';
$mageRunType = isset($_SERVER['MAGE_RUN_TYPE']) ? $_SERVER['MAGE_RUN_TYPE'] : 'store';

Mage::run($mageRunCode, $mageRunType);
```

FIGURE 20. Magento Index.php

This Magento Index.php file, as shown in the figure 20, handles the MVC system which examines the URL, and based on the set of rules, it routes the requests into an appropriate controller class and dispatches it. This is the first file that gets executed in the server when a user requests a specific URL.

During the implementation, many times the developer was also required to embed the PHP code inside HTML DOM in order to execute a certain functionality. Since, PHP is directly executed in the server side, it is a very easy way to embed inside HTML DOM.



```
<?php
/**
 * Magento
 *
 * NOTICE OF LICENSE
 *
 * This source file is subject to the Academic Free License (AFL 3.0)
 * that is bundled with this package in the file LICENSE_AFL.txt.
 * It is also available through the world-wide-web at this URL:
 * http://opensource.org/licenses/afl-3.0.php
 * If you did not receive a copy of the license and are unable to
 * obtain it through the world-wide-web, please send an email
 * to license@magento.com so we can send you a copy immediately.
 *
 * DISCLAIMER
 *
 * Do not edit or add to this file if you wish to upgrade Magento to newer
 * versions in the future. If you wish to customize Magento for your
 * needs please refer to http://www.magento.com for more information.
 *
 * @category    design
 * @package     base_default
 * @copyright   Copyright (c) 2006-2015 X.commerce, Inc. (http://www.magento.com)
 * @license     http://opensource.org/licenses/afl-3.0.php  Academic Free License (AFL 3.0)
 */
?>
<?php
/**
 * Template for Mage_Page_Block_Html
 */
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $this->getLang() ?>" lang="<?php echo $this->getLang() ?>">
<head>
<?php echo $this->getChildHtml('head') ?>
</head>
<body<?php echo $this->getBodyClass()?' class="'.$this->getBodyClass().'"':'' ?>>
<?php echo $this->getChildHtml('after_body_start') ?>
<div class="wrapper">
    <?php echo $this->getChildHtml('global_notices') ?>
    <div class="page">
        <?php echo $this->getChildHtml('header') ?>
        <div class="main-container col1-layout">
            <div class="main">
                <?php echo $this->getChildHtml('breadcrumbs') ?>
                <div class="col-main">
                    <?php echo $this->getChildHtml('global_messages') ?>
                    <?php echo $this->getChildHtml('content') ?>
                </div>
            </div>
        </div>
        <?php echo $this->getChildHtml('footer') ?>
        <?php echo $this->getChildHtml('global_cookie_notice') ?>
        <?php echo $this->getChildHtml('before_body_end') ?>
    </div>
</div>
<?php echo $this->getAbsoluteFooter() ?>
</body>
</html>

```

FIGURE 21. Magento base default 1column.phtml

After the execution of the index.php in the server side, the requests also get routed and dispatched through Magento template files depending on the number of columns present on the page. For instance, figure 21 above shows the requests being landed for the page with 1 column. In this file, various PHP code is being embedded and they get executed in the server side. After the execution is finished, the functionality is provided to the user’s browser. Therefore, this file handles the request and provides all the required contents, such as “header”, “body content”, “footer” and other classes which were customized and programmed in other files as per the design.

During the backend functionality implementation, the author encountered numerous errors which were sometimes extremely tricky to fix in the project scope. Several times the author had to check the exception log in the server to figure out the problematic script. Besides those tricky situations, the IDE based debugging utility and browser based plugin for debugging the specific codes were always handy for the deployment and testing of all backend functionality codes.

In order to have the latest security patch and to execute the PHP script, the latest stable version of PHP version 7 was also installed in the server side.

5.6 Version Control

Version control is a system which keeps track on all the changes and records that are done with each specific version so that the programmer could easily track the changes and manipulate them if needed. These days every application development company is equipped with a different version control software. This will even facilitate the developer and it will help the organization to keep track of codes with every released version. There are many version control systems available, but Mercurial tools and Bitbucket repositories were used in this project to control the workflow. Version control tools and repositories played a very important role during the project. The author was able to develop features in the feature branch. Whenever a problem arose, it was possible to create a patch with a hotfix branch without any conflicts to the master branch. The author was also able to release the branch when the implementation was done and tested with specific codes and functionality. [8][9] figure 22, shows the workflow of different branching system in the version control system.

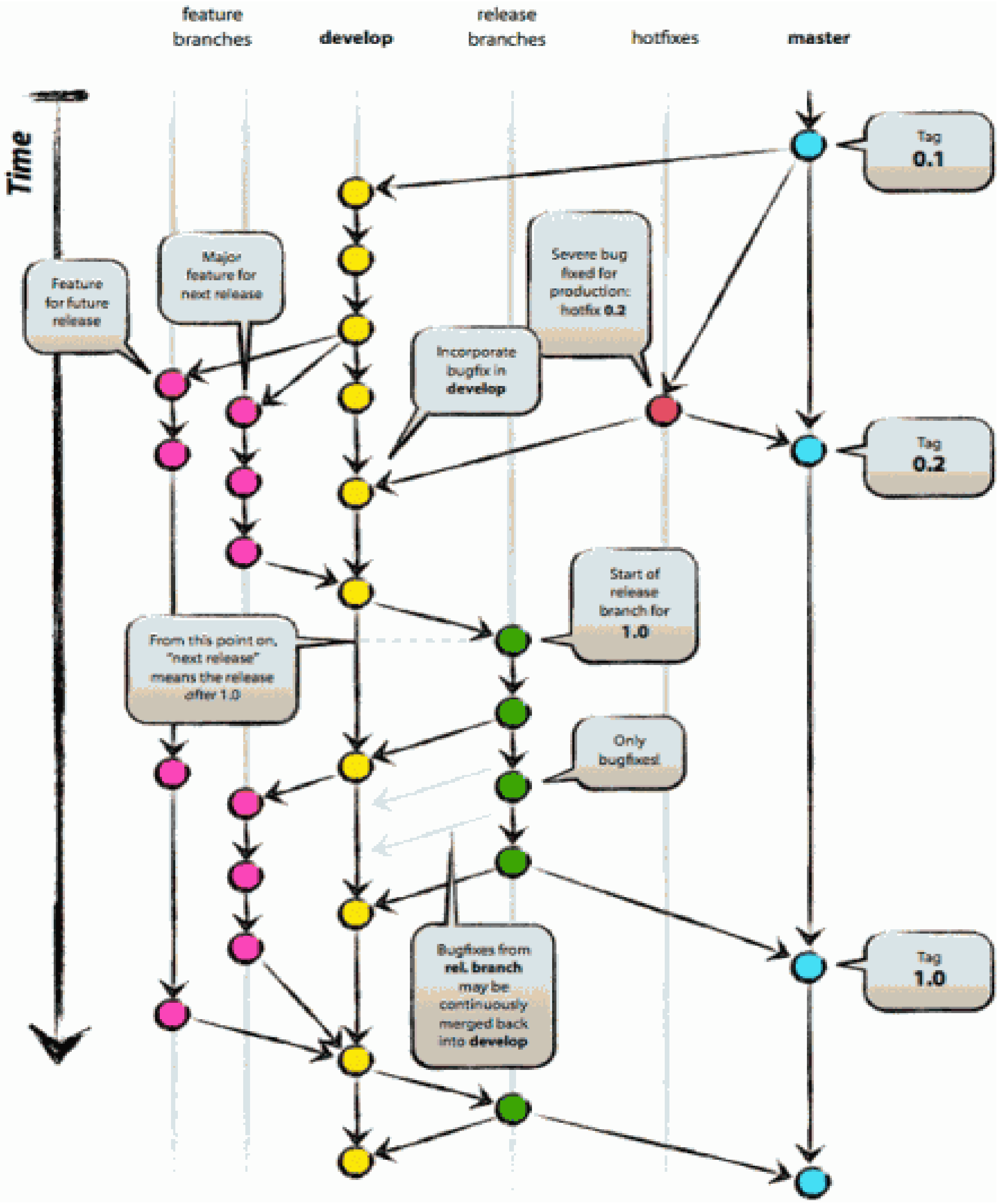

feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Feature for future release
Major feature for next release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0

FIGURE 22. Version Control System workflow [9]

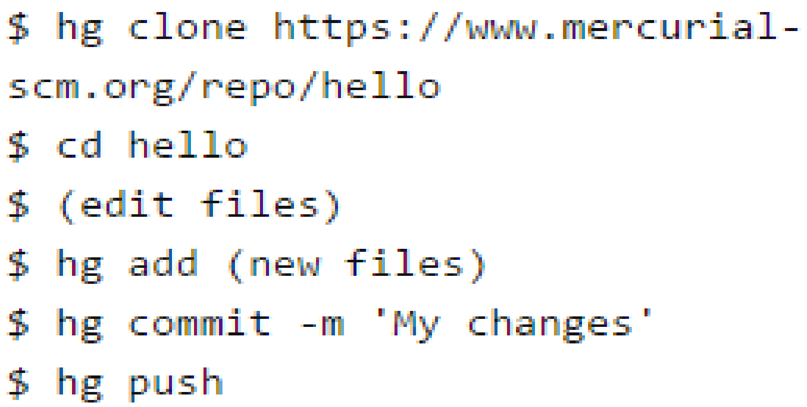
5.6.1 Jira (Atlassian)

Jira is a bug issues tracking system and project management system pioneered by Atlassian. Jira Kanban board was used to track and update the project status. Also, Atlassian bitbucket repositories were used to tag the bitbucket repositories to the assigned tasks multiple times a day.

5.6.2 Mercurial (hg)

Mercurial is one of the powerful and fastest project version management tools. As shown in the figures 23 and 24 below, it helps in cloning the project and tracking the project commits version during various development and production phases. It supports multiple extensions which enhanced the work environment during the development of the e-commerce store.

Mercurial is open source and it has a capability of handling large size projects. This tool was used throughout the project very efficiently. Since Mercurial supports both the command line tools and GUI interface for an easy management and versioning, it was very easy to track what changes were made between the different commits and history. If some codes were wrongly input or some function did not work then Mercurial provided with the capability of safely switching back to the previous working state.



