ARCADA

Choosing Craft CMS over Wordpress

A look at two different content management systems from a developer's perspective

Joel Ström

EXAMENSARBETE	
Arcada	
Utbildningsprogram:	Online Media & Art Direction
Identifikationsnummer:	15378
Författare:	Joel Ström
Arbetets namn:	Choosing Craft CMS over Wordpress
Handledare (Arcada):	Owen Kelly
Uppdragsgivare:	Yrkeshögskolan Arcada

Sammandrag:

Då det gäller att välja ett CMS finns det flera saker som måste tänkas på. I dagens läge är det allt viktigare att systemet man väljer passar projektet i fråga. Det finns en hel del system på marknaden som löser problem på olika sätt, men i detta arbete kommer jag att koncentrera mig på de populäraste systemet, Wordpress, samt Craft CMS som är en relativt ny tävlare. Jag kommer att ytligt gå igenom vad som CMS är, varför man skulle använda CMS, samt introducera både Wordpress of Craft CMS. Jag kommer att gå djupare in på hur båda systemen används, och demonstrera kod med hjälp av praktiska exempel. Allt detta sker från en webbutvecklares perspektiv. För att komma till en slutsats har jag gjort flera jämförelser mellan dessa två system. Förutom mina egna åsikter och resultat analyserar jag data som jag samlat från en frågeformulär som skickades åt andra utvecklare, och från en expert intervju var jag intervjuade två kollegor från min arbetsplats.
En längre sammanfattning på svenska kan hittas i slutet av arbetet.

Nyckelord:	CMS, Wordpress, Craft CMS, jämförelse, webbutvecklare
Sidantal:	79
Språk:	Engelska
Datum för godkännande:	1.6.2017

DEGREE THESIS	
Arcada	
Degree Programme:	Online Media & Art Direction
Identification number:	15378
Author:	Joel Ström
Title:	Choosing Craft CMS over Wordpress
Supervisor (Arcada):	Owen Kelly
Commissioned by:	Yrkeshögskolan Arcada

Abstract:

There are multiple different factors to think about when choosing a CMS for your web project. In today's world, it is important that the system you choose fits the project at hand. There are a whole lot of systems on the market that try to solve things in a different manner, but in this thesis I will focus on the most popular system, Wordpress, as well as Craft CMS which is a relatively new CMS. I will briefly go through what a CMS is, why you would want to use a CMS, as well as introduce both Wordpress and Craft CMS. I will go deeper into details on how both systems are used and demonstrate code with the help of practical examples. All of this is done from a developer's perspective. In order to come to a conclusion, I have made multiple comparisons between the two systems. Besides my own input and results I have also analysed data that I have gathered from a questionnaire that was answered by other developers, as well as from interviews with two colleague developers.

Keywords:	CMS, Wordpress, Craft CMS, comparison, web developer
Number of pages:	79
Language:	English
Date of acceptance:	1.6.2017

OPINNÄYTE	
Arcada	
Koulutusohjelma:	Online Media & Art Direction
Tunnistenumero:	15378
Tekijä:	Joel Ström
Työn nimi:	Choosing Craft CMS over Wordpress
Työn ohjaaja (Arcada):	Owen Kelly
Toimeksiantaja:	Yrkeshögskolan Arcada

Tiivistelmä:

Kun on aika valita CMS, on useita asioita joita tulee ajatella. Nykyään on yhä tärkeämpää, että ajatellaan projektia ja valitaan sen mukaan sopiva järjestelmä. Tällä hetkellä markkinoilla on monta erilaista järjestelmää jotka tekevät asiat omalla tavallaan, mutta tässä opinnäytetyössä tulen keskittymään tällä hetkellä suosituimpaan järjestelmään Wordpress:iin, sekä Craft CMS:ään joka on suhteellisen uusi kilpailija. Käyn lyhyesti läpi mikä CMS on ja miksi niitä käytetään ennen kuin vertailen Wordpress:ia ja Craft CMS:sää. Syvennyn järjestelmien käyttöön koodin ja käytännön esimerkkien avulla. Kaikki tämä tehdään ohjelmoijan näkökulmasta. Tullakseni päätökseen olen tehny useita vertailuja näiden kahden järjestelmien välillä. Sen lisäksi kirjoitan omista kokemuksistani, ja analysoin muiden ohjelmoijien mielipiteitä jotka olen kerännyt kyselykaavakkeen ja haastatteluiden avulla.

Avainsanat:	CMS, Wordpress, Craft CMS, vertailu, kehittäjä
Sivumäärä:	79
Kieli:	Englanti
Hyväksymispäivämäärä:	1.6.2017

CONTENTS

1 INTRODUCTION

For this thesis I have decided to compare two very different CMS solutions with each other, Craft CMS and Wordpress. The requirements for a good CMS are vast these days, and the competition on the market is fierce. Given the popularity of Wordpress on the current market, why would someone end up choosing Craft CMS instead? This is one of the questions that I will attempt to answer in this thesis.

This thesis will briefly describe what a CMS is and what they can be used for. I will introduce both systems so that the reader has a better understanding of the subject at hand. After that I will do a short comparison between the selected systems.

