

## **Building a WordPress site for Sports Academy of Kouvola**

Aaro Helander

15.5.2015



<b>Author(s)</b> Aaro Helander	
<b>Title of report</b> Building a WordPress site for Sports Academy of Kouvola	
<b>Supervisors</b> Tero Tuoriniemi	<b>Number of pages and appendix pages</b> 31 + 8
<p>The objective of this study was to create two WordPress based websites for the Sports Academy of Kouvola. The Main functions of the Sports Academy website are to enable efficient marketing for upcoming events and to act as a billboard for company news and other public announcements. The scope included creating the basic layout and functionality for the site, and did not include creating the actual content for the site – except for the essential information transferred from the old site.</p> <p>The site was drafted using Twitter Bootstrap 3 as a front-end framework. A template layout was created which was then converted to a WordPress theme. All essential functions – such as the landing page, blog page and other relevant pages were created, and by the time the project was considered complete, the existing design was re-created using a theme called Divi 2 from Elegant Themes – a company creating themes and plugins for WordPress.</p> <p>The thesis was started in late 2014, and the site was completed in March 2015, but due to lack of resources and available content the site was published in April 2015.</p> <p>The learning objectives were to understand how WordPress works by creating a theme from scratch, and then evaluating the pros and cons of creating a site from scratch versus using a commercial theme.</p> <p>This thesis is aimed at people with a basic understanding of terminology and conventions related web design.</p>	
<b>Keywords</b> WordPress, Web site development, Bootstrap, Elegant Themes, Divi	

## Table of Contents

1	Introduction .....	3
1.1	Out of scope .....	3
1.2	Concepts and terms .....	4
2	Consulting for freelancers .....	4
2.1	How much does WordPress development cost? .....	4
3	Front-end Frameworks .....	5
3.1	Which framework to choose? .....	6
4	Introduction to WordPress .....	7
4.1	The Codex .....	8
4.2	Hosting and installation .....	8
4.3	WordPress user roles.....	10
4.4	WordPress Themes .....	10
4.5	WordPress Template Hierarchy .....	11
4.6	Child Themes.....	13
4.7	WordPress Theme Frameworks.....	13
4.7.1	Deciding between Genesis Framework and Elegant Themes .	14
4.8	The WordPress Loop .....	14
4.9	WordPress Plugins .....	15
4.10	Plugin dependencies and security.....	16
5	WordPress Theme Security.....	18
5.1	What is a safe WordPress theme? .....	18
5.2	Clean code.....	18
5.3	Plugin safety .....	19
5.4	Security breaches .....	19
6	Converting a Bootstrap 3 Template to a WordPress Theme.....	20
6.1	An overview of the initial site structure .....	24
6.2	Used Template Files (On our initial Bootstrap based theme).....	25
7	Changing the Theme.....	26
7.1	Divi 2 by Elegant Themes .....	27
7.2	Pros of using a premium theme .....	28
7.3	Plugins I used with Divi 2 .....	29
7.3.1	Divi Booster.....	29
7.3.2	Divi Module Editor.....	29

7.3.3 Contact Form 7 .....	29
7.3.4 WordFence .....	29
8 Summary of results .....	30
9 Conclusions .....	30
10 Appendix.....	32
10.1 Overview of the site .....	32
11 References.....	38

# 1 Introduction

This thesis is about creating a modern website suitable for marketing for a small company, using the popular Content Management System WordPress. The case company is the Sports Academy of Kouvola.

Initially, the Bootstrap 3 front-end framework was used to prototype the layout for the website, which was then converted to a WordPress Theme that can be set as the website theme from the WordPress administration panel

The project has several main goals. The first one is to make a modern website to support the expansion of the company to provide more value to the public using their services. The website will contain a blog section featuring posts and articles from the academy coaches and other partners regarding physical and mental training, nutrition, health, and general information about the company events.

The second goal is to provide marketing for various projects and events the Sports Academy has been working on. One of the most recent ones is a physical preparation training program for young athletes called "Yleistaito-olmennus". I'm one of the co-founders of this project and the main coach, so this Thesis has personal value to me as well..

All of the photos used on the site for Yleistaito-olmennus are taken by me. I used a course provided Team Treehouse called "From Bootstrap to WordPress" as the guide when I first began creating the theme for these sites. The course material is not available to public, but only to subscribe members.

## 1.1 Out of scope

Out of scope of this thesis are the content production, meaning blog posts and articles, except for the ones with high business value at the moment, such as registration forms for an upcoming summer camp and the "yleistaito-olmennus" training sessions. Also, conclusive search engine optimization is out of scope, since at the moment it does not provide any business value to the company.

It should be made clear that the approaches and ideas presented in this thesis are not the only way to approach this kind of project. As of 2015, WordPress development has become such a huge business that it is out of the scope of this thesis to thoroughly discuss even a fraction of the conversation around the issue of theme development. Also, many of

the referenced sources in this thesis are web development related sites, blogs and articles, since I believe that for a modern freelance developer, these medias are the most current ones to use when deciding what is the best approach to do something.

## **1.2 Concepts and terms**

- CMS (Content Management System) – An application used to create, publish, edit and manage content for a website using a central interface requiring little to no technical understanding of how the website works.
- WordPress – A popular, free CMS known for its ease of use and potential to be used with simple blog sites to complex web stores.
- Bootstrap, a frontend framework – A collection of HTML- and CSS-based design templates and interface elements designed to ease web development and fast prototyping of web applications.
- Responsive design – A site design that fluidly scales properly to fit multiple screen sizes, providing an optimized viewing experience for both desktop and mobile users.
- WordPress plugins – Modular small applications working within the WordPress installation, used to carry out different types of functionality, like contact forms, security add-ons, parts of the site layout etc.

Rest of the concepts are explained within the chapters in which they are presented

## **2 Consulting for freelancers**

The main function of customer consulting is to discover customer needs, and in the absence of clearly defined needs, the consultant will have a chance to create needs for the customer. In my experience, clients usually do not exactly know what they want, and by providing them more value by introducing new concepts allows for win-win deals to be made, where both the client and the developer gain profit. (Entrepreneur, 2014)

### **2.1 How much does WordPress development cost?**

People are interested in WordPress because it is popular, open source and free. Free, right? Although WordPress is free, the answer to this question is not as simple as people think. In a fairly recent article on Codeable titled “How much does a WordPress site really

cost?” the author Tomaž Zaman describes some typical scenarios of consulting a client about building their WordPress site.

According to the article, the usual scenario goes like this:

1. The client has no proper copywriting or artwork for their site, and decides to search out a theme provided by some WordPress theme company, because they're cheap. There are lots of great looking themes available for low prices.
2. The client and consultant install WordPress and set up the theme, and then struggle for weeks to come up with enough copy and other material to fill out all the placeholders in the theme.
3. After weeks of struggling, the client accepts the fact that their website will never look as good as the theme demo, since they just don't have that much to say.

The article states that clients are often amazed that the modification of the theme they bought will cost ten to hundred times the price they paid for the theme. Amounts up to tens of thousands of euros are not that uncommon. Building a complete theme from scratch can be valued even higher in many cases. (Codeable 2015 a).

In this case I started by creating the theme from scratch, and actually encountered several of the problems mentioned again and again. In my experience, lack of proper content made the design process slower. A lot of time was also spent trying out various approaches to design, choosing the right plugins, and trying to decide whether a function was best performed by a certain plugin, or if making it part of the core WordPress functionality was the right choice.

### **3 Front-end Frameworks**

Creating every design element from scratch can be a time-consuming process. A framework with a pre-defined grid system and design elements would save time, especially when the purpose was to quickly prototype web applications. The framework I used to draft both of the sites is Twitter Bootstrap 3.