```
$ hg clone https://www.mercurial-
scm.org/repo/hello
$ cd hello
$ (edit files)
$ hg add (new files)
$ hg commit -m 'My changes'
$ hg push
```

FIGURE 23. Cloning and pushing changes during the development

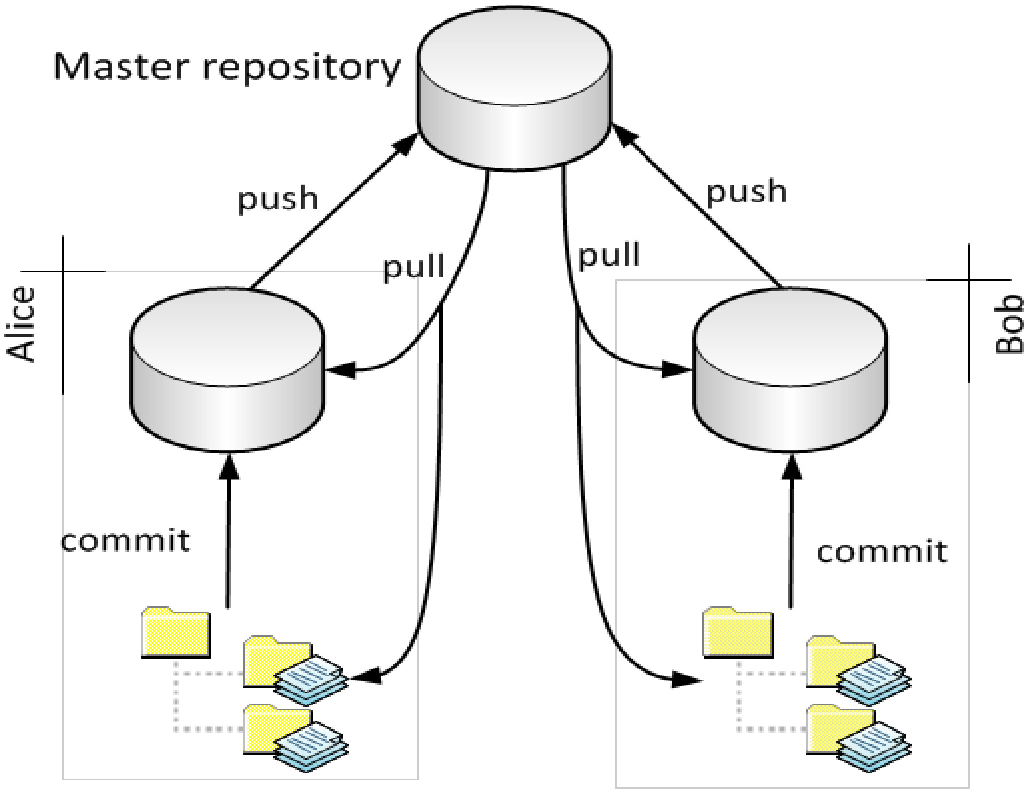


FIGURE 24. Distribution workflow [8]

5.7 Required Tools

Below is a description of tools which are used throughout the application development process.

- PHPStorm: It is the IDE specially designed for PHP to work with popular systems, such as Magento and other frameworks. It has many features including an intelligent coding assistance. PHPstorm IDE was used to write the application code heavily.

- iTerm: iTerm was used as a Mac alternative terminal to use all the CLI based commands as it is feature enriched.

- VirtualBox: It is a full virtualization application software to run the virtual operating system in the host machine. It was used to run the Ubuntu Linux operating system as the localhost server in the machine. This way the Magento e-commerce system could be easily run as a development server and run all the tests could be run before pushing to the production server.

- Vagrant: It is a tool used by developers, operators, designers and many others for managing and creating the virtual machine environments in the fastest way. Using Vagrant it was possible to control the Linux virtual operating system and work with many shortcut commands that were configured. It helped to work faster and to coordinate with the team members efficiently whenever necessary.

- Spectacleapp: It is a simple application which gives an ability for the Mac windows to resize and move with different keyboard shortcuts keys. During the implementation of the mobile design this tool provided a great help to resize the windows as per the different size mobiles. It also enabled to verify the implemented code matching with the design.

6 DESKTOP VERSION

During the project implementation, the author have received the User interface design that needed to be implemented for rendering in the browser of the client side. The author had to create a fluid grid container e-commerce web application, which would not break under any other different media devices such as tablets and mobiles. Therefore, considering these requirements it was started with the header section and the Bootstrap grid column system was used to get the content fit under different media.

6.1 Header content and Floating Header

The floating header has been one of the very basic needs in many e-commerce stores. The floating header provides the flexibility for the users to navigate through the header content without needs of scrolling back to top when they are browsing and scrolling down the content. During the implementation which is seen in figure 25, the logo was placed to the left and the header menu to the middle and social media icons to the right corner. With the help of CSS properties, the header section was styled with all the necessary font-size and margins. Therefore, checking all the acceptance criteria the header position was fixed and made to float across the browser and screen of various media. [21]

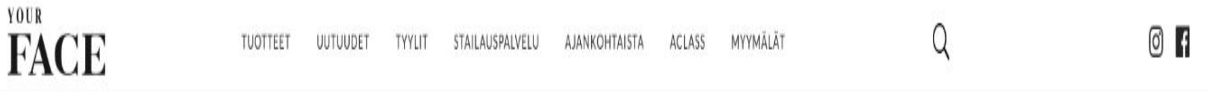
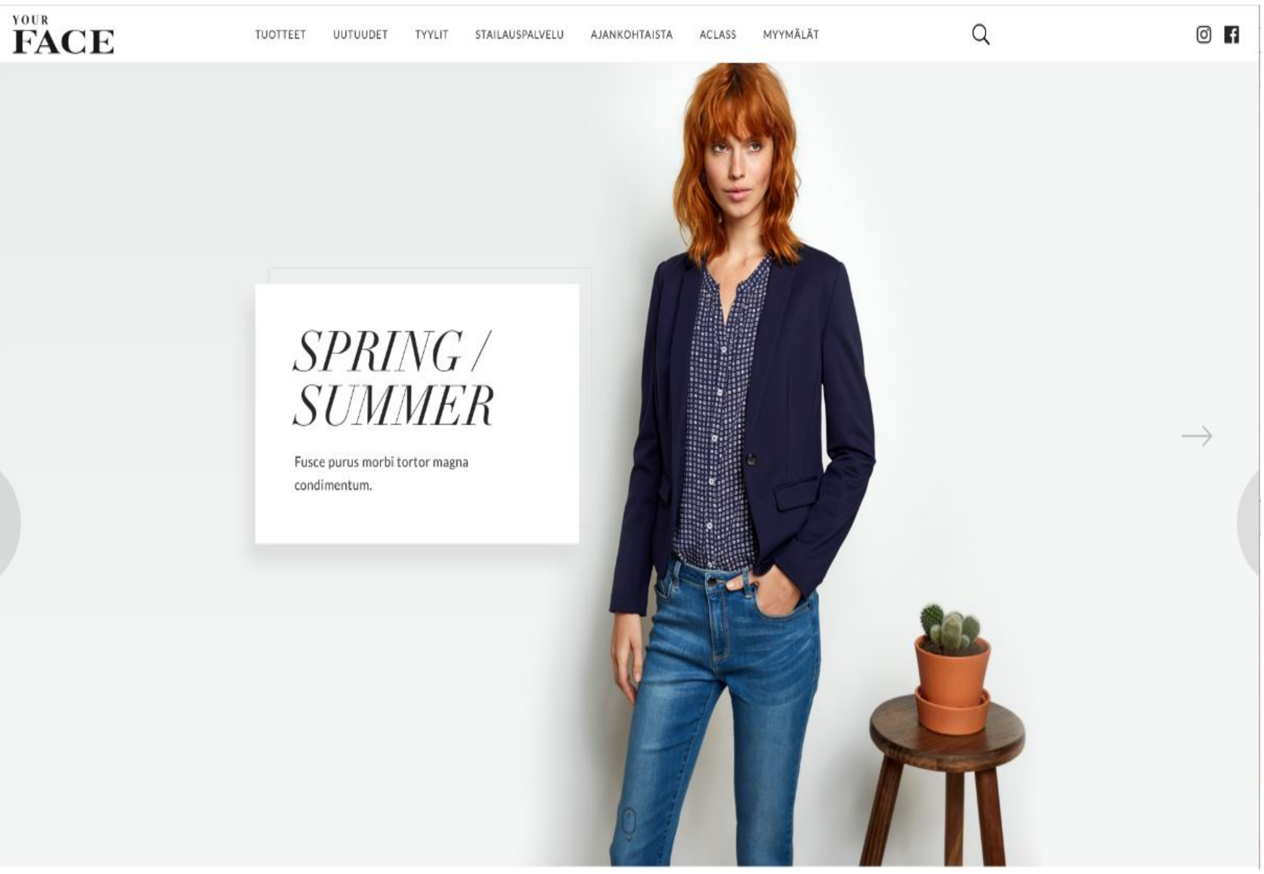


FIGURE 25. Floating header [21]

6.2 Slideshow Manager

Magento has a very tight integration and compatibility with Bootstrap. One of the acceptance criteria for this section was to make a slideshow content with the text banner on top of the image. In this project, different Magento blocks were utilized to build the webpage. One of them is a slideshow block. Bootstrap carousel plugin was used to make the slideshow. Also, many CSS properties were used to a specific margin, a font family and to make it look similar to the design. The problem encountered during the implementation was to setup SASS compiling tools and to utilize them in the first place. After learning the tips and tricks of SASS, it became extremely helpful during nesting any CSS properties, using a wide variety of different level selectors quickly and efficiently [4][21]. The results can be seen in figure 26 below.



SPRING / SUMMER

Fusce purus morbi tortor magna condimentum.

FIGURE 26. Desktop Slideshow section [21]

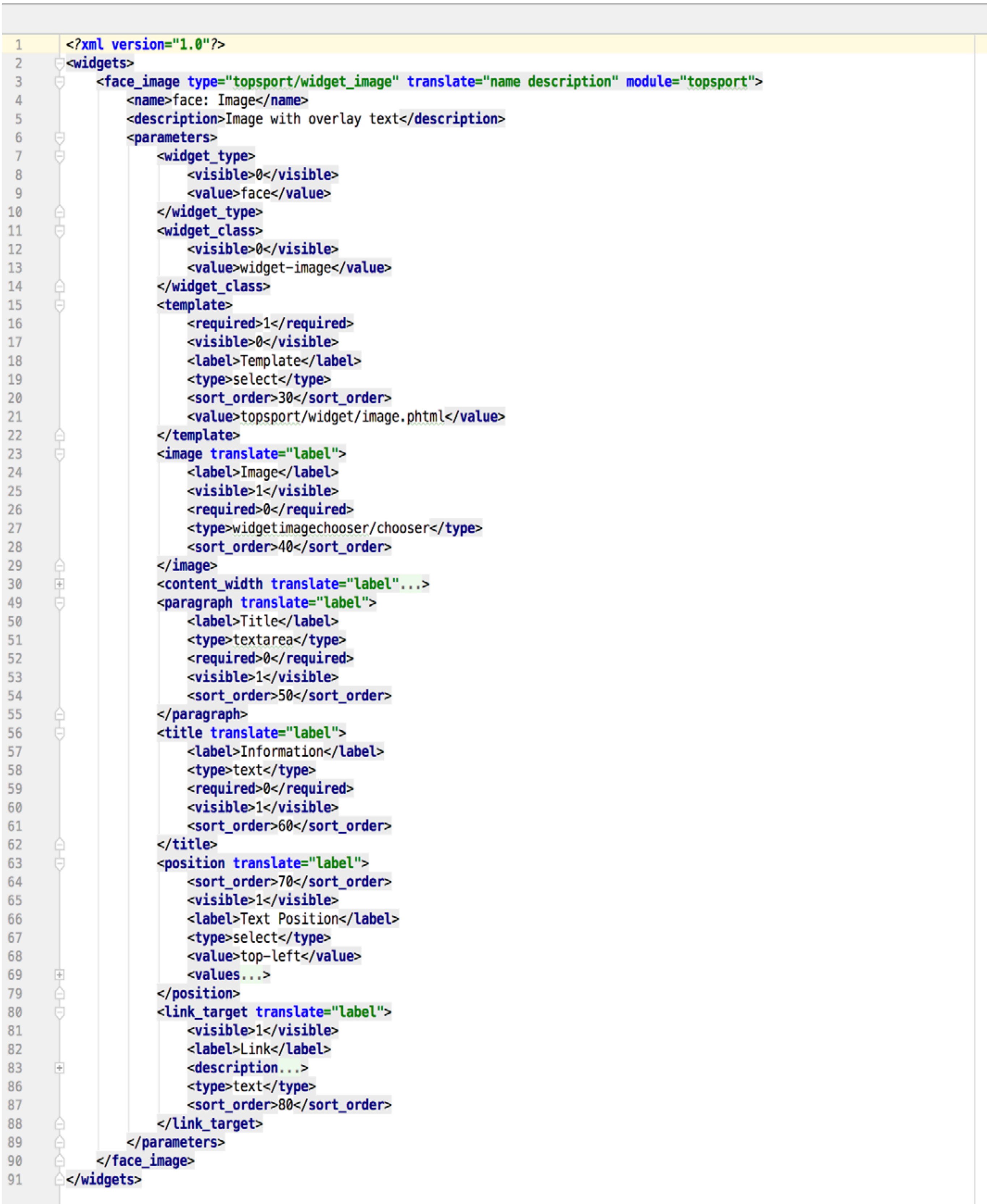
6.3 CMS Editor

Content Management System (CMS) is a powerful utility that was integrated with Vaimo CMS in the YourFace project to give the users a possibility to create, update, delete and modify the content without having a need to code. This way less technical users gets a flexibility to modify the web contents easily from on screen options. With the help of this CMS editor, users are also able to create the static block for inputting the contents directly through the CMS editor.

6.4 Banner Widgets

Widgets is a special block in Magento where different kind of content can be placed. Different kinds of widgets have the capability of handling different kind of content, such as an image with a banner text for a promotional event, displaying real time data, or creating dynamic data and giving the customer possibilities to interact.

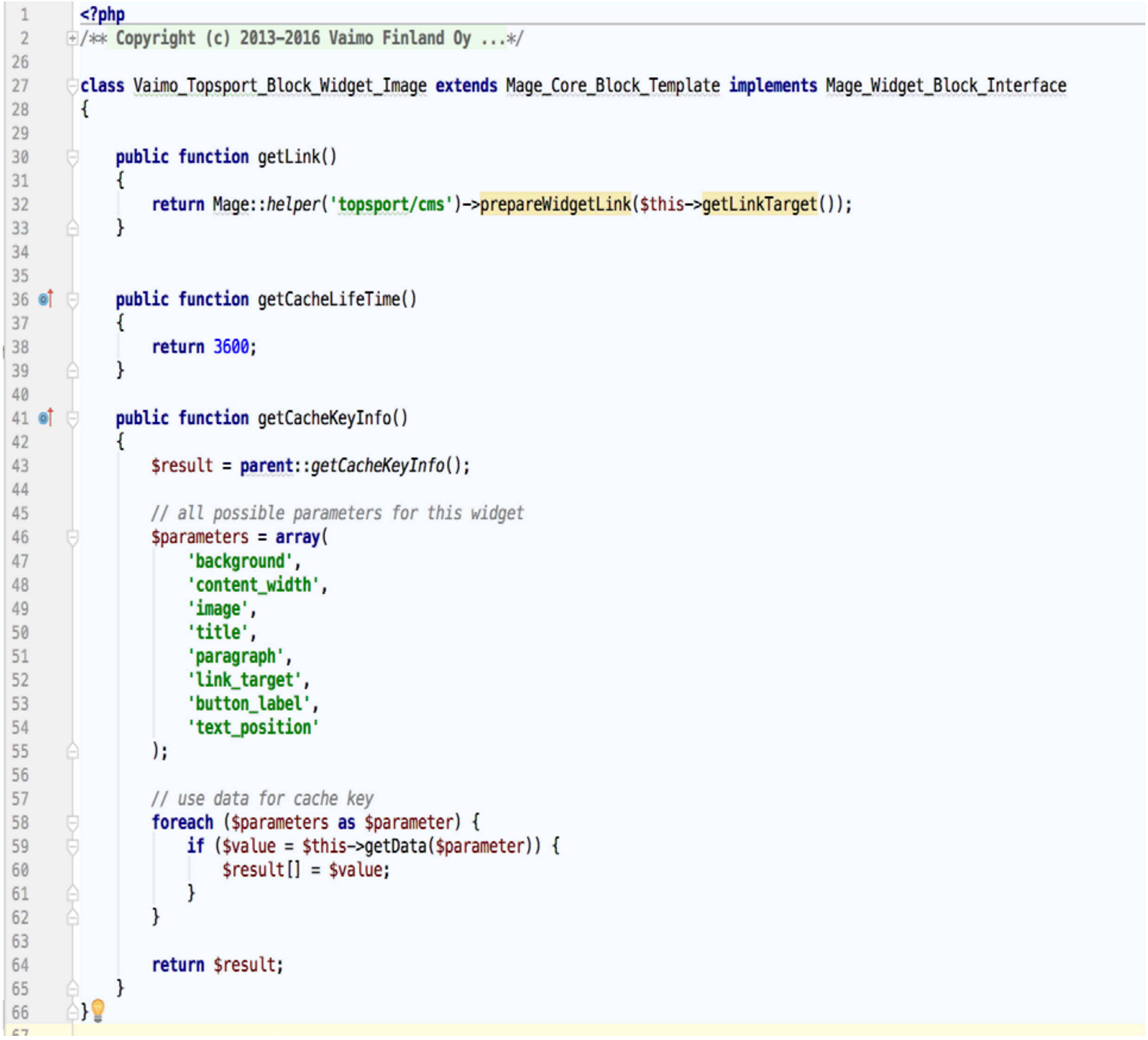
The banner widgets were created to give the users a possibility to select the text position and edit the text as per necessity from the CMS editor. In addition to these features the author also implemented the features for the customer to give a more flexibility to upload a picture during the promotional event and to keep it as a banner text image from the widgets.