I will go deeper into the core functionalities and solutions of each system, and with the help of a few practical examples, my own knowledge and my co-worker's vast experience I will attempt to find out the positive and negative sides of each CMS, and attempt to find a reasoning for choosing Craft CMS over Wordpress.

1.1 Motive for the subject

As an Online Media student and a full-time developer, CMS plays a big part in my daily life. Back before I started working in the field I only knew about Wordpress, because that was the only thing that we got told about. I never really thought much about these systems, they only felt like some necessary step between a website and its content.
As my experience as a developer grew, I realized that a CMS was something much more, and that a different CMS could fit different projects. Not only can a different CMS fit a different project better, a developer should never get stuck on using one solution, whether it be a CMS or a coding language. The pace at which technology advances these days is fast, and being stuck using the same solution is not the best idea. This is especially true for developers who are starting in the field, and for developers who are interested in keeping their skills sharp.

Craft CMS was a system that I got introduced to when I started working at Agency Leroy. It was the first time that I actually got interested in these systems, because I thought Craft CMS was so well made. What still boggles my mind to this day is why so many customers and developers choose to go for Wordpress for their websites and other projects, without ever even thinking about other solutions. This is one of the core reasons for choosing this subject.

1.2 The goal of the thesis and its target audience

The intention of this thesis is to compare Craft CMS and Wordpress. The things that I will research and present are:

- Brief explanation of CMS
- Introduce Craft CMS and Wordpress
- Compare the two and present the results
- Present expert knowledge and opinions about the subject

The goal of this thesis is:

- To get a better understanding of both systems
- Try to prove which CMS is the better one
- Give other developers a reason to try out Craft CMS

The target audience for this thesis is:

- Developers, designers, content editors, customers and students of the field
- Other people that have to do with websites and content management on the web

It is assumed that the reader has basic knowledge of websites or web based applications and the solutions used to create them.

1.3 Limits to the research

This thesis will compare and analyse Craft CMS and Wordpress. It will attempt to give the reader a better understanding of both these systems with the help of comparisons and examples.

This thesis will not attempt to explain fundamental web development practices.

This thesis will not serve as a guide for the selected systems.

The code examples that are shown are not necessarily cases of best practices, but merely code that works well for me personally and for this thesis. Everything was written for demo purposes only.

This thesis will not focus on the editor's or the frontend user's experiences.

The core of the thesis will focus on explaining the differences between the features of both CMS, and will be based heavily on the developer's perspective.

1.4 Definition of terms

Apache is a free open-source web server software.

Backend is the functionality. It is the driving force behind all websites and applications and handles the logic and the processing. Sometimes, the dashboard of a CMS can be called backend, but the meaning differs from the functionality.

CMS (Content management system) is a computer application and a framework that supports creation and management of content on the web.

CSS (Cascading style sheets) is a language that describes the style of an HTML document.

Dashboard is the central location for a CMS. This is where the user logs in and manages the content and settings of the website.

Frontend is simply the visual. It is what you see when you visit a website, the layout, the design, styles and effects. It is what the user interacts with.

HTML (Hyper Text Markup Language) describes the structure of the web using markup. HTML elements are the building blocks of the web. The elements are represented by tags, which browsers do not show but use to render the content of a page.

PHP is a server-side scripting language that can be embedded into HTML. It's primarily designed to be used for web development but can also be used for general programming.

MYSQL is a database management system that is used on the web and runs on a server.

Twig is a template engine for PHP. It is what Craft CMS uses for its templates.

2 METHODOLOGY

This thesis will mostly use an empirical research method, based on the shared experience of the developers at Agency Leroy. This experience and the data that is already available will help me get a good overview of each CMS and how they compare to each other.

A short questionnaire was answered by multiple developers, and two experts from Agency Leroy were interviewed for their own experience as well as for their reasoning for choosing Craft CMS over Wordpress. A part of group who answered the questionnaire were my colleagues from Agency Leroy, while the majority of people who answered it were developers from other places.

A part of the research was done as traditional desk research. Some of the references come from different books and articles, while the technical information for each CMS come from their own official documentation.

Tables will be used to display comparisons between features. Images and other examples will be taken from the official documentations as well as from my own code examples. Some of the logic and the ideas behind these code examples are things that I use in my daily developer life.

The comparisons only focus on the most recent versions of the different CMS.
Specific information about technical details and versions can be found from chapter 4 forward.

3 CONTENT MANAGEMENT SYSTEMS

3.1 Introduction

A content management system, or CMS for short, is a system that lets the user edit the content of a webpage or application without the need to change the code itself.

These systems come in many different kinds and for many different purposes. However, in this thesis I am specifically talking about content management systems designed to edit and create the content of traditional websites.

A CMS provides multiple different tools for the user. Here are some of the basic things that you can expect to find in a simple CMS:

- Create, edit, publish and archive web pages
- Create, edit, publish and archive articles

More advanced systems also offer the user options for advanced user permissions, plugins, possibility to change the look of the site and so on.

3.2 Why use a CMS?

CMS allow the user to do changes to the websites by themselves, when previously they had to call their webmasters, and ask them to do the changes in the code. Not only was this approach clumsy but it was also expensive and very vulnerable to errors.