A straight quotation from the Bootstrap project Github page describes the project like this:

*“Bootstrap is a sleek, intuitive, and powerful front-end framework for faster and easier web development, created by Mark Otto and Jacob Thornton, and maintained by the core team with the massive support and involvement of the community.”* (Github 2015 a)

Bootstrap quickly became one of the most widely used frameworks for prototyping websites, providing pre-defined grid for creating a consistent site layout, and pre-styled elements such as buttons, forms and other useful things to be used in within the site. The second version of Bootstrap was equipped with responsive capabilities out of the box, which made developing mobile-compatible websites a bit easier again.

Bootstrap 3.0 featured a mobile first design approach, meaning that all of its features were first handedly designed mobile use in mind. The “mobile first” approach is considered a modern guideline for web design, since it is often easier to add more things to a design compared to stripping down styling and possibly having to delete content that does not look good on mobile displays. (Sitepoint 2014).

At the moment there are two popular responsive front-end frameworks available: Twitter Bootstrap and Zurb Foundation. The Github page for Foundation defines it as follows:

*“Foundation is the most advanced responsive front-end framework in the world. You can quickly prototype and build sites or apps that work on any kind of device with Foundation, which includes layout constructs (like a fully responsive grid), elements and best practices.”* (Github 2015 b)

As of 2015 there exist many powerful frameworks, Bootstrap and Foundation being only two of the most widely used.

### **3.1 Which framework to choose?**

First of all it is good to note that using a framework is by no means required in order to develop a site, and it is up to personal preference to use or not to use one. There are good arguments of both sides of the conversation, but going deeper into comparing the pros and cons of using a framework is out of scope of this thesis.

Essentially both frameworks are suitable for most types of web development – both frameworks implementing the mobile-first approach to their design. Bootstrap 3 is commonly considered as the framework of choice for quickly prototyping of web applications, since it provides more user interface elements and widgets, and has a more pronounced style with its design.



In this project I decided to use Bootstrap 3, but it's up to the developer which one he or she wants to use. Both frameworks are decently easy to use after the initial learning curve is dealt with.

As a conclusion, using a framework will speed up the implementation of a business site to a large degree and allow both the developer and the customer to have a design approach more focused on the content and its looks rather than designing just features for the sake of having cool features on the site. Since the frameworks are free to use and easily customizable. For developers who are not focused on creating completely customized designer-sites, there is no reason not to use a framework. (Pettit, 2014)

## **4 Introduction to WordPress**

A Content Management System (CMS) is software designed to make authoring web pages as easy and efficient as possible (Wikipedia 2015 a). WordPress, released on 2003, was originally designed to be a fairly simple blogging platform, but has evolved to a fully equipped CMS. There are two "versions" of WordPress - the Wordpress.com which is a free service where anyone can create an account for free and create a blog.

WordPress.org provides the WordPress software as a download for self-hosting the CMS, providing more control and customizability over the application. Since WordPress is free software licensed under the GPL v2 software license, it can be modified by any means the developer sees fit. (WordPress 2015 a).

WordPress is essentially a PHP-application that uses PHP to dynamically insert content to what is called a Theme. Themes in WordPress are exchangeable visual layouts which are separate from the content of the website, making it easy to switch the look of a site.

WordPress is an open source project under the General Public License (GPL), meaning that as other free software, WordPress is free to use, redistribute, reinvent, expand etc (Onishi 2013 a, 1-2.).

One of the reasons for the popularity of WordPress is its huge amount of available plugins, and its excellent Search Engine Optimization (SEO) capabilities. WordPress is commonly seen as a platform that is very intuitive to use, which is one of the reasons I decided to use it as the platform of choice for this company.

Site layout is controlled by creating a theme which can be activated from the WordPress administration panel without affecting the content of the site. Updates for all plugins and WordPress itself are easily managed from the administration panel.

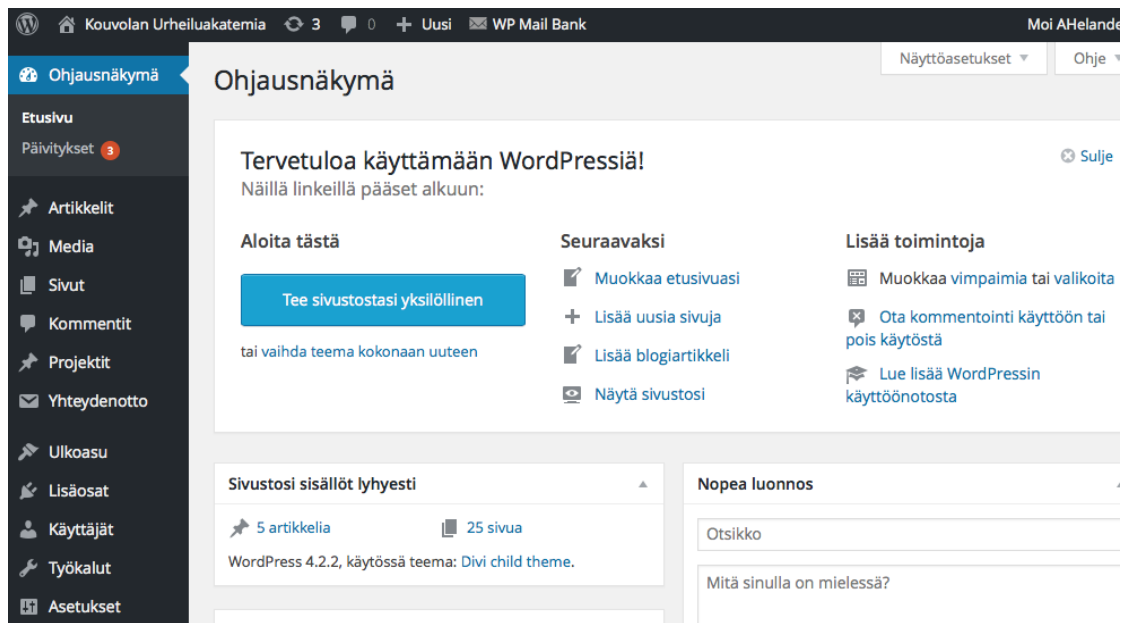


Figure 1 - The WordPress Admin Panel

## 4.1 The Codex

WordPress has a vast documentary called The WordPress Codex, which is an encyclopedia containing explanations for all elements related to WordPress. The codex site can be used as a tutorial for learning WordPress, although in my experience it serves better as a reference material when specific instructions about some component or used function is required. (Onishi 2013 b, 1-2)

## 4.2 Hosting and installation

WordPress based sites can be hosted on several different providers, and the simplest hosting configuration consists of a server running LAMP (Linux, Apache, MySql and PHP).

Installation requirements of WordPress:

- A web server with LAMP or LEMP (Linux, Nginx, MySql, PHP)
- a text editor such as Notepad, Brackets etc.
- A FTP client

Detailed installation instructions can be found in the WordPress Codex and various other WordPress related communities. Many hosting providers also provide related support regarding WP configurations. (WordPress 2015 b).

Many hosting providers and VPS (Virtual Private Server) providers offer a one-click WordPress installation, which configures the server with a LAMP (Linux, Apache, MySQL & PHP) or LEMP (Linux, Nginx, MySQL & PHP) stack. Both server configurations are viable options for a WordPress installation with LAMP still being the most commonly used configuration (WordPress 2015 c).

Some important factors when selecting a hosting provider:

1. Performance
2. Fast customer support and knowledge resources
3. Pricing and accessory services offered

Depending on the current and future requirements of your site, there are many more important factors to be considered when selecting a hosting provider. The three mentioned above are what I personally look for when working on small scale projects like this.

If the website will supposedly have a huge amount of daily visitors, these issues become very important. In our case, the Sports Academy already had their old site hosted by WebHostingPad with an active membership until August 2015. At this point we decided that selecting the perfect hosting for future was not essential, so I set up the new site temporarily to my own VPS (Virtual Private Server) provided by Digital Ocean.

Sticking to my own server gave me good control over the site when I was trying different approaches and learning about server configurations.