```xml
<?xml version="1.0"?>
<widgets>
    <face_image type="topsport/widget_image" translate="name description" module="topsport">
        <name>face: Image</name>
        <description>Image with overlay text</description>
        <parameters>
            <widget_type>
                <visible>0</visible>
                <value>face</value>
            </widget_type>
            <widget_class>
                <visible>0</visible>
                <value>widget-image</value>
            </widget_class>
            <template>
                <required>1</required>
                <visible>0</visible>
                <label>Template</label>
                <type>select</type>
                <sort_order>30</sort_order>
                <value>topsport/widget/image.phtml</value>
            </template>
            <image translate="label">
                <label>Image</label>
                <visible>1</visible>
                <required>0</required>
                <type>widgetimagechooser/chooser</type>
                <sort_order>40</sort_order>
            </image>
            <content_width translate="label"...>
            <paragraph translate="label">
                <label>Title</label>
                <type>textarea</type>
                <required>0</required>
                <visible>1</visible>
                <sort_order>50</sort_order>
            </paragraph>
            <title translate="label">
                <label>Information</label>
                <type>text</type>
                <required>0</required>
                <visible>1</visible>
                <sort_order>60</sort_order>
            </title>
            <position translate="label">
                <sort_order>70</sort_order>
                <visible>1</visible>
                <label>Text Position</label>
                <type>select</type>
                <value>top-left</value>
                <values...>
            </position>
            <link_target translate="label">
                <visible>1</visible>
                <label>Link</label>
                <description...>
                <type>text</type>
                <sort_order>80</sort_order>
            </link_target>
        </parameters>
    </face_image>
</widgets>
```

FIGURE 27. Banner widgets XML parameter

As shown in figure 27 above, during the implementation of this banner widget, it was also named with a widget name “face Image”. After that a list of required parameters, such as a widget type, a widget class, a template, an image, a content width, a paragraph, a title, a position and a link were defined as different options for the user. Within these sets of parameters, also the functionality for labelling, visibility, requirement, value, type, and order positioning was provided.



```
<?php
/** Copyright (c) 2013-2016 Vaimo Finland Oy ...*/

class Vaimo_Topsport_Block_Widget_Image extends Mage_Core_Block_Template implements Mage_Widget_Block_Interface
{

    public function getLink()
    {
        return Mage::helper('topsport/cms')->prepareWidgetLink($this->getLinkTarget());
    }


    public function getCacheLifeTime()
    {
        return 3600;
    }

    public function getCacheKeyInfo()
    {
        $result = parent::getCacheKeyInfo();

        // all possible parameters for this widget
        $parameters = array(
            'background',
            'content_width',
            'image',
            'title',
            'paragraph',
            'link_target',
            'button_label',
            'text_position'
        );