In her book “The Elements of Content Strategy” Erin Kissane describes the clumsiness of working with the old web back when there were no such things as CMS with the following words:

> You know how websites all used to be made of individual, hard-coded HTML pages? That sucked.
>
> If you weren't around for that part of the web, think of it as the scribes-in-monasteries period of the web history during which all written human knowledge was hand-copied by very pale guys with poor eyesight. The invention of the web has been compared to Gutenberg's introduction to movable type, and for good reason – but on the web content side, our communication revolution really took place when software developers brought the database-and-display-template systems of the old offline computing world onto the web in the form of content management systems.

A CMS is a good choice for your website if it is:

- A website that gets regularly updated with content
- A website that has multiple people working on it

It is also a good choice for you if:

- You do not want to run risks of creating errors in the code when doing changes in the content
- You want to be cost effective

These are just some of the examples of why a CMS system is useful. There are obviously other reasons too, but in the end they are different for every project.

Let us take a look at why you would not want to run a CMS on your website:

- If your budget is very low and you can not afford a CMS powered website to begin with. There are free alternatives, but many solutions and features require expert skills which means that you would have to hire a developer. Also, maintaining a CMS powered website takes more resources compared to a static website.
- Your site does not need to get updated after it has launched. Examples of these kinds of sites are "splash" sites which are sites that are usually used for marketing something, for example a restaurant opening. A splash site can be

something that comes before a real site, and gets taken down after the real site has released.

- You have a homepage, but you are a developer yourself so you do not have to invest in a CMS

3.3 Common misconceptions and issues

For CMS, there are multiple different things that can be seen as misconceptions or potential issues. Many of these things often rise from lack of knowledge. A few common misconceptions and potential issues are:

1. That a CMS is the main ingredient for a website's success. This is untrue as the main purpose of a CMS is to manage and distribute content, but many of the other factors that are important for websites are outside of the realm of CMS.
2. That CMS would eliminate the need for hiring a web developer. This can be true in some cases, for example with site builders that are specifically designed to be easy to use and do not require any programming skills. However, if you are looking to create a website that has any advanced functions, if it hosts important data or is simply a big website, it is highly recommended to hire a web developer.
3. A website needs branding in order to keep the visitors interested. Most CMS have very little style customization, branding or personality.

The following chapters will introduce Wordpress and Craft CMS. Both of these chapters include a few things that can be seen as potential issues for each respective CMS.

4 WORDPRESS

This chapter will focus on Wordpress. After a short introduction on what Wordpress is, I will go through its biggest features, demonstrate it with a few examples and write about some of its issues. I will also briefly discuss about the target audience at the end of the chapter.

4.1 Introduction

Wordpress is a CMS system that was originally built for blogging in mind. Since then it has been developed to a full-fledged content management system. Wordpress is an open source project, and with the help of all its plugins and widgets, it can be something very different depending on the website it is used for.

Wordpress is the most popular CMS on the web. In W3Techs report we see that for every website that uses a CMS, Wordpress is used 58.8% of the time. This makes Wordpress the clear leader in the market. For more in-depth data about the statistics of the usage of Wordpress and other systems visit the w3tech article (Usage statistics and market share of Wordpress for websites).

Not only is Wordpress an open source project meaning that you can use it however you please, it is also free. This is one of the driving factors for its popularity.

At the time of this writing, the newest Wordpress version is 4.7.3. This is the version that is used for this thesis.

In the following sections I introduce the features of Wordpress, and take a look at some of its details. I will also take a short look at the templating and PHP. This will only function has a brief overview and should not be taken as a definite guide for the system.

4.1.1 Features

This section introduces the most important features of Wordpress. A few things have been left out on purpose in order to only keep the relevant things in focus, and to be able to compare them easier with Craft CMS's own features.
The features that do exist on the homepage of Wordpress, but are not highlighted here are "Simplicity", "Publish with ease" and "Multilingual". These are not relevant because Craft CMS has these exact same features. In the case of "Simplicity" this is something that is totally dependant on the project and is a question of taste more than anything else. You can also argue that Wordpress is anything but simple.

The list of different features are as following:

Flexibility: Although Wordpress was originally created for blogging in mind, it can be something much more. With the help of themes and plugins Wordpress can handle multiple different kinds of websites.

User management: Different users, permissions and community features.

Media management: Drag and drop, folders and subfolders for easier asset management. Also includes built-in editing-tools.

Easy theme system: Wordpress comes bundled with multiple different themes. There's a separate theme directory that the user can use to find other themes to install. Besides this, it's also possible to create your own theme.

Extend with plugins: You can argue that plugins are that makes Wordpress what it is. Plugins are what makes Wordpress so flexible.

Built-in comments: A commenting system that also includes tools for a forum or discussions with moderating options.

Importers: A tool designed to import your blog or website software over to Wordpress. The importer works for blogger, LiveJournal, Movable Type, TypePad, Tumblr, RSS, Blogroll, and other Wordpress powered websites.

Freedom: Wordpress has been created with the GPL licence. You are free to use Wordpress however you please.

Community: Community really is a feature for Wordpress. As the most popular CMS in the world, the community around is large and active.

Besides these features, Wordpress has a separate section that highlights developer features. These features are:

Plugin system: The Wordpress API lets a developer create their own plugins. Besides this, Wordpress also has a repository that the developer can host their plugins on.