### 4.3 WordPress user roles

Many CMS's, like WordPress, include different user roles with different privileges, allowing people with different responsibilities to handle specific tasks, increasing security for the site and making content administration and quality control easy. The user roles of WordPress (From the WordPress Codex):

- [Super Admin](#) – somebody with access to the site network administration features and all other features.
- [Administrator](#) – somebody who has access to all the administration features within a single site.
- [Editor](#) – somebody who can publish and manage posts including the posts of other users.
- [Author](#) – somebody who can publish and manage their own posts.
- [Contributor](#) – somebody who can write and manage their own posts but cannot publish them.
- [Subscriber](#) – somebody who can only manage their profile.

(WordPress 2015 d)

Choosing a proper role for a specific individual contributing to the site is both a security concern and an usability concern. Someone who only writes blog posts does not have to, and usually doesn't even want to have access to any administrative settings in the administration panel.

### 4.4 WordPress Themes

WordPress uses a Theme to define the layout and appearance of the website. There are many free and paid premium Themes available for WordPress, provided by various companies and individual web developers. In our case we began the development by downloading a basic Bootstrap 3 template and then converting it to a WordPress theme with basic blogging functionalities.

Creating a WordPress theme from scratch is not always the best choice for every project, but doing so will provide a developer not familiar with WordPress a great deal of understanding of how the software works, and this knowledge will be useful when working with any kind of theme in the future. The most usual case for WordPress development is to choose a premium theme with the customer, and then modify that theme to match the customer's preferences. (Codeable, 2015)

I began this project by developing the theme from scratch using Bootstrap 3 to quickly prototype an initial layout for the site. I first designed the site and then converted it into a WordPress theme. After running into ElegantThemes – a WordPress theme company - and buying their developer membership I decided to trash the design I had made, and re-created it using their theme called Divi 2. Reasons for this were mostly personal, and just the plugins included in the membership were reason good enough for me to make the switch from my own hand-made theme. In the last month I've built already five sites using the Divi theme as the framework, which shows its huge business value for a developer like me.

#### 4.5 WordPress Template Hierarchy

Any WordPress theme is built on a couple of core template files which together make a functioning theme. Most themes are comprised of a lot more template files, which all have a specific purpose and function which are described in detail in the WordPress Codex.

Only two files (**index.php** and **style.css**) are required to make a technically valid WordPress Theme. For a very basic theme with some practical functionality, some more files are required. The files are:



Figure 2 - Required Template Files for a Very Basic Theme

The **index.php** is the default template file for the home page of your site. In other words, it's the first file that WordPress looks for to display your site. In the WordPress administrator view, it is possible to select either a static front page or the blog section to be displayed as the front page.

The **header.php** and **footer.php** template files are used to include the code for elements visible on all pages, such as the site navigation in the header, and company info and external links in the footer. This is very handy compared to having to copy the same code over and over again to every page of the site.

The **functions.php** file is a file containing optional functions to customize WordPress. For example, functions to add custom navigation properties, creating widgets and so on.

The **style.css** is the basic css-stylesheet for styling the content. WordPress requires a commented out declaration of the site info in the very beginning of this site in order for the theme to be recognized as a valid one. Here's a screen from the Sports Academy site's main css-file:

```
1 /*
2 Theme Name: Kouvolan Urheiluakatemia
3 Theme URI:
4 Author: Aaro Helander
5 Author URI:
6 Description: A WordPress theme for Kouvolan Urheiluakatemia
7 Version: 1.0
8 License: GNU General Public License v2 or later
9 License URI: http://www.gnu.org/licenses/gpl-2.0.html
10 Text Domain: Kouvolan Urheiluakatemia
11
12 This theme, like WordPress, is licensed under the GPL.
13 Use it to make something cool, have fun, and share what you've learned with
14 */
```

Figure 3 - Style.css with the proper initialization

- Other Useful Template Files
- **comments.php** - Includes the code for outputting comments in a blog's comment section. There are also plugins for taking control of this file and to control the comment functionality from the WordPress administration panel.
- **single.php** - Defines the layout of a single blog post. It contains a loop which displays a single blog post from the database.
- **page.php** - Defines the layout of individual pages on your site. It is important to remember to distinct between a page and a post - which are two of WordPress's main content types.
- **archive.php** - This template file is used to create the layout for a blog archive page. Without this file WordPress will look for the **index.php** file and display the blog archives according to this file.
- **front-page.php** - If this template file is present, WordPress will display its contents as the front page for your site If you have "Display static front page" -setting enabled in the WordPress administration panel.

The full overview of all available template files can be found in the WordPress Codex page called Template Hierarchy. The page includes a visual representation of every template file, how they are arranged in the hierarchy and what their relations to each other are. (WordPress 2015 e)

## 4.6 Child Themes

Using a child theme is considered one of the best practices of WordPress theme development. A child theme is an individual theme folder, which inherits all of the styles and content from the parent theme. The practice is similar to editing a copy of a photo instead of editing the original one: In case you mess up your editing, you can easily revert back to the original file and start over.

Even a bigger benefit than the previously mentioned is that since premium themes are often updated regularly by the developer, having your customizations in a child theme instead of the parent means that there is no risk of losing your customizations in case that the parent theme is updated and some files are overridden.

With my final design using Divi 2, the child theme only consist of one file: style.css. Other files and code can be inserted into the child theme to override the parent theme functionality.

```
/*
Theme Name:     Divi2 child theme
Author:         Aaro Helander
Author URI:     Aaro Helander
Template:       Divi
Version:        1.0
*/

@import url("../Divi/style.css");
```

Figure 4 - Child theme with styles imported from the parent

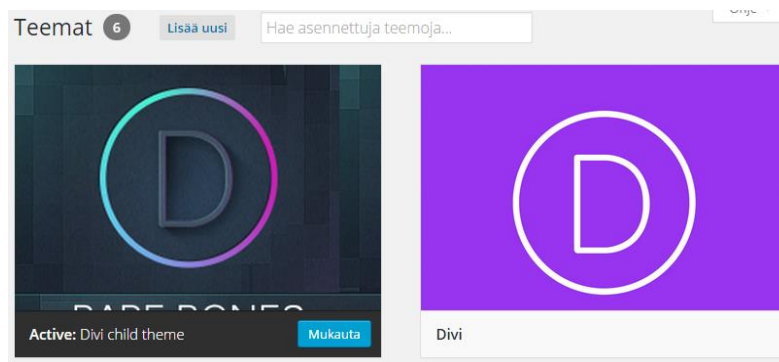


Figure 5 – The Child theme activated from the theme selection panel

## 4.7 WordPress Theme Frameworks

A Theme framework (not to be confused with the front end frameworks discussed earlier) is a collection of tools used to build WordPress Themes. They are often likened to a framework for a building consisting of the foundations and the building frame that set the

limits of what kind of a building you can create. For example, a popular framework called Genesis Framework sells for \$59,95 and allows for creating highly customized looking pages with a secure, search engine optimized and future proof foundation under the hood. The Elegant Themes developer membership for one year costs \$89, which transferred to around 110 € with VAT included. (Livingston 2015)

There are many high quality theme frameworks available from various developers, and some are intended for more advanced developers – such as the Genesis Framework - requiring more knowledge of PHP and WordPress, and others, like drag and drop builders such as the Elegant Themes Divi 2 are more easy to get started with, but still offer much room for customization for the advanced developers.

For a detailed review of the most popular frameworks I recommend reading the article “12 Best WordPress Theme Frameworks” on AThemes, found in the references for this chapter.

#### **4.7.1 Deciding between Genesis Framework and Elegant Themes**

Out of personal interest, I started to look for a framework when doing my thesis, and had a hard time choosing between the Genesis framework and the Elegant Themes membership, which included their Divi 2 theme builder.

I ended up choosing Elegant Themes, because I thought it had a greater business value for a developer like me, who intends to develop mainly simple portfolio websites quickly, while focusing on something else at the same time. According to my research I did on the subject, Genesis had too high of a learning curve to suit my schedules for different projects, and Elegant Themes was not only offering all of their Themes (87 themes at the moment of writing) but also some very nice plugins and, of course, the Divi 2 theme, which allowed creating sites fast with still enough control to create unique site features.