        // use data for cache key
        foreach ($parameters as $parameter) {
            if ($value = $this->getData($parameter)) {
                $result[] = $value;
            }
        }

        return $result;
    }
}
```

FIGURE 28. Banner Widget Class file

After finishing the list of parameters in the xml file, the author created the functionality in php file shown in figure 28 to call and implement the Magento widget block interface by extending the Mage core block template. Also a list of arrays for passing data through the controller function was created. The author used different array parameters, such as a background colour option, a content width for making it full width of fixed half width, an image for the user to choose different images, a title for inputting the text title, paragraph to put the key information about the promotion phase, a link target for users to redirect to a certain link when they click, a button label for giving a name of the button, and a text position to position to a banner text either to the left corner or the right bottom. [21]

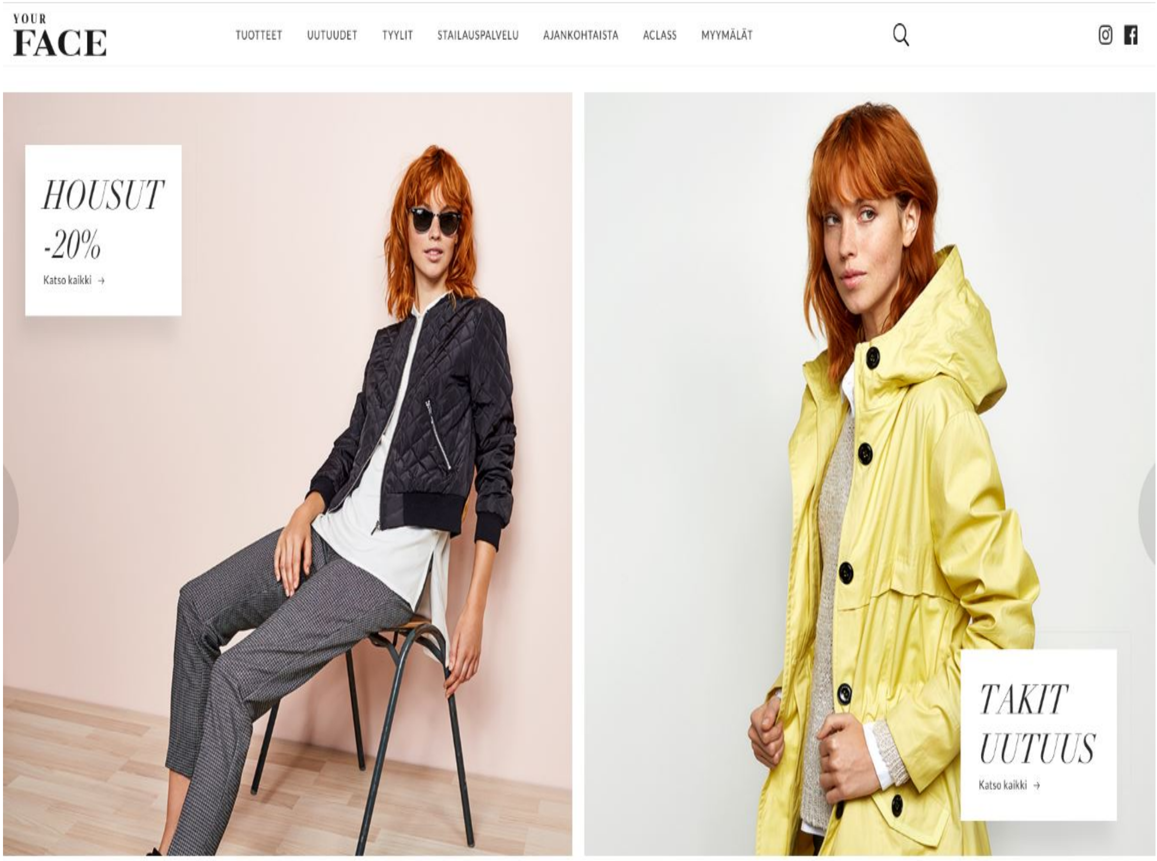
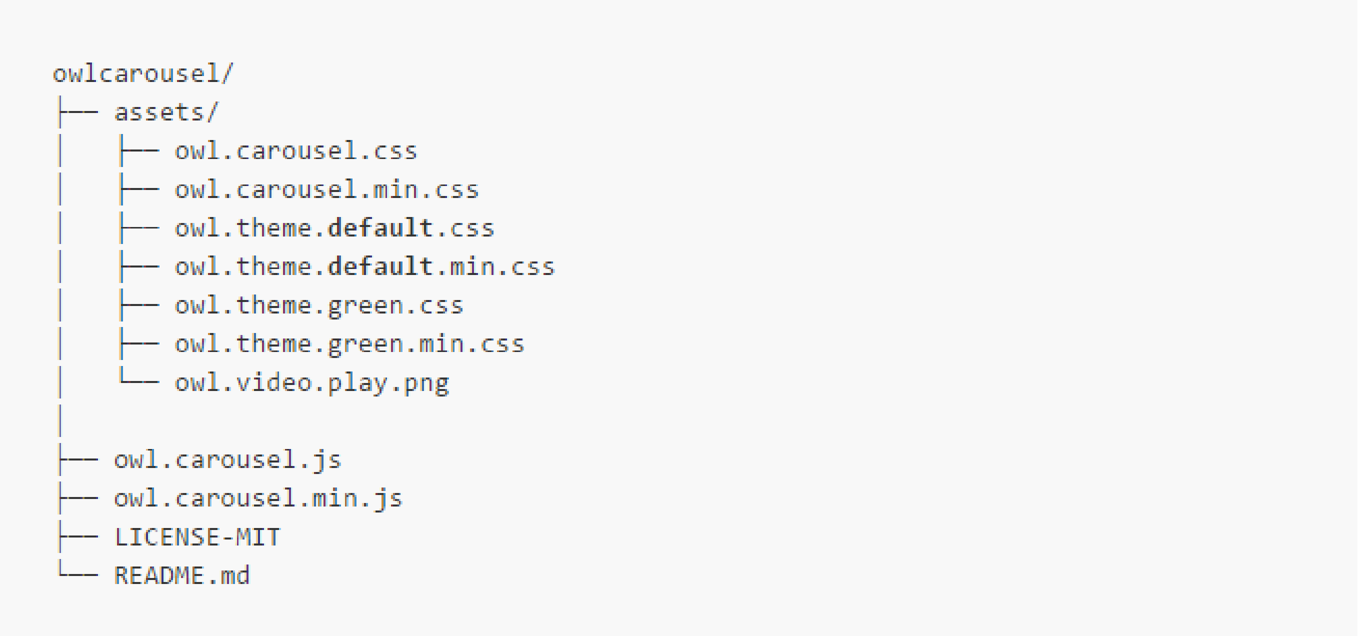


FIGURE 29. Banner Widget [21]

Figure 29 above shows an example of the final output of the implementation of banner widgets. It has many helpful features for the users, for instance, they can keep two different pictures positioning on the left hand side and the right hand side with the custom floating text banner on top. The author also implemented the options for the users to input a link for “katso kaikki” reference. [21]

6.5 Selected Product Widgets

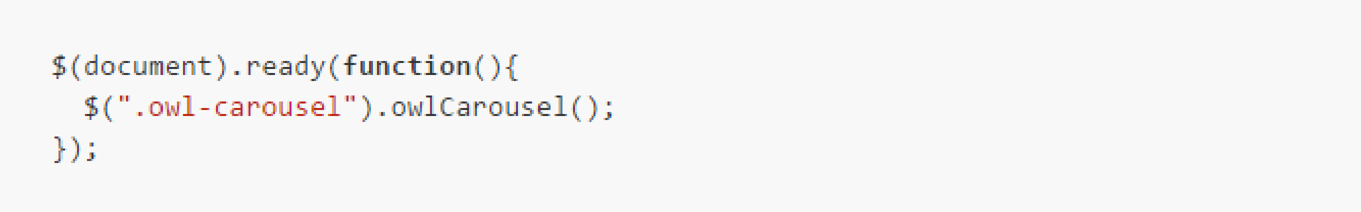
Selected products are one of the most essential widgets that many e-commerce stores demand to have it. The reason behind this is that the users have a possibility to create the different categories for the selected products and to display them with the help of the selected product widgets. The selected product widgets were implemented to give the users a flexible option to keep the different selected products listing on the frontpage. When the user creates the selected products, they are visible with a 4 grid horizontal listing at a time. If the users input more than 4 products, then the rest of the products are visible when the users scroll horizontally or click the left right arrow. Also, the author used the owl carousel, i.e. a third party JavaScript library which gives the mobile users a possibility to have the swap gesture.



```
owlcarousel/
├── assets/
│    ├── owl.carousel.css
│    ├── owl.carousel.min.css
│    ├── owl.theme.default.css
│    ├── owl.theme.default.min.css
│    ├── owl.theme.green.css
│    ├── owl.theme.green.min.css
│    └── owl.video.play.png
│
├── owl.carousel.js
├── owl.carousel.min.js
├── LICENSE-MIT
└── README.md
```

FIGURE 30. Owl Carousel files [11]

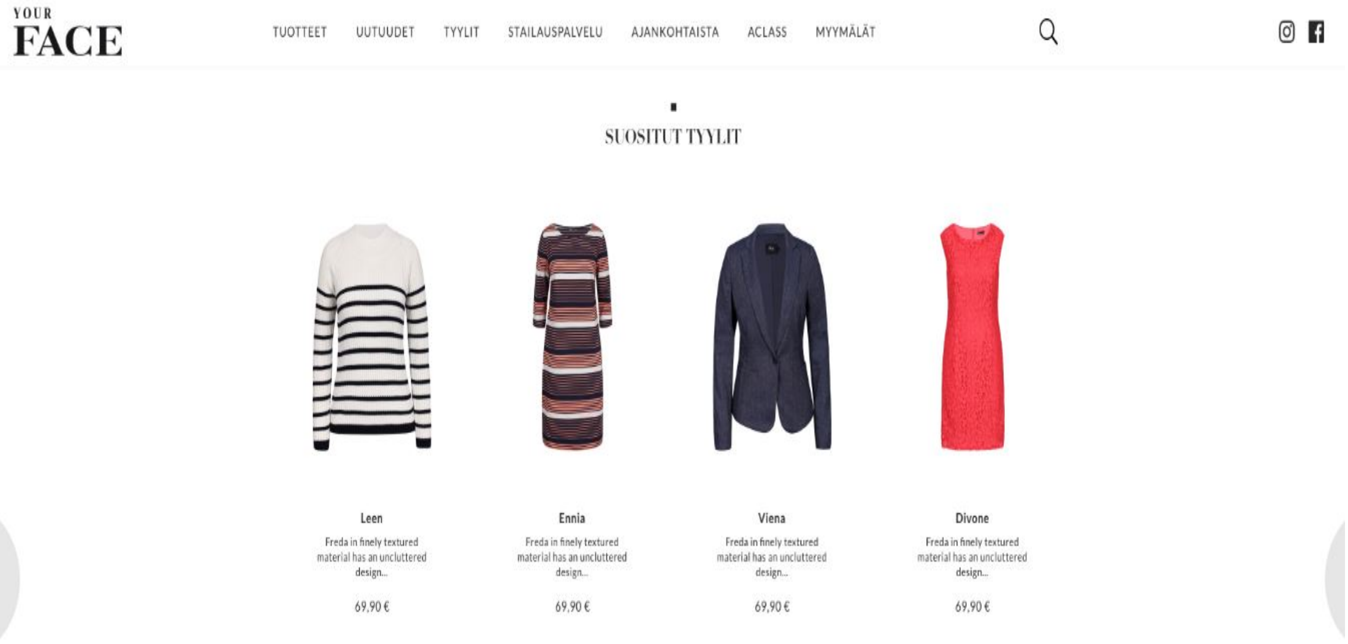
Figure 30 contains the list of files and Installing owl carousel was straight forward even though I encountered slight problem during integration with Magento [11].