Theme system: As a developer you have the freedom to create your own themes for clients. You can also host the theme in the theme directory and offer it for free to all users.

Application framework: Wordpress provides multiple features that helps developers create their own application. Some of these features include translations, user management and HTTP requests.

Custom content types: Wordpress comes pre-built with content types, but you're also able to create your own custom post types, taxonomies and metadata.

The latest libraries: Wordpress comes with the latest script libraries for developers. These include jQuery, Plupload, Underscore.js and Backbone.js.

4.1.2 Example Wordpress websites

Examples of many different websites that are built on Wordpress can be found at https://wordpress.org/showcase/

4.2 Setting up Wordpress

In this chapter I will present a brief look on what the Wordpress files and folders look like, as well as a short step by step guide on installing the system. I will also present the dashboard of the system, as well as some example code in the form of a simple news landing page built with PHP.

The Wordpress version used in the examples is 4.7.3, which by the time of this writing is the newest version available.

4.2.1 Folder structure

When you download Wordpress you get all the required files needed by the system. The whole folder contents look like this:

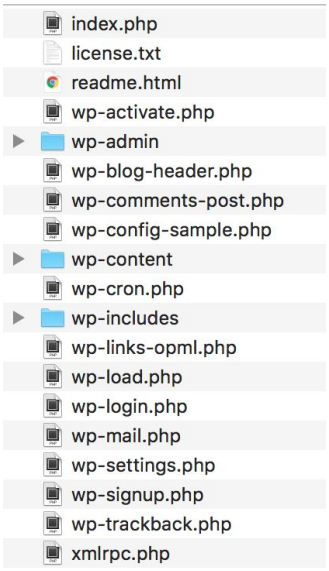

Figure 1. The root of the Wordpress folder

The functionality of Wordpress comes from the files outside of the folders, as well as from the files inside of wp-includes and wp-admin. Wp-content is where your theme resides. This is where you define how your templates work and how your site looks like. Out of all the files in the root of the folder, the wp-config, or the wp-config-sample.php file in this case, is one of the most important files for basic usage. This is where you set up your database credentials. You can read more about the installation and setup phase in the next chapter.

Because both the wp-includes and the wp-admin both include so many files, I will skip on including images of them here. However, these folders are not important for basic usage, as you will never have to go into those folders to change anything.
Let us take a look at how the wp-content folder is structured.

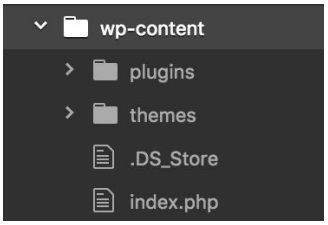

Figure 2. The wp-content folder

This view is divided into two folders, the plugins folder and the themes folder. The plugins folder is obviously where your plugins go, as they are not a directly a part of your theme, it means that plugins can be considered sitewide. The themes folder is where you build your website.

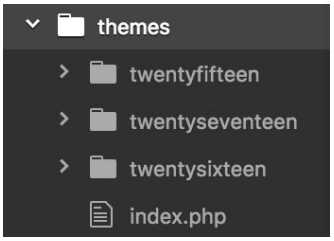

Figure 3. The contents of the themes folder

This view includes all the different themes that you have installed on your system. Only one of them can be active at any one time. Wordpress comes with a few pre-installed themes, and these get updated regularly and new ones get added.

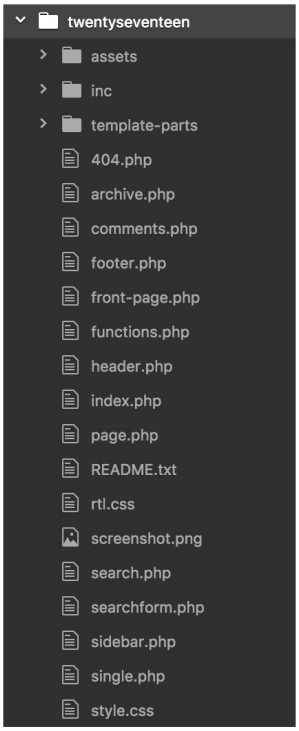

Figure 4. A look inside the twentyseventeen theme folder, which is one of the default Wordpress themes.

In the figure above, you see the building blocks of your website. The assets folder contains things such as your css files, js files, images and other assets. The inc file consists of different functions while the template-parts folder includes your partials such as different elements for the footer, header and navigation. In the root of the theme folder we have the templates that, when stitched together, become the website.

4.2.2 Installation and setup

Now that we are familiar with the structure of the folders, I will demonstrate a step by step guide on how to get a Wordpress project up and running. The following steps will not have to be completed in the exact same order. A similar step by step guide will be presented for Craft (chapter 5.2).

Step 1, creating a database

The first step is to create a new database for our project.
In my example I have created an empty database with the help of a program called Sequel Pro. For this task you can use the database management software of your choice.
Next, we have to tell Wordpress how to connect to our newly created database. As mentioned in chapter 4.2.1, this is done in the wp-config file which can be found in the root of the project.