### **4.8 The WordPress Loop**

The site structure and styling is controlled by the template files. All content that is created using the WordPress page and post editor are displayed by using what is called The Loop. This is a piece of PHP-code that simply iterates through the database fetching all content available. The loop can be modified with different parameters to fetch content that is specific for the section of the site you are working with.

A very basic version of the loop shown below fetches all posts available:



```
1 <?php
2     if (have_posts() ) :
3         while ( have_posts() ) :
4             the_post();
5
6             // All post content shows here!
7
8         endwhile;
9     endif;
10 ?>
```

Figure 6 - A very basic loop

The above code is pretty easy to understand even for a person with little understanding of programming languages. In short, it has an **if**-statement that checks if there exists one or more posts in the database. If there are posts available, a **while** loop gathers every post and displays them.

It is out of scope of this section to provide a thorough review of the WordPress loop and other components.

#### 4.9 WordPress Plugins

One of the reasons for the popularity of WordPress is the huge amount of high quality free and paid plugins available. Plugins are used to extend the basic functionality of a WordPress site. Most common functionalities handled by plugins are related to Search Engine Optimization, creating contact forms with mail sending capabilities, controlling commenting of blog posts and helping with the security of the site.

At the moment, over 35000 plugins are available with almost a billion downloads so far. Plugins can be searched in the WordPress plugin directory. The same view is also integrated into the WordPress Administration panel, which allows for searching and installing plugins directly on your site without having to download the files separately.

While many high quality plugins are available, it can be a tedious process to find the right ones which really work the way you want them to work. Not every plugin works as you might think, and it is up to the developer to decide whether some functionality should be handled by a plugin or be built into the core of the site. The best case scenario is that using a plugin saves both time and effort, and provides a higher quality of functionality for the site than what the developer would be able to pull off by hand coding the functionality himself. (Onishi 2013 c)

It is important to note that the more a site has plugins installed and activated, the more there is a chance for site performance decreases due to badly written plugins. It is also important to take in note the potential security and compatibility issues which might occur when haphazardly using random plugins to carry out functionalities. It is also common for developers to use a plugin to handle some functionality which should be part of the WordPress core functionality, causing wasting of time and resources. (Onishi 2013 d)

#### **4.10 Plugin dependencies and security**

As noted previously, not all plugins are of high quality, and even some of the high quality plugins can cause compatibility issues with some other plugin that is in use. It can be a time consuming process to track down an issue with your site that is caused by a plugin dependency or compatibility problem. On our initially drafted my site, a plugin called “Easy Bootstrap Shortcodes” caused an issue for the site’s dropdown menu not working due to the Bootstrap CSS and JavaScript files being initialized both in the **header.php** of my site, and also by the plugin. Disabling the initialization of these files from the plugin settings resolved this issue, but it took some time and serious googling to figure out what was wrong. (Onishi 2013 e)

When selecting a plugin it is of essence to check out when was the last time a specific plugin was updated. As anyone checking out the WordPress plugin directory will see, out of all the available plugins there are many which haven’t been updated for months or even years. From a general security viewpoint it is important that your core WordPress installation is always up to date, and some plugins might not be compatible with the latest WordPress version at the time. For making things easier, the WordPress plugin directory shows all relevant data about the compatibility of any plugin with the WordPress version you have installed. If a plugin hasn’t been tested with the WordPress version you have installed, it will be declared in the plugin overview.

Before deciding to use a specific plugin it is a good practice to spend some time digging around WordPress related forum conversations about that plugin. As is usually the case with most issues of the first world, often people have had the same problems you might end up having with a certain plugin, and it will save you a lot of time and frustration to read all you can about any plugin that you’re about to install. I ran into multiple frustrations with plugins during this development project, and spent hours trying to solve issues caused by plugin clashing with another.

Basically, the more minimal your WordPress installation is, the less room there is for a badly coded plugin to cause a potential security threat to your site. The last thing a developer needs is to have his client's site hacked due to a badly written or un-updated plugin.

## 5 WordPress Theme Security

### 5.1 What is a safe WordPress theme?

A few important considerations need to be taken into account when deciding between using a self-made theme with custom functionality, and purchasing a paid theme with guaranteed security updates. Getting an angry phone call about your site being hacked six months after you left your customer enjoying their new site is not something you will want to be stressing about during your vacation. Although a fairly simple theme with not much special functionality will probably not going to be hacked, and there will be lots of other factors to this not related to your theme, counting on mere good luck might backfire in the near future.

Most of the free themes available are probably safe, but many themes written by individual developers can be flawed in several ways, since not all developers are too serious with their development. A more detailed analysis of this subject is out of the scope of this subject.

### 5.2 Clean code

Defined in an article by WPSecurity, clean code means that the template files of your theme do not contain intentionally harmful code placed there by the developer. It is a known fact that there are sites offering premium WordPress themes for free. The problem is that these premium themes are probably altered with some form of malicious code, and unless you're a profane PHP and WordPress expert, you're going to have a hard time going through all of the template files and knowing what's really in there. (WpSecurity 2015 a)

Some guidelines for buying or choosing a free theme are:

- When purchasing a premium theme, prefer reputable companies like Elegant Themes, StudioPress and Woocommerce who have a large scale business and offer regular, guaranteed updates and support for their paid customers.
- Support for an individual theme by an individual developer can be a hit and miss venture, and finding support can be hard.
- In the long run, going with a fairly expensive premium theme or a membership for a theme company can actually end up being the cheapest alternative, when thinking about the time and effort you might have to spend trying to eliminate problems or add features to a theme made by yourself or a cheap theme with no support offered by the developer.

### 5.3 Plugin safety

Choosing the right plugins can be even more taxing process than choosing the right theme. The WordPress plugin directory contains a fair amount of plugins which aren't actively being developed anymore and might not be even compatible with the latest WordPress version available.

An old or non-optimally written plugin can not only pose a potential security threat to your site, but can also destroy your business by causing serious performance issues and issues with other functionality on your site, making your site traffic drop off dramatically. While WordPress is one of the simplest applications to use, it is good to remember that under the surface it is a very complex piece of software. Making an already complex application even more complex by adding some haphazardly selected components to it can make keeping all strings under control can be an impossible task. (Forrester, 2013)

In some instances one plugin by itself won't cause any problems, but can present a serious problem with security when combined with another plugin. The general good practice is to choose all plugins carefully and keeping the amount of active plugins to the very minimum.

### 5.4 Security breaches

Since WordPress has become a huge business, fixing hacked WordPress sites has also become a nice form of business for many companies. WPSecurity - a company specialized in WordPress security problems - references a list of typical actions taken when fixing a hacked site:

- All site files and theme template files are inspected for malicious code and backdoors.
- The site database is inspected and cleaned, or deleted and rebuilt in some cases.
- In some cases a new theme is installed if the previous one is seriously flawed, and some plugins might have to be changed.

Fixing a hacked WordPress site is considered custom work, and the starting price for repairs is listed at \$1,250 on wpsecurity.com. Keeping all of your site components updated will be essential in ensuring that you won't be running into some serious trouble in the long run. (WPSecurity, 2015 b)

## 6 Converting a Bootstrap 3 Template to a WordPress Theme



### Bootstrap starter template

Use this document as a way to quickly start any new project.  
All you get is this text and a mostly barebones HTML document.

Figure 7 - A basic Bootstrap 3 template (the rendered HTML)

The image above is a screenshot of a simple site layout built with Bootstrap 3. A Bootstrap template consists of a handful of files (the source files linked to the header of the site): The **bootstrap core CSS files (minified)**, **optional theme file** and the **bootstrap JavaScript file**. Bootstrap also requires **jQuery** to work, and the latest jQuery version is included in the footer of the **index.html** file included with the template.