```
$(document).ready(function(){
  $(".owl-carousel").owlCarousel();
});
```

FIGURE 31. Carousel JS Call [11]

After installing owl carousel files, the Carousel initializer function was called by selecting the class name as shown in figure 31. After this the Carousel function gets triggered and all the content gets wrapped in the horizontal list which provides a nice sliding feature with a responsive layout as shown in the figure 32 picture below [21].



Leen
Freda in finely textured material has an uncluttered design...
69,90 €

Ennia
Freda in finely textured material has an uncluttered design...
69,90 €

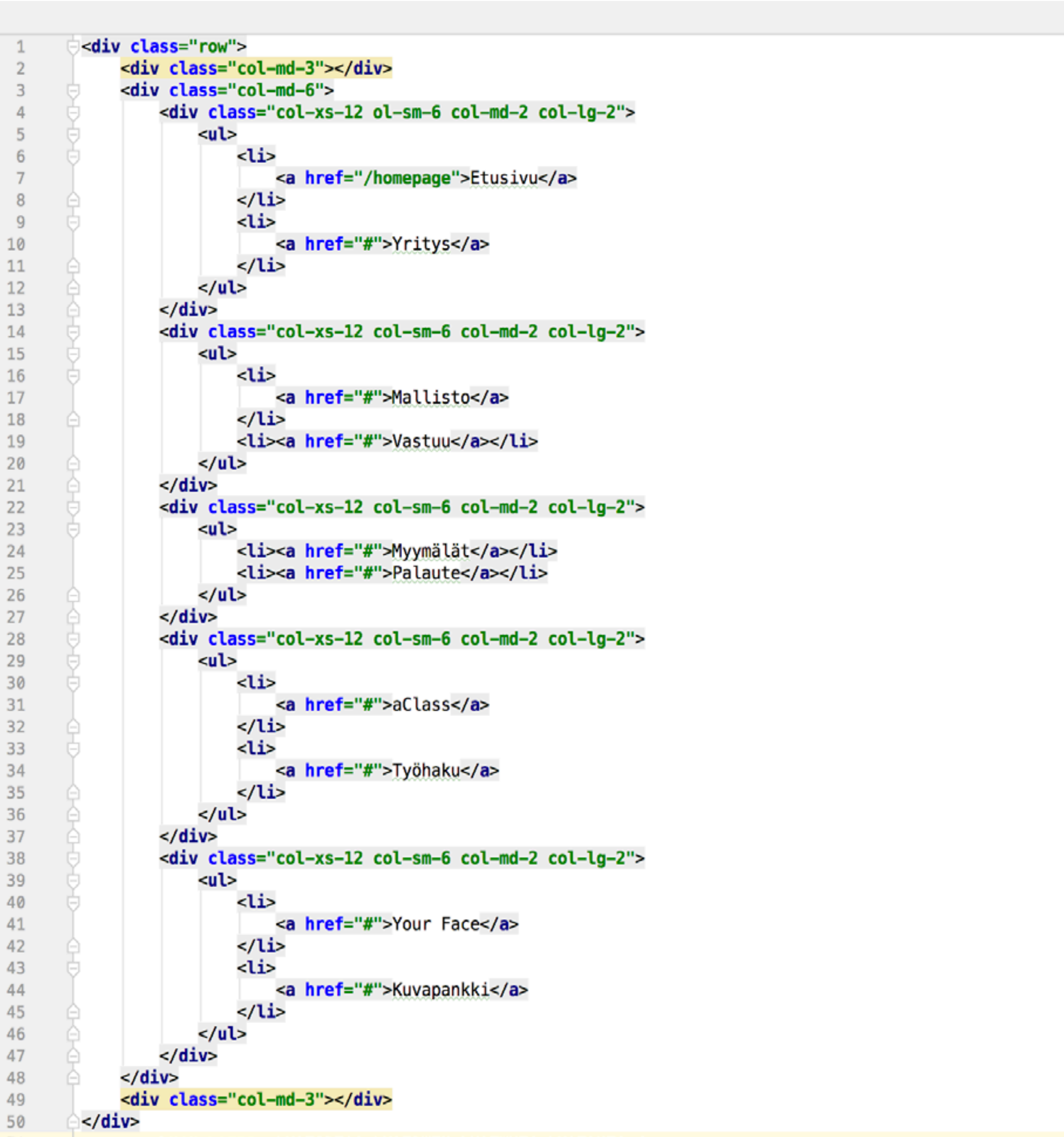
Viena
Freda in finely textured material has an uncluttered design...
69,90 €

Divone
Freda in finely textured material has an uncluttered design...
69,90 €

FIGURE 32. Selected Products Widget [21]

6.6 Footer

Footer is another essential building block for the Magento web system. A web application would be almost incomplete without the footer content. In order to implement the footer, the static block in the Magento template file was created for rendering the content. After that, the logo was placed on the top of the footer content and copy right information was placed underneath.