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'database_name_here');

/** MySQL database username */
define('DB_USER', 'username_here');

/** MySQL database password */
define('DB_PASSWORD', 'password_here');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');
```

Figure 5. A look at the wp-config file.

The figure above is quite self-explanatory. In my case, I have created a database named wordpress_dev, which will be filled in as a string after DB_NAME. Because we created a new database, I will use the default credentials to connect to it. In this case, the defaults are “root” for both the username and the password. Everything else in this view can be left as it is.

Step 2, set-up a server

Like creating a database, you also need to set-up a server for your installation. This can be done in many ways using different programs, and in my case I have used a software called Mamp Pro. I will not go through the step by step procedure of doing this as it is

expected that the reader is familiar with this. The exact same procedure will be used for setting up the Craft CMS project.
The only unique thing for Wordpress is that the server will have to be pointed at the root of the folder.
In this example I have named my server to wordpress.dev, but just like the database name, this could be anything.

As of this writing, Wordpress has the following minimum server requirements:

- PHP 5.2.4
- MySQL 5.0

These versions are old and can expose your site to security risks, which is why Wordpress recommends the following versions:

- PHP 7 or greater
- MySQL 5.6 or greater or MariaDB version 10 or greater
- HTTPS support

For more details on the requirements of Wordpress, visit Wordpress requirements.

Step 3, run the installer

The final step is to go through the installation process. By visiting the domain that I chose for my server, I am able to start this process.
First we choose the language. The chosen language will be used during the installation process and is also used in the dashboard of the system. This can be changed later on.
After this we choose a name for our website and set-up our admin user by choosing a username, a password and an email address.
This step concludes the installation process. For in-depth guide on installing Wordpress, visit the installing Wordpress article

4.2.3 Dashboard overview

In this chapter I will quickly introduce the dashboard of Wordpress. After going through the installation process, which was detailed in the previous chapter, we get to the first view of the dashboard.

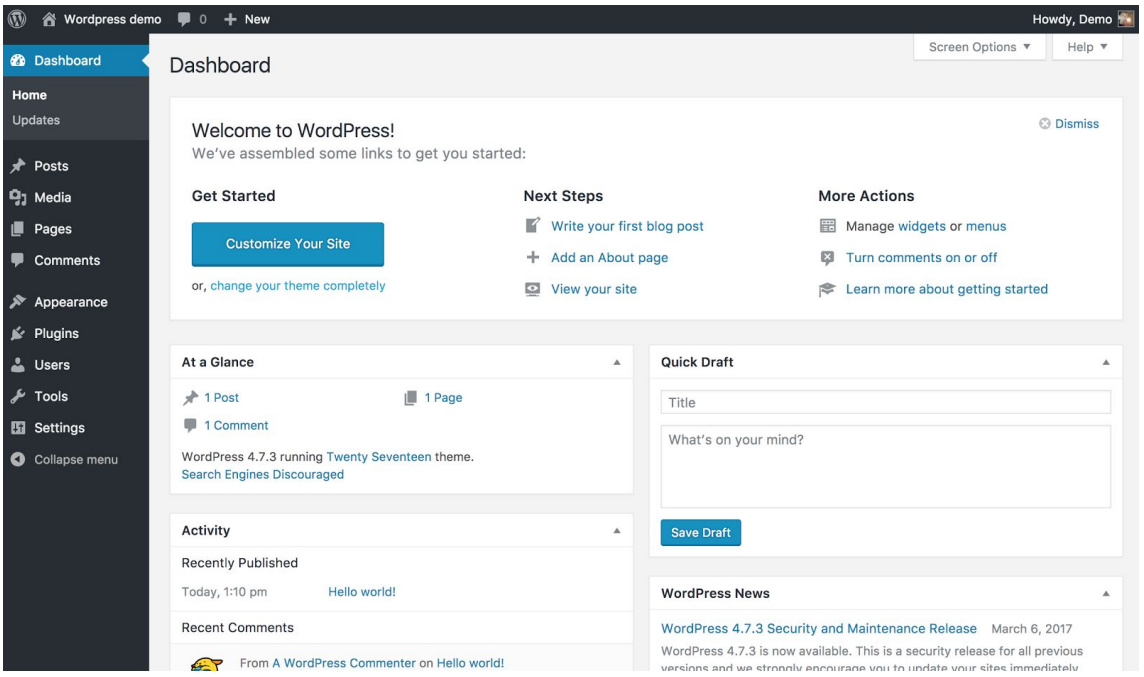

Figure 6. The first view of the Wordpress dashboard

This view is a bit more packed than the Craft CMS equivalent. If you are logged in, the black header follows you not just in the dashboard view, but also on the frontend. This can be used to navigate to the dashboard or to the frontend, has a quick link for creating new posts and so on. It is possible to hide this header altogether.

The navigation can be seen on the left side. This is used for navigating the dashboard.

The default view of the first page gives you some statistics of your website, options to customize your website as well as a news feed. This view is possible to edit to your liking, and many Wordpress plugins use this view to display relevant data.

Starting from top to bottom on the navigation, we have posts as the first link. This is where you can write articles for your typical blog website.

Media is where you upload your assets, and this is also where you can manage your assets as well as edit them as mentioned in the features listing in chapter 4.1.1.