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/css/bootstrap.min.css">
<!-- Optional theme -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/css/bootstrap-theme.min.css">
<!-- Latest compiled and minified JavaScript -->
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/js/bootstrap.min.js"></script>
```

Figure 8 - The files required by Bootstrap, included through a CDN

When these files are included in the header of the site, all of the components provided by the framework are available to be called using their representative HTML tags.

The Bootstrap home page has several pre-built layouts available, and anyone can just copy the source code and save the files to get started with building a web site. The **index.html** file looks like this:

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <title>Bootstrap 101 Template</title>
8
9     <!-- Bootstrap -->
10    <link href="css/bootstrap.min.css" rel="stylesheet">
11
12    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -
-->
13    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
14    <!--[if lt IE 9]>
15    <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
16    <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
17    <![endif]-->
18  </head>
19
20  <body>
21    <nav class="navbar navbar-inverse navbar-fixed-top">
22      <div class="container">
23        <div class="navbar-header">
24          <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#navbar" aria-expanded="false" aria-controls="navbar">
25            <span class="sr-only">Toggle navigation</span>
26            <span class="icon-bar"></span>
27            <span class="icon-bar"></span>
28            <span class="icon-bar"></span>
29          </button>
30          <a class="navbar-brand" href="#">Project name</a>
31        </div>
32        <div id="navbar" class="collapse navbar-collapse">
33          <ul class="nav navbar-nav">
34            <li class="active"><a href="#">Home</a></li>
35            <li><a href="#about">About</a></li>
36            <li><a href="#contact">Contact</a></li>
37          </ul>
38        </div><!-- /.nav-collapse -->
39      </div>
40    </nav>
41
42    <div class="container">
43
44      <div class="starter-template">
45        <h1>Bootstrap starter template</h1>
46        <p class="lead">Use this document as a way to quickly start any new project.
<br> All you get is this text and a mostly barebones HTML document.</p>
47      </div>
48    </div><!-- /.container -->
49
50    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
51    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js">
</script>
52
53    <!-- Include all compiled plugins (below), or include individual files as needed -->
54    <script src="js/bootstrap.min.js"></script>
55  </body>
56 </html>

```

Figure 9 - The index.html of the Bootstrap Template

This template file is converted to a simple WordPress theme with a couple of pretty easy steps:

1. Create a theme folder and the previously mentioned required **WP** template files
2. Copy the main CSS (not bootstrap.css) from the existing page into the **styles.css** file and add the info section to the top of this file.
3. Divide the HTML code from **index.html** into three parts
  - **header.php** - All code from the first line to the closing tag of the navigation

- **index.php** - All code from the jumbotron to the last closing tag of the main content container
  - **footer.php** All code beginning from the footer tag to the very last line of code
4. Invoke the **header.php** and **footer.php** from within the **index.php** by adding the code `<?php get_header(); ?>` to the beginning of the file and `<?php get_footer(); ?>` and `<?php get_sidebar(); ?>` to the end of the file.
  5. Finally, add The Loop inside your main content section. Your **index.php** file should now look like this:

```

index.php
1  <?php get_header(); ?>
2
3  <div class="container">
4
5      <div class="starter-template">
6          <h1>Bootstrap starter template</h1>
7          <p class="lead">Use this document as a way to quickly start
any new project.<br> All you get is this text and a mostly barebones
HTML document.</p>
8      </div>
9
10     <div class="col-md-9">
11         <?php if ( have_posts() ) : while ( have_posts() ) :
the_post(); ?>
12
13             <div class="page-header">
14                 <h1><?php the_title(); ?></h1>
15             </div>
16
17             <?php the_content(); ?>
18
19             <?php endwhile; else: ?>
20
21             <div class="page-header">
22                 <p>No content available!</p>
23             </div>
24
25             <?php endif; ?>
26         </div>
27
28         <?php get_sidebar(); ?>
29
30     </div><!-- /.container -->
31
32
33 <?php get_footer(); ?>

```

Figure 10 - Index.php with header and footer called, and the loop implemented

Likewise, the **header.php** and **footer.php** files should look like this:



```

header.php
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <title>Bootstrap 101 Template</title>
8
9     <!-- Bootstrap -->
10    <link href="css/bootstrap.min.css" rel="stylesheet">
11
12    <?php wp_head(); ?>
13  </head>
14
15  <body>
16    <nav class="navbar navbar-inverse navbar-fixed-top">
17      <div class="container">
18        <div class="navbar-header">
19          <button type="button" class="navbar-toggle collapsed" data-
20toggle="collapse" data-target="#navbar" aria-expanded="false" aria-controls="navbar">
21            <span class="sr-only">Toggle navigation</span>
22            <span class="icon-bar"></span>
23            <span class="icon-bar"></span>
24            <span class="icon-bar"></span>
25          </button>
26          <a class="navbar-brand" href="#">Project name</a>
27        </div>
28        <div id="navbar" class="collapse navbar-collapse">
29          <ul class="nav navbar-nav">
30            <li class="active"><a href="#">Home</a></li>
31            <li><a href="#about">About</a></li>
32            <li><a href="#contact">Contact</a></li>
33          </ul>
34        </div><!--/.nav-collapse -->
35      </div>
36    </nav>
37

```

Figure 11 - The header.php with a navigation bar implemented

```

footer.php
1   <?php wp_footer(); ?>
2
3   <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
4   <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js">
5     </script>
6
7   <!-- Include all compiled plugins (below), or include individual files as needed -->
8   <script src="js/bootstrap.min.js"></script>
9 </body>
10 </html>

```

Figure 12 - The footer.php with scripts included

So we basically just chopped the **index.html** into three parts from top to bottom and used some WordPress's functions to call the other parts into play. Take a note that at this point the stylesheets and scripts are included in these files, but following WordPress best practices they will be included via the functions.php file.

## 6.1 An overview of the initial site structure

The **functions.php** template file is used to customize the default functionality of WordPress. When building the theme from the Bootstrap template, I used custom WordPress functions to include

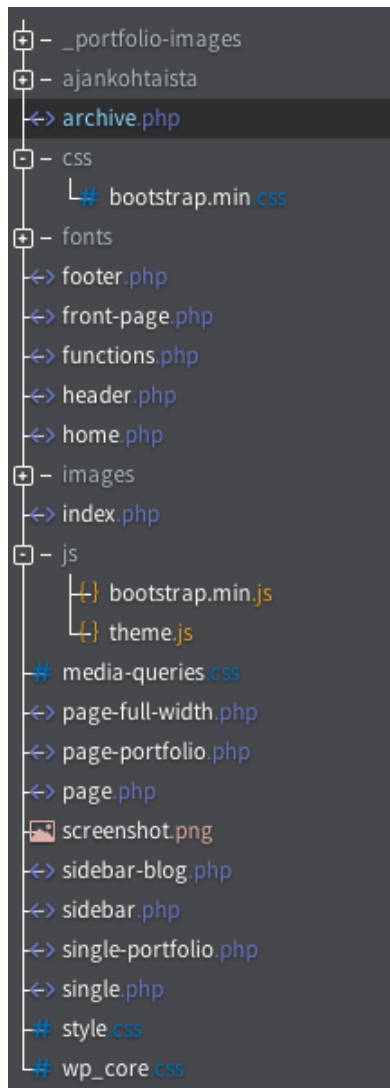
- All of the style sheets: bootstrap.min.css, the custom style.css, and the WordPress core css wp\_core.css.
- JavaScript files: html5\_shiv, respond\_js and the latest version of jQuery.
- Add the capability for a custom menu structure that can be modified from the WordPress Administration Panel by authenticated users
- A capability for having post thumbnails to be used with post excerpts etc.
- A custom Widget creation function for the capability of adding pre-formatted content sections to the page from within the WordPress Administration Panel

When dealing with static web pages, all external files such as stylesheets are included to the site in the header and footer sections of the index.html. As mentioned previously, with WordPress, the recommended practice is to include them via the functions.php file.

(WordPress 2015 f).

## 6.2 Used Template Files (On our initial Bootstrap based theme)

- **wp\_core.css** - The WordPress' internal styling which is applied when formatting posts from within the post editor (image placement, text placement etc.).