```
<div class="row">
    <div class="col-md-3"></div>
    <div class="col-md-6">
        <div class="col-xs-12 ol-sm-6 col-md-2 col-lg-2">
            <ul>
                <li>
                    <a href="/homepage">Etusivu</a>
                </li>
                <li>
                    <a href="#">Yritys</a>
                </li>
            </ul>
        </div>
        <div class="col-xs-12 col-sm-6 col-md-2 col-lg-2">
            <ul>
                <li>
                    <a href="#">Mallisto</a>
                </li>
                <li><a href="#">Vastuu</a></li>
            </ul>
        </div>
        <div class="col-xs-12 col-sm-6 col-md-2 col-lg-2">
            <ul>
                <li><a href="#">Myymälät</a></li>
                <li><a href="#">Palaute</a></li>
            </ul>
        </div>
        <div class="col-xs-12 col-sm-6 col-md-2 col-lg-2">
            <ul>
                <li>
                    <a href="#">aClass</a>
                </li>
                <li>
                    <a href="#">Työhaku</a>
                </li>
            </ul>
        </div>
        <div class="col-xs-12 col-sm-6 col-md-2 col-lg-2">
            <ul>
                <li>
                    <a href="#">Your Face</a>
                </li>
                <li>
                    <a href="#">Kuvapankki</a>
                </li>
            </ul>
        </div>
    </div>
    <div class="col-md-3"></div>
</div>
```

FIGURE 33. Footer content grid layout

Figure 33 is a snippet of the Bootstrap column and grid system which were utilized during the creation of the footer section. The Bootstrap column was defined from a larger screen size, such as a desktop to an extra small screen size, such as mobile devices. By using Bootstrap column grid system, it was possibile to make a responsive footer which scales between various screen sizes. Within the Bootstrap column div, the author created the list and styled it eventually to achieve the figure 34 layout [21].

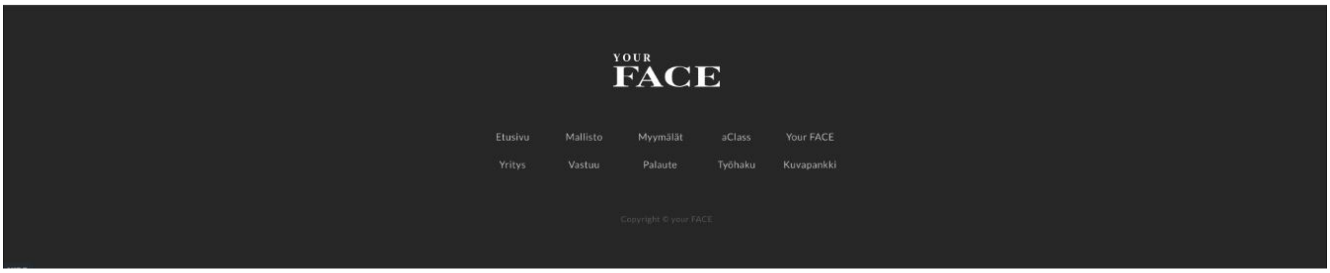
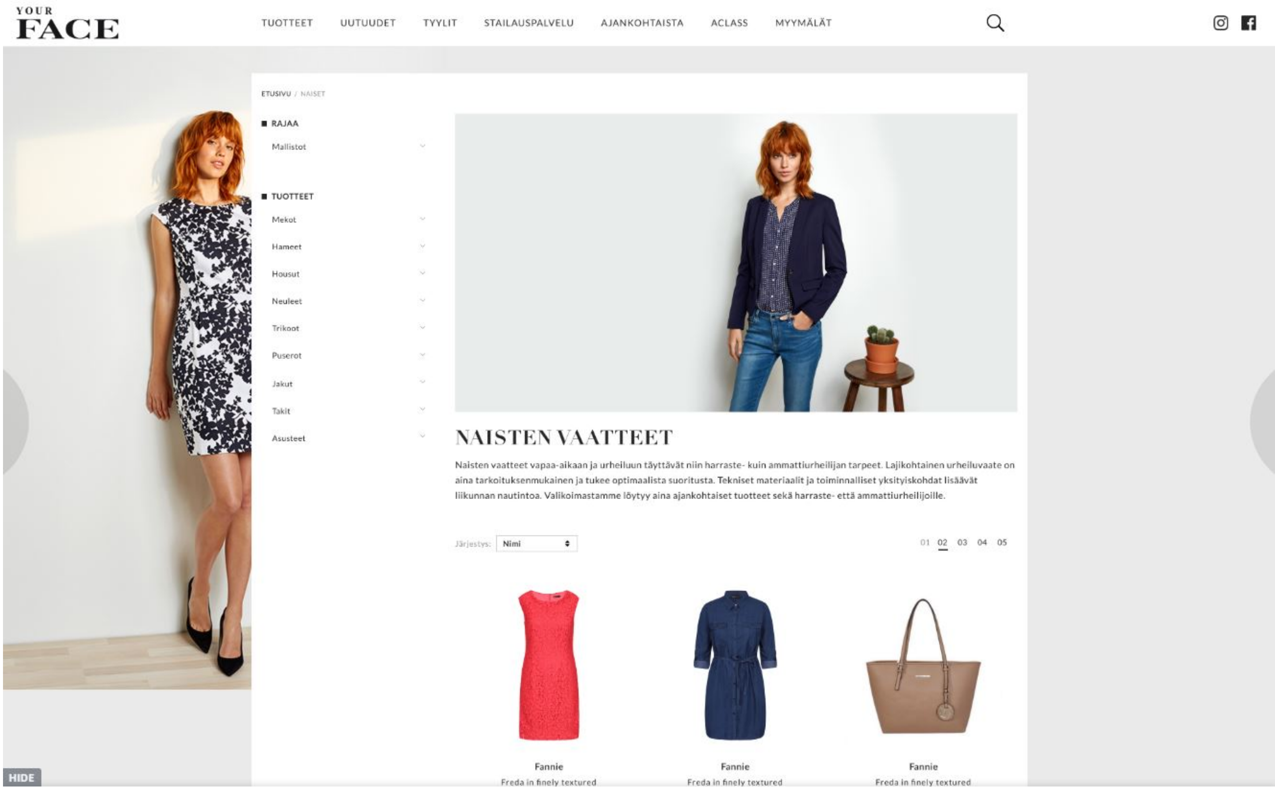


FIGURE 34. Footer section

6.7 Category

Magento provides useful ways of organizing the products and displaying them into a grid view or a list view as a collection. The Magento category creation system was used to create a 3x3 grid product collection as seen in the figure 35 below. Also, the category to be sortable by a name, a price and a location was created.

On the left-hand side, the author created category filters to limit the products based on the price and categories. [21]



NAISTEN VAATTEET

Naisten vaatteet vapaa-aikaan ja urheiluun täyttävät niin harraste- kuin ammattiurheilijan tarpeet. Lajikohtainen urheiluvaate on aina tarkoituksenmukainen ja tukee optimaalista suoritusta. Tekniset materiaalit ja toiminnalliset yksityiskohdat lisäävät liikunnan nautintoa. Valikoimastamme löytyy aina ajankohtaiset tuotteet sekä harraste- että ammattiurheilijoille.

Fannie
Freda in finely textured

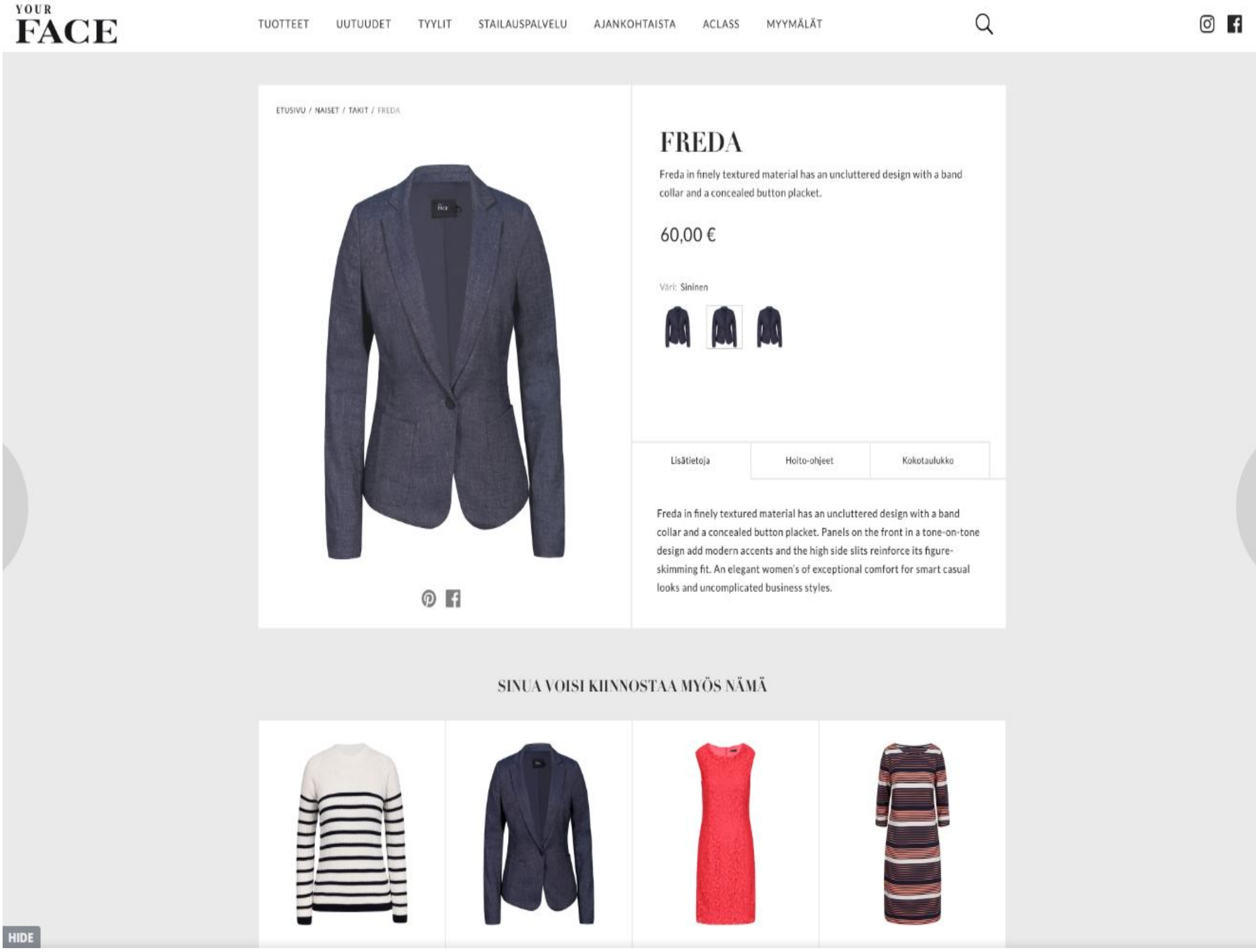
Fannie
Freda in finely textured

Fannie
Freda in finely textured

FIGURE 35. Category page [21]

6.8 Product Page

The Magento Product page contains every detail about the products. Magento provides flexibility for the developers to create the product content if needed. The Magento default template was utilized to render the custom attributes, such as more information on products, care instructions, a size chart for the product pages. On the left side, it was created the product image to be displayed and right hand side it was created the placeholder for the Product name, price, colour, and size chart. At the bottom side, the author added a couple of social icons for giving users a possibility to share the products in the social networks. Underneath this product information, 4 related horizontal product list were implemented. These horizontal product lists let the user select the similar products as a suggestion. Therefore, the final output of the result is shown in the figure 36 below [21].



FREDA

Freda in finely textured material has an uncluttered design with a band collar and a concealed button placket.

60,00 €

Väri: Sininen

Freda in finely textured material has an uncluttered design with a band collar and a concealed button placket. Panels on the front in a tone-on-tone design add modern accents and the high side slits reinforce its figure-skimming fit. An elegant women's of exceptional comfort for smart casual looks and uncomplicated business styles.

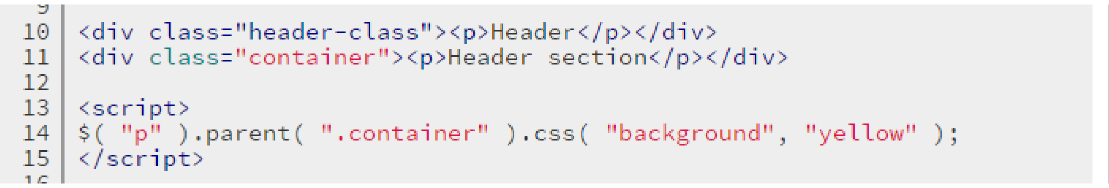
FIGURE 36. Product Page [21]

7 MOBILE VERSION

Implementing the mobile version and fixing the styles for a broken layout, which did not match when switching back from the desktop part, became extremely tricky. One mistake that the author realised lately was that the mobile design should have been implemented first as many styles could have been easily extended to the desktop version by up scaling responsibly.

After struggling for a few hours, the author started to implement the Mobile floating header, its content and footer section effectively. Understanding the PHP function and having a sound knowledge of the JavaScript and CSS selector were very important to overcome the tricky problematic situation.

One tricky situation thatwasI encountered frequently was to select the parent class style. After doing research for some amount of time, it was realised that it is still not possible to select the parent class or id or any element through CSS. There was a long debate going on even starting from the ancient version of CSS, but the functionality was still not implemented until the current version of CSS3. Therefore, in order to overcome these limitations, starting from the jQuery version 1.0 they introduced these features and hence made the work easier. Any html element could be easily targeted using jQuery Parent selector.



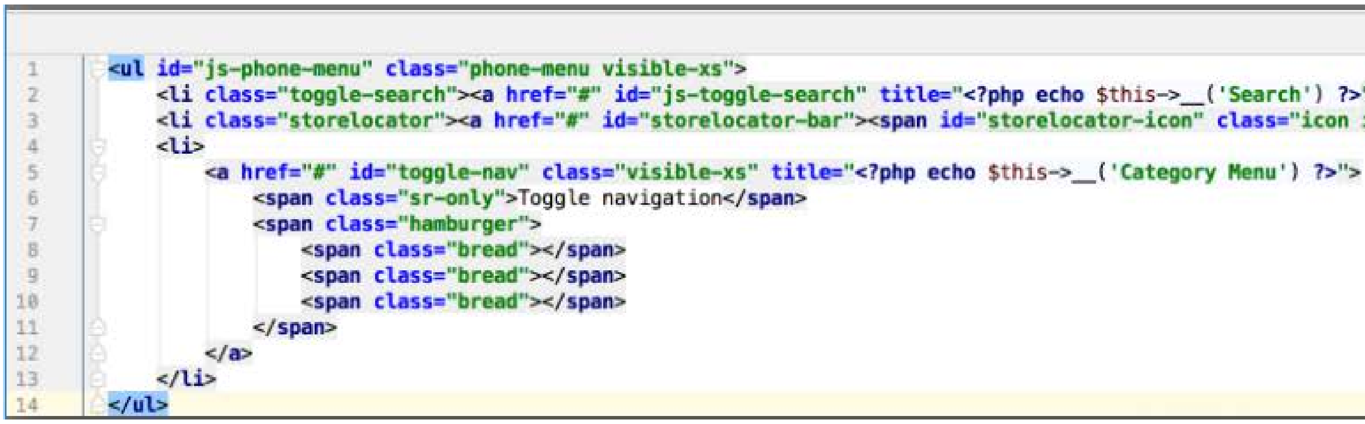
```
<div class="header-class"><p>Header</p></div>
<div class="container"><p>Header section</p></div>

<script>
$( "p" ).parent( ".container" ).css( "background", "yellow" );
</script>
```

FIGURE 37. jQuery parent selector

Figure 37 contains a piece of the jQuery snippet which is being selected for the “p” html element. Using the jQuery, the parent class “container” is selected and based on that selection CSS is applied. Therefore, this strategy helped me to do complex and tricky selection in a very short span of time.

7.1 Floating Header and Menu

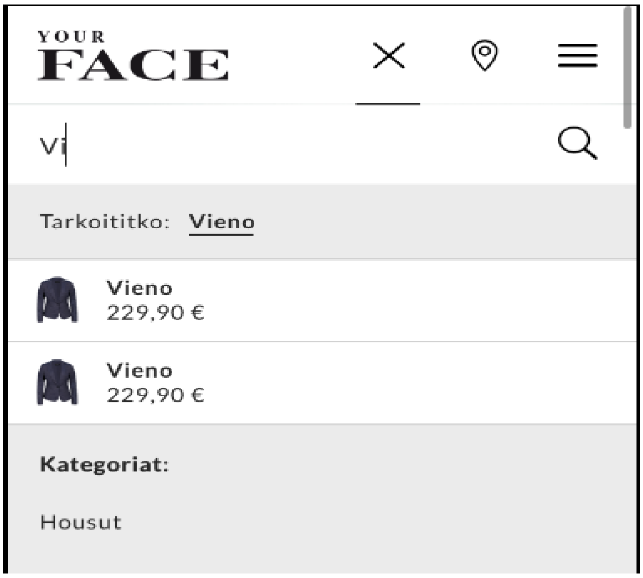
Having the floating header in a mobile was one of the key requirements from the customer and project aspects. Considering this requirement, the Floating header in the mobile version was implemented.



```
<ul id="js-phone-menu" class="phone-menu visible-xs">
    <li class="toggle-search"><a href="#" id="js-toggle-search" title="<?php echo $this->__('Search') ?>'
    <li class="storelocator"><a href="#" id="storelocator-bar"><span id="storelocator-icon" class="icon :
    <li>
        <a href="#" id="toggle-nav" class="visible-xs" title="<?php echo $this->__('Category Menu') ?>">
            <span class="sr-only">Toggle navigation</span>
            <span class="hamburger">
                <span class="bread"></span>
                <span class="bread"></span>
                <span class="bread"></span>
            </span>
        </a>
    </li>
</ul>
```

FIGURE 38. Floating header Code

As seen in the figure 38, a certain class for each HTML required element was created and also the PHP code to render the translation file was embedded. Therefore, each class and id could be targeted to style through CSS. Figure 39 below consists of the final output after the implementation. [21]



YOUR FACE

Vi

Tarkoititko: Vieno

Vieno
229,90 €

Vieno
229,90 €

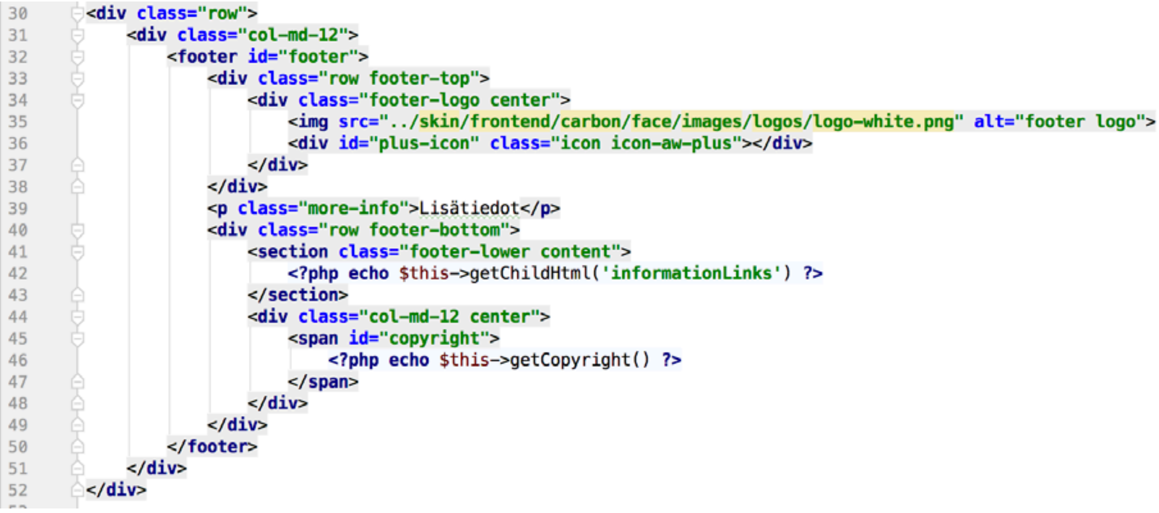
Kategoriat:

Housut

FIGURE 39. Mobile floating header and menu after implementation [21]

7.2 Footer

Due to the limitation of a mobile view port and giving a smooth UI, the footer section was also required to be implemented differently for the mobile version.



```
<div class="row">
    <div class="col-md-12">
        <footer id="footer">
            <div class="row footer-top">
                <div class="footer-logo center">
                    <img src="../skin/frontend/carbon/face/images/logos/logo-white.png" alt="footer logo">
                    <div id="plus-icon" class="icon icon-aw-plus"></div>
                </div>
            </div>
            <p class="more-info">Lisätiedot</p>
            <div class="row footer-bottom">
                <section class="footer-lower content">
                    <?php echo $this->getChildHtml('informationLinks') ?>
                </section>
                <div class="col-md-12 center">
                    <span id="copyright">
                        <?php echo $this->getCopyright() ?>
                    </span>
                </div>
            </div>
        </footer>
    </div>
</div>
```

FIGURE 40. Mobile footer section code

As shown in the figure 40, the footer section was initiated with the class row, which is then followed by the Bootstrap class “col-md-12” to take the full column width in the screen below 1200pixels media. All the footer DOM element was styled by targeting the specified class and id and the final output could be achieved as shown in the figure 41 below. [21]

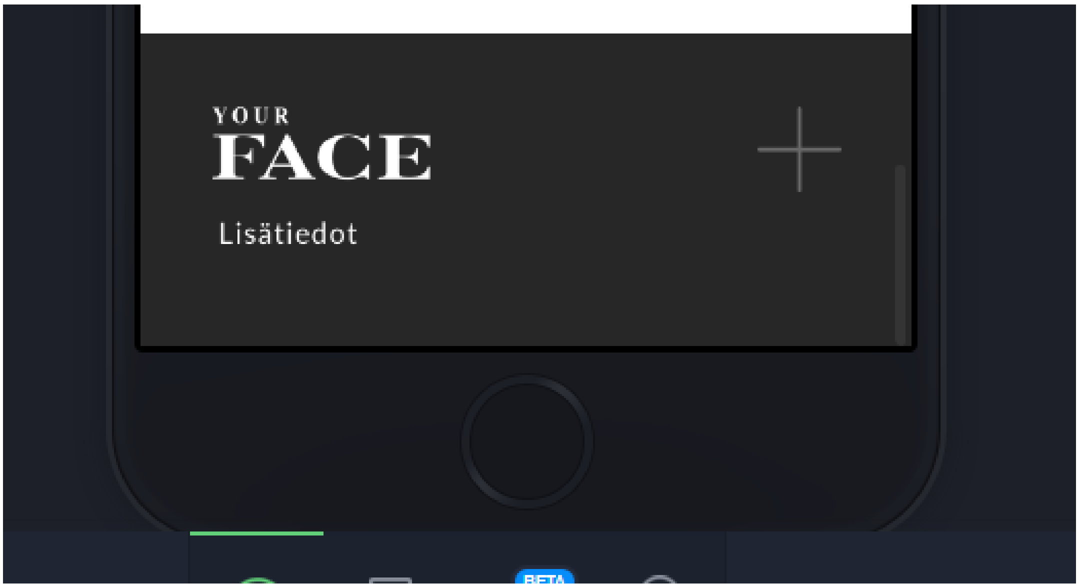
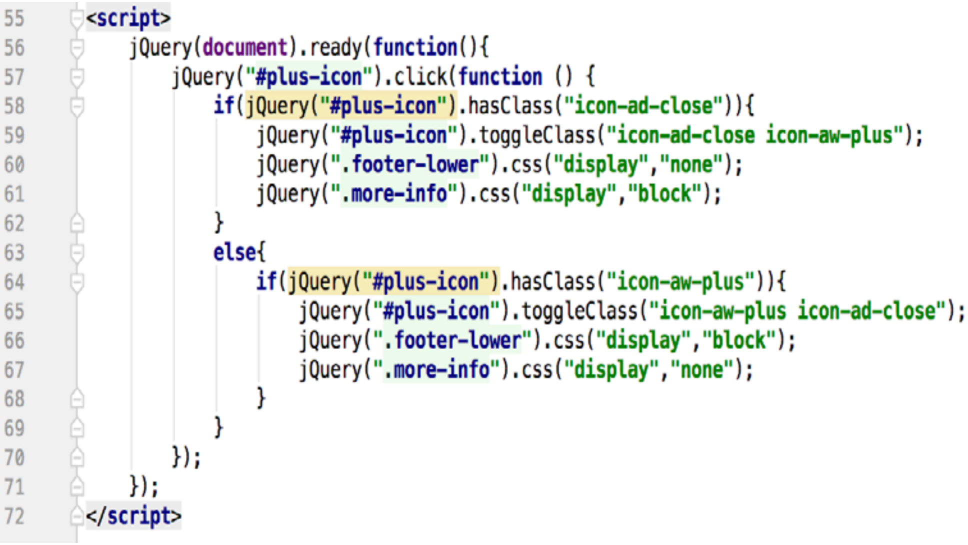


FIGURE 41. Mobile footer section without menu [21]

Having this much output was not sufficient. Therefore, based on the few aforementioned classes and id within the DOM element the author implemented the certain functionality using jQuery so that when the user clicks on the plus button, all the content would be displayed and when clicked back it would remove those contents.