- **style.css** - All of the customized css used to override or to add to the bootstrap core css files.
- **single.php** - This template file determines the page layout for a single post displayed (when opening up a post from an excerpt for example).
- **sidebar.php** - This template is used for the sidebar which can be used with all of the pages (the blog page has it's custom sidebar).
- **sidebar-blog.php** - This template is used for the sidebar specific to the blog (displaying blog archives and whatever the content creators want to be displayed).
- **page.php** - The default template file for controlling the layout for individual pages. This template can be overridden with an `page_id` or `page_slug` template (`page_hobbies` for example) if the page called "hobbies" needs to have a custom layout.
- **page-full-width.php** - Here is an example of a template used to override the **page.php** file. For example, our theme has the possibility of choosing a full width -layout or the generic page-layout from the WordPress Administration Panel.
- **home.php** - This template controls the look of the blog listing page.
- **archive.php** - This template is used for setting the layout for the post archive page.

WordPress was originally just a blogging platform, and this is one of the reasons it has different templates for displaying a blog page and a static home page (Gordon 2015 b). Our site, like most websites today uses a static front page as our home page instead of the blog page, since it is more suitable for marketing for the company using visuals and relevant pieces of information.

WordPress uses its so-called fallback nature to prioritize more specific templates to override the generic templates, and if a more specific template is not present, WordPress will

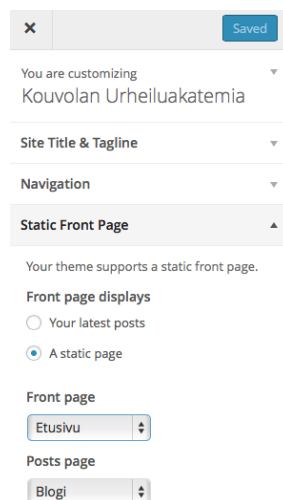
fall back to the next template in the hierarchy to display the site. For example, the **front-page.php** template is used to display the static front page we have as our home page, and when this template is present, the **home.php** falls back to control the blog page layout. (Gordon, 2014 c).

If the administrator decides to not use a static front page, then the **home.php** will be used as the template for displaying the front page, and the **front-page.php** is ignored, even if the file is present in the folder.

It is a good practice to design both templates to look good, so that if the site administrator decides to switch the blog page to be the home page, it will still look good and make the website recognizable as the company's site.

As shown in the WordPress Hierarchy Chart, **index.php** is the most generic template, and WordPress will fall back to it if **home.php** is not present.

To enable a static front page, on our site that will be the page "Etusivu", we select the "Customize" section from the WordPress Administration Panel, and make sure that the "Front Page" drop down list has the desired page selected. Here we also set the "Blogi" -



page to act as the page for posts.

Now that the static front page is enabled, WordPress will use the **front-page.php** template to be used as the home page of our site.

## 7 Changing the Theme

The most exciting part of this project was when I realized that I'm far better off throwing my own design out of the window, and using the Divi 2 theme from Elegant Themes to create the same layout again. As I mentioned previously, I had a hard time choosing between Genesis Framework and Elegant Themes, but ended up with Elegant Themes.

There were many reasons to do the switch. First of all, the Elegant Themes developer membership which I purchased included a handful of very useful plugins for social media integration and email marketing which I had already used on various other sites I had been working on during this spring. By doing so I also managed to get rid of many plugins I was using to implement some functionality on the site. Some of these plugins (Easy Bootstrap Shortcodes) had already caused me some problems, and maintenance of the site was far from optimal.

A big bonus with changing to Divi was that I was free from maintaining my own theme and keeping it up to date.

## 7.1 Divi 2 by Elegant Themes

It took me about a week to recreate the site with Divi. I kept the design pretty much identical to the one I had created with Bootstrap, but managed to easily add a lot of new functionality, visual effects and other minor stuff utilizing Divi's inbuilt features. The result was almost identical. although more pleasing, to the original, but a lot easier to modify and maintain. Here's an overview of the Divi Site Builder which allows for easily creating site designs:

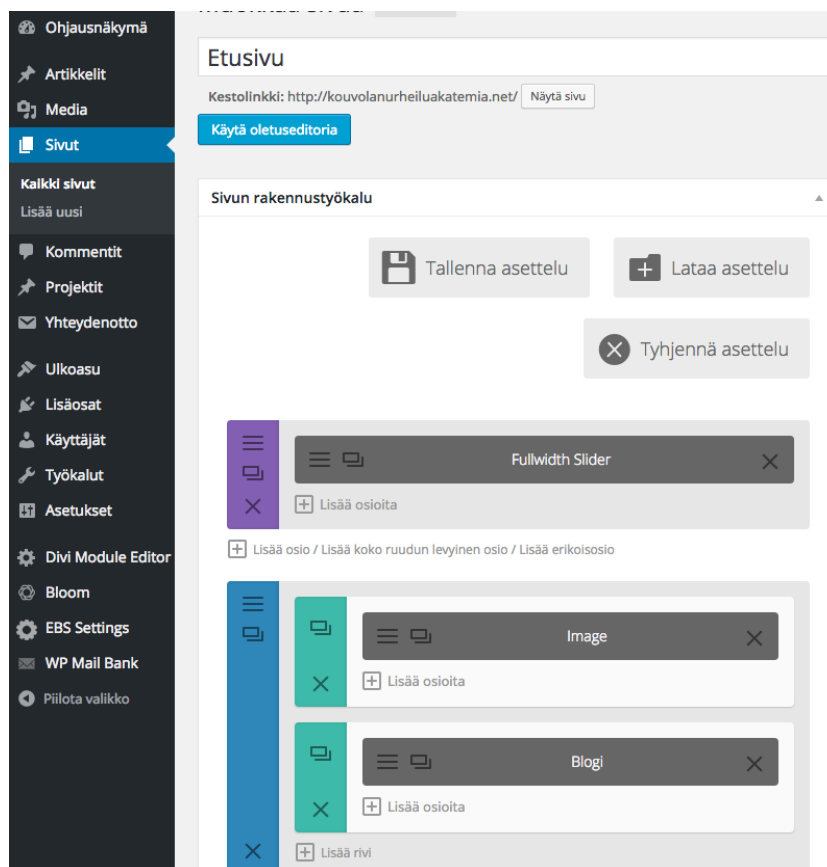


Figure 13 - The Divi Page Builder

## 7.2 Pros of using a premium theme

Theme frameworks seem to be the future of WordPress theme development, and during this thesis I have concluded that a “business oriented” developer like me is best off choosing a framework which allows fast development of sites with good customization possibilities. All in all, the obvious benefits to me are:

- Speed of development and trying out different approaches to design.
- All frameworks are built with security as a high priority. When developing your own theme, you will need to have a bit higher knowledge about security.
- I was able to get rid of some plugins I previously used for the site layout. Google site speed test ranks higher pages that don't rely on too many plugins. (Google 2015)
- Pages automatically scale pretty well on mobile, although sites built with Bootstrap 3 or any other responsive framework do this as well.
- A solid framework, but still huge amounts of customizability.
- Future security and feature updates. For example, Elegant Themes is releasing Divi 2.4 this spring, which is said to bring a vast amount of new features to the theme.
- Developer and community support. Since a large amount of developers around the world are using the same framework, it is easy to find support for your problems.
- Do It Yourself can be fun, but most of the time it is not profitable. If your competitors are using a framework to do things in on tenth of the time you spend doing the same thing by yourself, you're playing a losing game.

## **7.3 Plugins I used with Divi 2**

Switching to Divi from my self-made theme allowed me to get rid of some plugins (Bootstrap Shortcodes for example) and made controlling the site layout a lot easier. The Elegant Themes membership also includes some high quality plugins for social media marketing and email marketing, which alone would justify the price of the membership (around 110 € with VAT included).

### **7.3.1 Divi Booster**

Divi Booster is an unofficial collection of bug fixes and other modifications to the core Divi theme. It includes several bug fixes and modifications for the design elements provided by the theme.

### **7.3.2 Divi Module Editor**

The Divi Module Editor is a bit similar to Divi Booster: An unofficial plugin featuring several CSS-related tweaks to control the layouts of the site. It is being developed and maintained by a company called Gritty Social and the philosophy behind the project is to implement features and fix problems commonly encountered by Divi theme users.

### **7.3.3 Contact Form 7**

Contact Form 7 is a free and widely used plugin that allows the creation of custom contact forms that are able to send their data through a mail function to a specified email address. On our site I used this plugin to create a sign up form for a summer sports camp arranged by the Sports Academy. Sometimes it is not as simple as it should be to have your server send mail, but most hosting providers have pre-configured their WordPress hosting solutions with mail-sending capabilities. Sometimes the mail will be sent, but some mail clients classify the received mail as spam due to incomplete sender information.

### **7.3.4 WordFence**

WordFence is a free security plugin with numerous security features. There are many alternatives plugins with similar, more or less overlapping set of features. I ended up installing WordFence as the new default security plugin for every site I have created so far, due to ease of use and the fact that it also features a very good caching engine called Falcon Engine, which claims to speed up site performance up to 30 times of the original.

## 8 Summary of results

This project resulted in two webpages: The main site for the Sports Academy featuring easy blogging capability for multiple users, a small scale marketing page for a sports camp for young athletes and a specific marketing page for the Yleistaitovalmennus project on which I had worked with the Sports Academy as of spring 2014.

Until this spring the main method of applying for events like sports camps etc. was purely email or “pen and paper” based. Now the site features a registration form for the summer camp, and after the summer we are going to implement an electronic registration for every activities provided by the academy.

I was also assigned a new project by Kymenlaakson Liikunta (the parent company of the Sports Academy), which will be completed during the summer.

## 9 Conclusions

This project didn't start out with specific goals or even an actual need from the customer's side. This is actually not that uncommon, especially with freelancers. Being able to ignite ideas in the mind of the potential client is an important skill.

Only seeing the process and seeing the site design and proposed ideas taking form made the company heads excited about what potential this development would bring them in the future.

The project was lagging behind the initially planned schedule, since many of the changes and new concepts that were presented would have required more human resources in the future to be actually useful.

The problematic cycle of creating a new feature goes like this: Creating a new feature for something like marketing is not very useful if there is no-one utilizing it to actually create the marketing. Creating an effective tool for something requires a consistent feedback loop between the developer and the product user. If no-one is using the product and providing feedback, it's up to the developer's imagination to keep making the product more useful.

The company budget for this year was already set, and although people got really excited about the new website and its potential, there were not many resources available to get a proper feedback loop running.



Discussion forums for web developers are filled with threads about failed or soon-to-be-failed projects that ended up costing the developer more than what was earned. As a freelancer it is very important to make proper deals with clear conditions before starting any project. Since I had personal connections and personal motives to do this project deal where everybody would get something they needed, and since I had my personal motives to pull off this project, the ambiguity was acceptable.

The good part of not having enough human resources was that the company actually hired me to coordinate the implementation of their future marketing through the website, and I also managed to get another project do be done during summer.

In the process I ended up scratching the custom theme I had made, because I saw no reason not to use my new membership benefits and the theme to re-create these sites. I think that developers should never be too attached to something they created if replacing it with something else will be a better option long term.

Did I waste time by starting the development from scratch and then changing it for a totally different solution? I don't think so. By doing everything by myself from scratch I learned a lot about how WordPress functions on a very basic level, and I also have now a much deeper understanding of how different frameworks work and how to effectively modify an existing theme to suit my needs.

In the future I myself will be the on taking care of the maintenance and managing the content production of the site, which means that with this project I created a part-time job for myself.

## 10 Appendix

The URLs for the sites are:

[www.kouvolanurheiluakatemia.net](http://www.kouvolanurheiluakatemia.net)

[www.yleistaitovalmennus.fi](http://www.yleistaitovalmennus.fi)

### 10.1 Overview of the sites

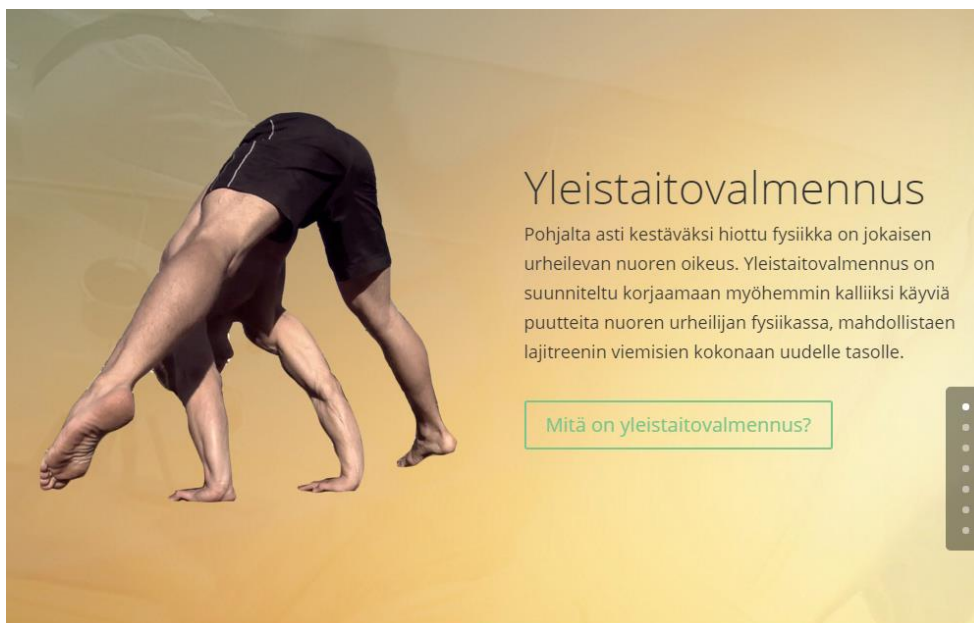


Figure 14 - yleisheitovalmennus.fi front page

## Valmentajat



**Petri Pätäri**  
Liikunnanopettaja

Petri Pätäri toimii päätoimisesti liikunnanopettajana Valkealassa, ja valmentaa ohella junioreiden jalkapalloa. Petu on Kymiin palkitsema vuoden 2015 nuorisovalmentaja.



**Aaro Helander**  
Personal Trainer, GymnasticBodies Athlete

Aaro toimii fysiikkavalmentajana Urheiluakatemialla erikoisosaamisenaan voimistelupohjainen voima- ja liikkuvuusharjoittelu, sekä mentaalinen valmennus. Aaro on EHFA-sertifioitu personal trainer, sekä sertifioitu GymnasticBodies Athlete.



**Teemu Turpiainen**  
Liikunnanopettaja

Teemu toimii opettajana Korian koululla sekä kartanokoululla. Valmennuspuolella Teemun erikoisosaamista ovat motorinen oppiminen, ryhädynamiikkojen tutkimus sekä hyödyntäminen yhdessä nuorten tunnetaitojen kehittymisen kanssa

Figure 15 - yleistaitovalmennus.fi coaches section

### Ilmoittaudu!

<input type="text" value="Name"/>	<input type="text" value="Email Address"/>
<input type="text" value="Message"/>	
$10 + 3 =$ <input type="text"/>	
<input type="button" value="Submit"/>	

<b>Kouvolan Urheiluakatemia</b> Ilmarinkuja 3 45100 Kouvola timo.pahkala@kymli.fi	<b>Valmentajat</b> Petri Pätäri Aaro Helander Teemu Turpiainen	<b>Valmennuspaikat</b> Valkealatalon urheilusali Tornionmäen koulun sali
---	--	--