```
<script>
    jQuery(document).ready(function(){
        jQuery("#plus-icon").click(function () {
            if(jQuery("#plus-icon").hasClass("icon-ad-close")){
                jQuery("#plus-icon").toggleClass("icon-ad-close icon-aw-plus");
                jQuery(".footer-lower").css("display","none");
                jQuery(".more-info").css("display","block");
            }
            else{
                if(jQuery("#plus-icon").hasClass("icon-aw-plus")){
                    jQuery("#plus-icon").toggleClass("icon-aw-plus icon-ad-close");
                    jQuery(".footer-lower").css("display","block");
                    jQuery(".more-info").css("display","none");
                }
            }
        });
    });
</script>
```

FIGURE 42. jQuery functionalities

As can be seen in the figure 42, the script tag was opened and the jQuery document ready function was called. When the document was ready, the plus icon was selected, and a few condition were passed. When the condition was met, few functionality, such as toggling the class and adding some CSS properties to the HTML DOM element were specified. Therefore, the final output of the result is shown in figure 43 below.

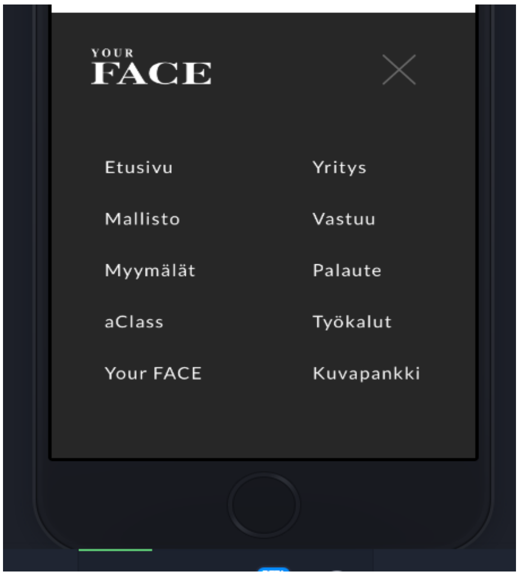
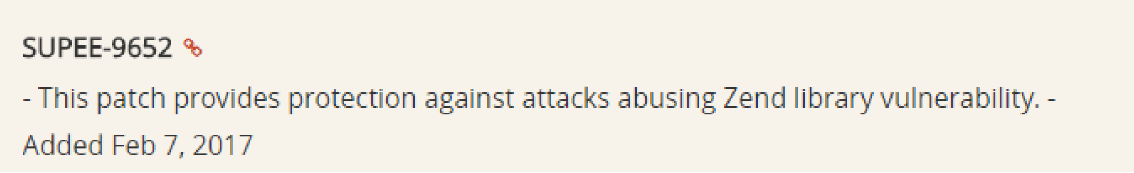


FIGURE 43. Mobile footer section with expanded mobile

8 WEB APPLICATION SECURITY, ITS PATCHES AND FIXES

Magento has released various patches and fixes for its different products, Such as Magento Enterprise as well as Community edition differently. These patches consist of various fixes for fixing the bug and security loop holes. Due to the security concern and to make the project free from vulnerability, the author applied the latest patches which included various fixes to the project.



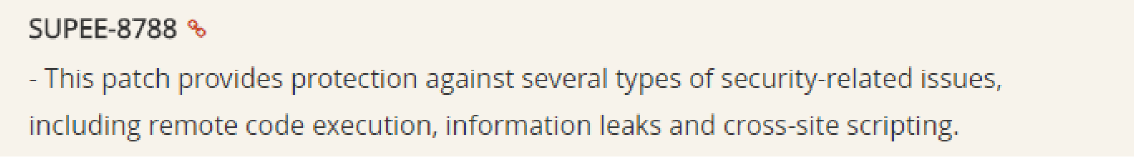
SUPEE-9652

- This patch provides protection against attacks abusing Zend library vulnerability. - Added Feb 7, 2017

FIGURE 44. SUPEE-9652 Patch support from Magento [14]

For example: Figure 44 shows SUPEE-9652 critical patch which was used as soon as it was released on Feb 7, 2017 to apply the fixes for the Zend Framework 1 vulnerability [14].

Similarly, The SUPEE-8788 patch (shown in figure 45) was applied to this project. By applying this patch, it ensured from security related issues, such as a remote code execution, information leaks and cross site scripting. [13]



SUPEE-8788

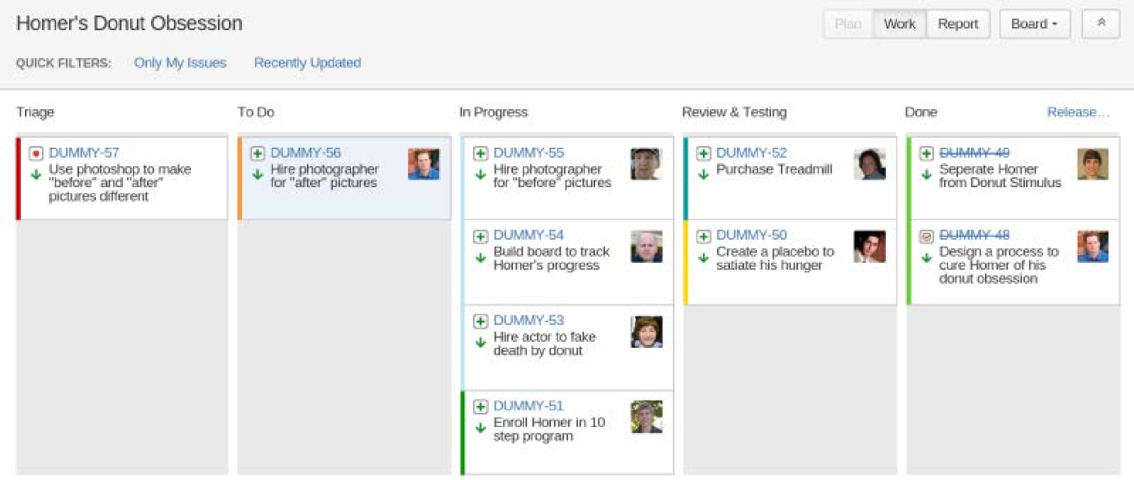
- This patch provides protection against several types of security-related issues, including remote code execution, information leaks and cross-site scripting.

FIGURE 45. SUPEE-8788 Patch support from Magento [13]

Therefore, applying the patch was one of the vital parts to make the project secure and risk free from many known security vulnerabilities.

9 QUALITY ASSURANCE AND TESTING

Quality Assurance is a vital part of the project management system and workflow in VAIMO. Every completed task goes through a strict code review session and a Quality Assurance and testing phase. The author's were divided into several parts. Before the tasks were started, they were moved to the 'In Progress' column then After completing of one task, the author was required to move that task into the 'Review' and 'Testing phase' from the Jira Kanban board. When the Quality Assurance and Testing was done and the task was accepted, it was moved to the 'Done' column. This also means that the task is now ready to be deployed to the production environment. When the task has been moved to the Done column in the Jira Kanban Board as shown in the example figure 46 and received the signal for Go Live, then the changes are pushed through the help of Command Line Tools (CLI) and Mercurial to the version control system and the tasks is released to the production environment. [19]



Triage: DUMMY-57 Use photoshop to make "before" and "after" pictures different

To Do: DUMMY-56 Hire photographer for "after" pictures

In Progress: DUMMY-55 Hire photographer for "before" pictures; DUMMY-54 Build board to track Homer's progress; DUMMY-53 Hire actor to fake death by donut; DUMMY-51 Enroll Homer in 10 step program

Review & Testing: DUMMY-52 Purchase Treadmill; DUMMY-50 Create a placebo to satiate his hunger

Done: ~~DUMMY-49~~ Seperate Homer from Donut Stimulus; ~~DUMMY-48~~ Design a process to cure Homer of his donut obsession

FIGURE 46. Jira Kanban Board Example [19]

10 CONCLUSION

YourFace is a Finnish high level fashion service chain clothing store for women. It is also part of the Finnish Luhta Sportswear company group, which is one of the Nordic region’s largest apparel companies. Therefore, delivering the project on time was extremely vital to me and the company. [21]

On April 2017, the YourFace Project was completed and delivered to the customer for a further review and testing before making it available to the public. With various efforts, dedication and hard work, I was able to deliver one more successful project to the customer before the deadline. Looking at the happy and satisfied customers, I was also highly motivated to move forward with a greater satisfaction.

During the development phase, numerous bugs were encountered. Surprisingly, I am happy to admit that those obstacles were fixed. Obviously, some problems were very complicated and challenging, but with the help of other developers I was able to overcome the situation easily.

By working with the YourFace project, I gained plenty of experience and I am highly satisfied with my improved competence. Even though I have worked with many projects, YourFace gave me very much in-depth knowledge and further improved my Magento e-commerce webstore development skills. I was also required to do a lot of research and self-study during the development phase in order to work with the project efficiently.

REFERENCES

1. Magento 2017. Magento Community Edition. Date of retrieval 08.03.2017

   https://magento.com/products/community-edition

2. Mercurial-SCM. 2005. Mercurial(Version Control Tool). Date of retrieval 09.03.2017

   https://www.mercurial-scm.org/wiki/Mercurial

3. Wikipedia 2017. Wikipedia(web development tools). Date of retrieval 10.03.2017

   https://en.wikipedia.org/wiki/Web_development_tools

4. Sass-Lang 2016. SASS(documentation and reference). Date of retrieval 12.03.2017

   http://sass-lang.com/documentation/file.SASS_REFERENCE.html

5. Bootstrap 2010. Bootstrap(HTML, CSS, JS as a front end framework). Date of retrieval 13.03.2017

   http://getbootstrap.com/css/

6. Wikipedia 2017. PHP. Date of retrieval 14.03.2017

   https://en.wikipedia.org/wiki/PHP

7. PHP 2005. PHP manual. Date of retrieval 15.03.2017

   http://php.net/manual/en/intro-whatis.php

8. GitHub 2017. Introducing Git Flow. Date of retrieval 16.03.2017

   https://datasift.github.io/gitflow/IntroducingGitFlow.html

9. University of Wisconsin 2012. Version Control. Date of retrieval 17.03.2017

http://pages.cs.wisc.edu/~driscoll/software/vcs/

10. Tutorials point 2017. Magento-Architecture. Date of retrieval 18.03.2017

https://www.tutorialspoint.com/magento/magento_architecture.htm

11. OwlCarousel2 2014. Owl Carousel 2. Date of retrieval 19.03.2017

https://owlcarousel2.github.io/OwlCarousel2/

12. Oasis Digital 2015. Jira Case Study Nested Workflows. Date of retrieval 20.03.2017

https://blog.oasisdigital.com/2015/jira-case-study-nested-workflows/

13. Magento 2017. SUPEE-8788 Patch. Date of retrieval 22.03.2017

https://magento.com/tech-resources/download

14. Magento 2017. SUPEE-9652 patch. Date of retrieval 23.03.2017

https://magento.com/security/patches/supee-9652

15. Paul Molluzzo 2017. Pesticide an open source extension for google chrome and safari. Date of retrieval 25.03.2017

https://paul.molluzzo.com/

16. Mathijs Kadijk 2016. XDebug Helper for Google Chrome. Date of retrieval 27.03.2017

https://github.com/mac-cain13/xdebug-helper-for-chrome

17. Screen sizes. Smartphones and tablets and monitors screen sizes. Date of retrieval 05.04.2017

http://screensiz.es/phone

18. Google Analytics 2017. Google Analytics and web report. Date of retrieval 10.04.2017

https://analytics.google.com/analytics

19. Atlassian 2017. The #1 software development tool used by agile teams. Date of retrieval 15.04.2017

https://www.atlassian.com/software/jira

20. Vaimo 2017. Vaimo Finland Oy. Date of retrieval 20.04.2017

http://www.vaimo.com/about-us/our-story

21. YourFace 2017. YourFace brand store. Date of retrieval 22.04.2017

https://yourface.fi

22. Wikipedia 2017. Application Programming Interfaces. Date of retrieval 23.04.2017

https://en.wikipedia.org/wiki/Application_programming_interface