Figure 16 - yleistaitovalmennus.fi registration form



Figure 17 - The front site



Figure 18 - The blog section on front site

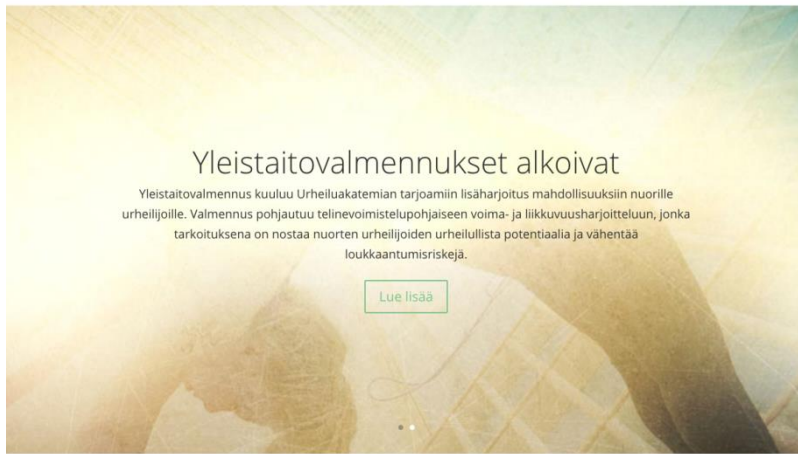


Figure 19 - The front site slider

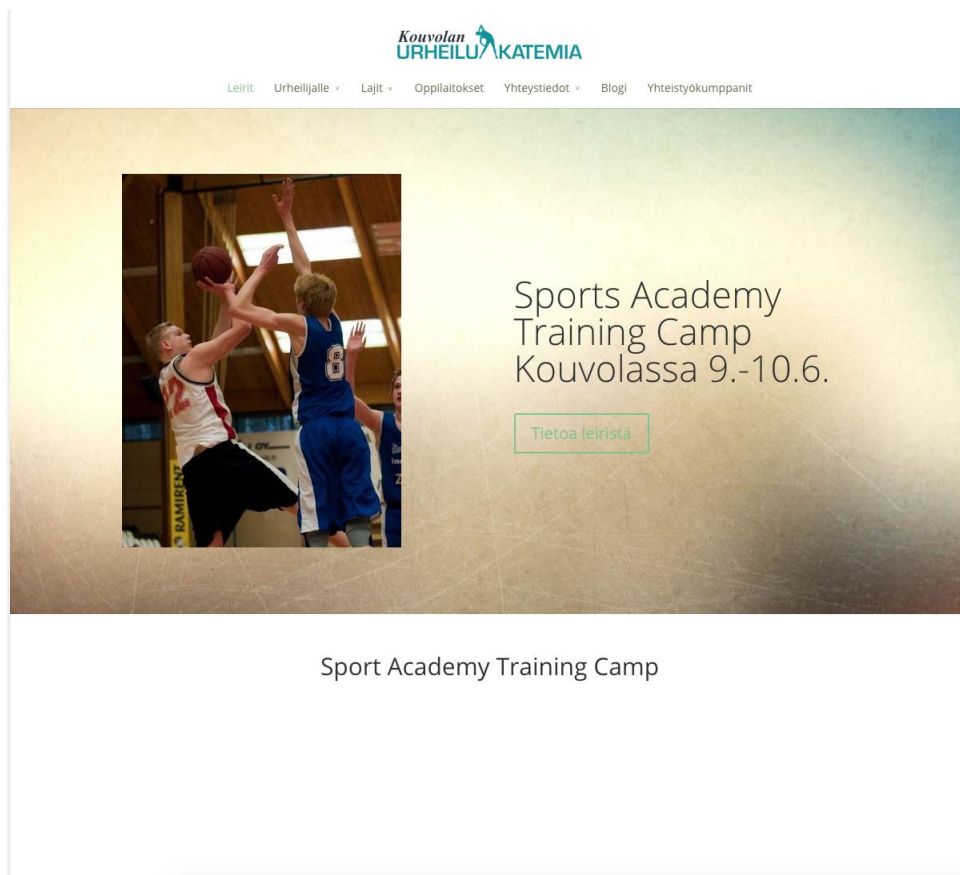


Figure 20 - The summer camp page



**Kouvolassa**  
Keskuspaikkana toimii Mansikkalahon urheiluhalli ja sen lähiympäristö



9.-10.6.2015



**Ohjelma**  
Yleistaitovalmennus, ravitsemustietoa, lihashuoltoa, sekä oman valinnan mukaan lajiharjoittelua (koripallo/jalkapallo) tai kestävyys- tai voimaharjoittelua



**Hinta 55 € / Urheilija**  
Sisältää valmennuksen, luennot, vakuutuksen, ruokailut ja välipalan



Ilmoittaudu!

Ilmoittautuminen leirille

Figure 21 - The summer camp page section 2

## Ilmoittautuminen leirille



Etunimi

Sukunimi

Syntymävuosi

Katuosoite

Postinumero

Postitoimipaikka

Puhelinnumero

Sähköpostiosoite

Huoltajan nimi

Ilmoittaudun Sport Academy Training campiin (ratulossa)

Koripallo  Jalkapallo  Kestävyys  Voi

Ruoka-aineallergiat:

onko lapsellanne erityistarpeita? Esim. lääkitys tms:



Figure 22 - The summer camp registration form (Contact Form 7 embedded into a Divi Builder text module)

## 11 References

Entrepreneur 2015, Starting a Consulting Firm? 10 Reasons Your Clients Need You. 28.4.2014. URL: <http://www.entrepreneur.com/article/233300>. Accessed 20.4.2015

Zaman, T. 2015. How Much Does a WordPress Site Really Cost? 13.4.2015 URL: <https://codeable.io/how-much-does-a-wordpress-site-really-cost/>. Accessed 20.4.2015

Jain, R. 2014. Making a Case for Mobile First Designs. 21.11.2014 URL: <http://www.sitepoint.com/making-case-mobile-first-designs/> Accessed 20.4.2015

The WordPress Codex 2015. Installing WordPress. URL: [https://codex.wordpress.org/Installing\\_WordPress](https://codex.wordpress.org/Installing_WordPress). Accessed 1.3.2015

Github 2015. Bootstrap. URL: <https://github.com/twbs/bootstrap> Accessed 11.2.2015

Onishi, A. 2013 1-2. Pro WordPress Theme Development. Expert's Voice in Web Development. Apress; 1 edition. Chapter 1: An Introduction to WordPress (November 26, 2013)

Onishi, A. 2013 1-2. Pro WordPress Theme Development. Expert's Voice in Web Development. Apress; 1 edition. Chapter 9: Plugins: When the Time is Right. (November 26, 2013)

Livingston, C. 2015. AThemes. 12 Best WordPress Theme Frameworks. 29.1.2015. URL: <http://athemes.com/collections/best-wordpress-theme-frameworks/>. Accessed 10.5.2015

"Content Management System" Wikipedia: The Free Encyclopedia. Wikimedia Foundation Inc. URL: [http://en.wikipedia.org/wiki/Content\\_management\\_system](http://en.wikipedia.org/wiki/Content_management_system). Accessed 28.4.2015

Team Treehouse Inc. Should You Use Bootstrap or Foundation? 1.7.2014. URL: <http://blog.teamtreehouse.com/use-bootstrap-or-foundation>. Accessed 1.3.2015

The WordPress Codex 2015. Theme Development. URL: [https://codex.wordpress.org/Theme\\_Development](https://codex.wordpress.org/Theme_Development) Accessed: 2.3.2015

Google 2015. Page Speed Tools. URL: <https://developers.google.com/speed/pagespeed/>. Accessed 14.5.2015



The WordPress Codex 2015. Functions File Explained. URL :  
[https://codex.wordpress.org/Functions File Explained](https://codex.wordpress.org/Functions_File_Explained). Accessed 3.3.2015

Forrester, M. 2013. The Dangers of WordPress Plugins Ignorance (And What To Do About It). 27.9.2013. URL: <http://www.woothemes.com/2013/09/wordpress-plugins-ignorance/>. Accessed 8.5.2015

Gordon, Z. 2014. From Bootstrap To WordPress. Team Treehouse Inc. Video series (member subscription).