Pages is where your static pages can reside. Contact page, About page and other similar ones are a good example of pages that reside under this navigation link.

Comments, is as the title suggests, a page where you can browse comments that users have posted on your pages. You can clearly see that Wordpress is still heavily influenced by its roots because of core features such as this.

The appearance page is where you can edit simple things related to the visuality of the page. Depending on the installed theme you might be able to change background colors and other similar things from here. This is not a very important feature as a developer.

Plugins is where all of your plugins reside. This is where you install them and manage them whether it be updating or managing their individual settings.

Users is where you can see the registered users on the site. This is also where you manage all the users and edit permissions.

Tools is where you can find a few nice Wordpress features. The best features here are the import and export features, which let you sync data from and to other websites. A list of the different services that are supported were mentioned in chapter 4.1.1.

Settings is where all the general configurations for your website reside. These options range from basic settings such as your website name to detailed permalink configurations and so on.

The posts view, which is one of the most important or most used views in any CMS, is where you see a list of all the posts on your page. By default, you can only filter by date. More options come available with published and published posts, categories and tags to name a few.

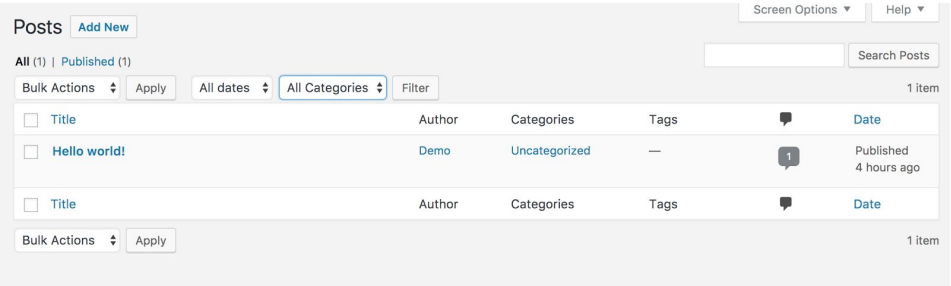

Figure 7. The posts page that lists all the posts on the website

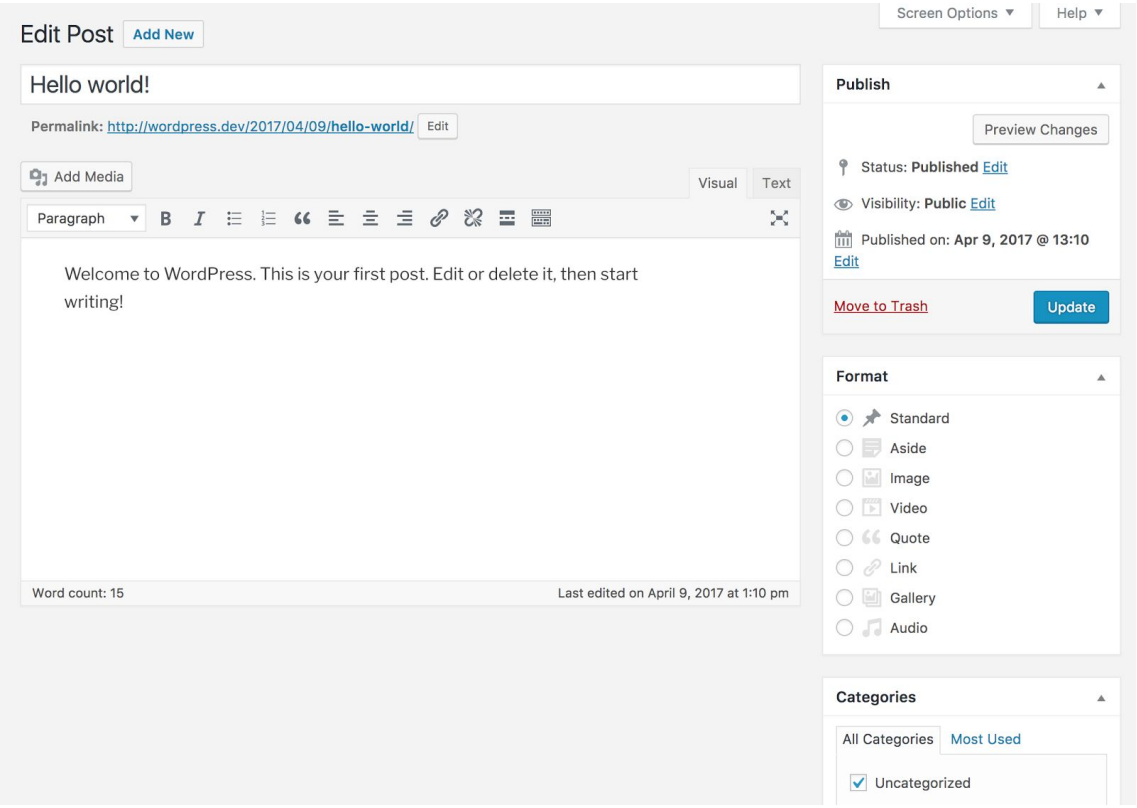

Figure 8. The single post view

Quite simple, as it should be. Here you can edit the title of the page and all of its content in the rich media field. Basic post specific settings can be edited on the right. The "Preview Changes" button in the right corner opens up a new tab that displays you the current version of the post. Unlike in Craft CMS, you will not be able to edit the post and see the changes in real time, but instead you have to manually refresh the page.

The settings view, can be quite intimidating even with a fresh install of Wordpress. This is how the first view looks like:

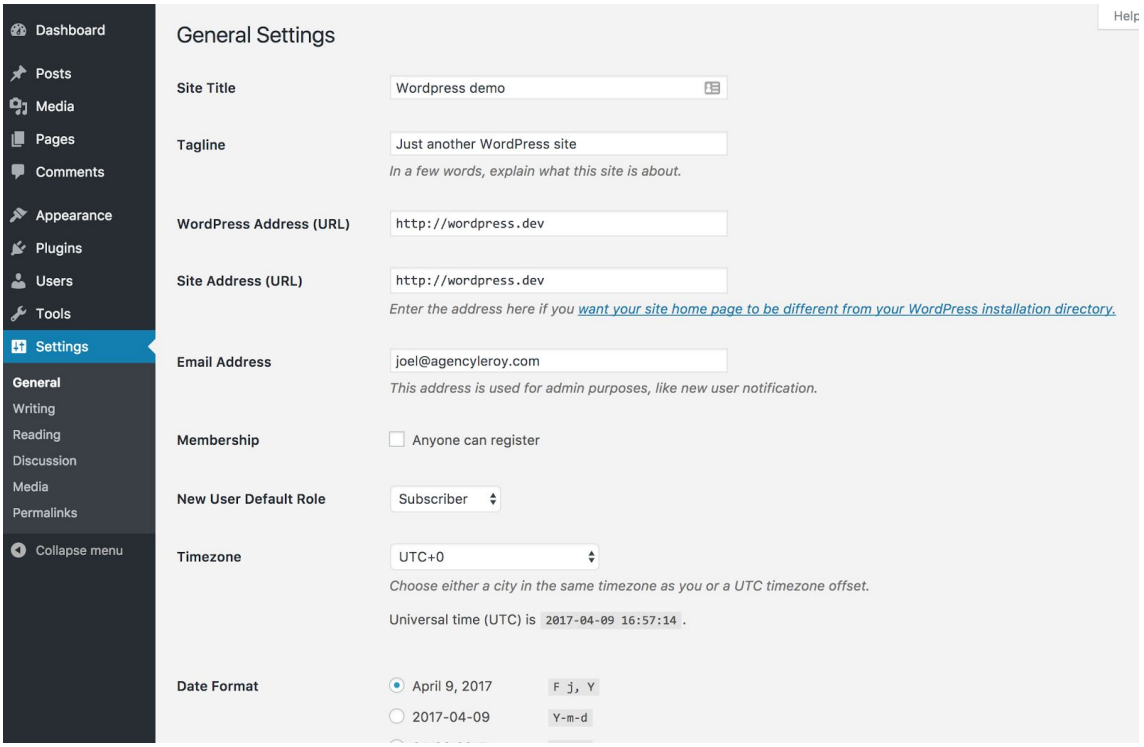

Figure 9. The settings view

The page is still scrollable, so everything is not visible in the figure above. It can sometimes be a chore to find the correct thing you are searching for because of how many items there are in the navigation and because of how many different options there are. A more user friendly way of doing this in my opinion is to do something akin to the solution used by Craft CMS. A screenshot of this can be seen in chapter 5.2.3 which discusses the dashboard of Craft CMS.

4.3 Code section

4.3.1 PHP in Wordpress

PHP is a programming and scripting language. Wordpress is written using PHP, and as the system itself, PHP is also open source.

Most Wordpress users will not have to learn PHP in order to use the system. However, as this thesis is written with a developer in mind, I will go through some of the basics that are needed when working with Wordpress and more specifically its themes.

PHP only processes the content that is inside the <?php and ?> tags. An example of this can be seen in the code snippet below:

```
<?php
  echo "hello world";
?>
```

This statement would simply display “hello world” as a string. In a similar manner we can use variables (and some PHP syntax) to output something like “The boy has a red shirt”:

```
<?php
  $red = "red";
  echo "The boy has a " . $red . " shirt.";
?>
```

What if you have multiple similar elements that you want to output without having to write out every single one of them manually? That is where a loop comes in handy. Perhaps you want to loop through images, effectively creating a gallery, or perhaps you have to loop through the most recent posts on a blog page. This is where the for loop comes into play. A basic while loop could look like this:

```php
<?php
  $number = 1;
  while ( $number <= 10 ) {
    echo "The number is $number";
    $number++
  }
?>
```

We start by opening the PHP tag. The first thing to declare is a variable that we use in our loop, the number variable. Here we set it to 1. The most important part in any while loop is the statement inside the parenthesis. In this case, we want the loop to continue running as long as our number variable is less than or equal to 10.

As long as the statement inside the parenthesis is true, the loop would continue and running whatever is inside it. In this example, the loop would output the string "The number is" followed by the number variable. Before closing the loop we increment our variable, which means that every time the code comes here it adds one and starts from the beginning if the statement is still true.

In this example, we would get "The number is X" echoed until X becomes 10.

Just as with any other CMS, conditional logic is important for creating websites with Wordpress. Many simple examples of these conditionals can be found in any Wordpress theme, and for the sake of having an example I have included a few of them from the twentyseventeen theme.

Perhaps we want to see if our current template has any posts in it:

```php
<?php
if ( have_posts() ) :
  /* Start the Loop */
  while ( have_posts() ) : the_post();
    /* Single post */
  endwhile;
else :
  /* Empty */
endif; ?>
```

With the help of the following snippet we can check to see if the navigation named "social" exists:

```
<?php
  if ( has_nav_menu( 'social' ) ) : ?>
    /* Output social navigation here */
  endif;
?>
```

4.3.2 Wordpress functions and the functions.php file

There are multiple functions that are readily available for use in Wordpress. Many of these are especially useful for creating and editing template files.

These functions are not the same as the functions.php file which resides inside the theme folder. This file works essentially in the same way as a plugin, in the sense that you have the option to include different kinds of functions there in order to add features to your website. The biggest difference between the functions.php file and a plugin is that the functions.php file only affects your theme, whereas a plugin affects your website and all of its themes.

For the full list of functions visit the function reference article.
For a full explanation of the functions.php file visit the functions file explained article.

4.3.3 Example news page

The default Wordpress themes come with a pre-built loop that is designed for some form of news. This can be used for blogging or any other articles and can be easily edited to your liking. The default code that is used in the twentyseventeen theme looks like this:

```php
<?php
if ( have_posts() ) :
  while ( have_posts() ) : the_post();
    get_template_part( 'template-parts/post/content', get_post_format() );
  endwhile;
  the_posts_pagination();
else :
  get_template_part( 'template-parts/post/content', 'none' );
endif;
?>
```

The code above checks to see if the page has any posts in it. If it does have posts, it goes ahead and gets the correct template for the post. After it has looped through all of the posts it includes a pagination which divides the posts up to multiple pages. If the page does not have any posts it passes an empty template part showing no posts.

In order to demonstrate how a basic news page could look like with Wordpress I went ahead and created a few dummy posts in the "Posts" section of the Wordpress dashboard. These are then looped through and shown on the index page.

The example landing page with the posts, as well as the default Wordpress sidebar on the right side looks like this:

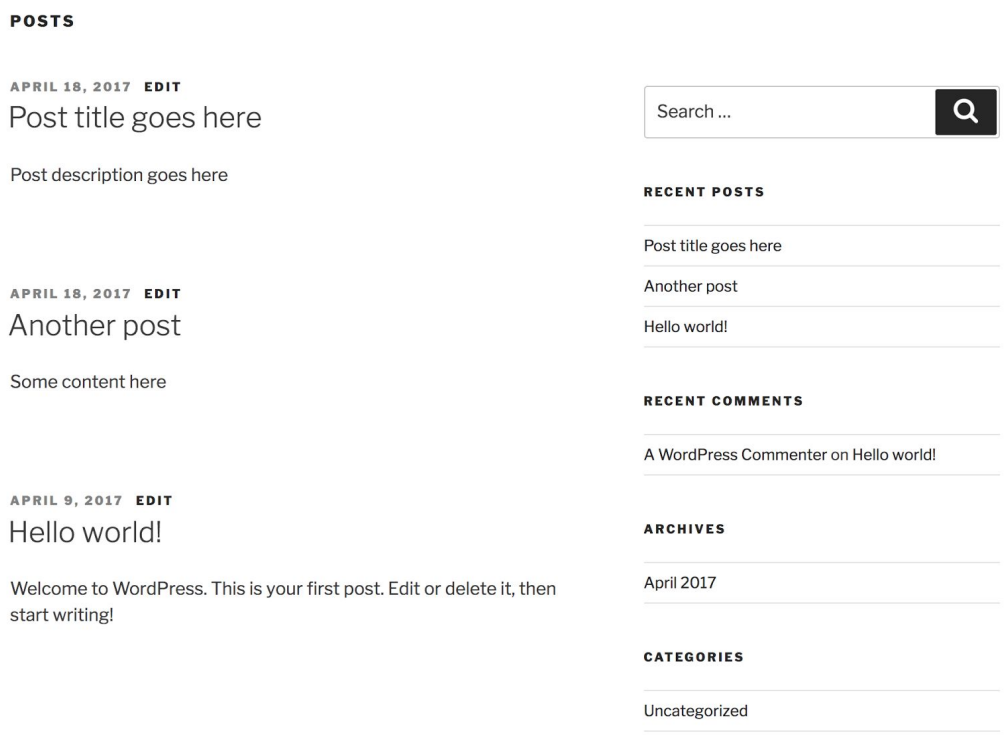

Figure 10. The default look of a page looping through posts

This short demo and showcase of Wordpress basics concludes the chapter. Everything shown here was created for demo purposes only and many of the things were simplified in order to better explain the major steps of setting up a website with the help of Wordpress. Most of the code snippets were taken from the twentyseventeen theme, and the same basic concepts are still used in every other theme.

4.4 Issues

In this section I will come up with three things that I qualify as potential issues or drawbacks to the system. Every system has tons of smaller issues, but in this case I will focus on bigger problems, things that a new user of the system should know of.

A comparison between the issues of Wordpress and the issues I will highlight in the upcoming Craft CMS section will not be made. Instead it is up to the reader to decide if these issues are a deciding factor between choosing one CMS over the other.

Issue 1: Dependant on plugins. If you want to do anything except for a simple blog with very basic functionality you will have to install multiple plugins on the site. This is an issue because many plugins do not work well together, if at all. Updating them can also be risky, and security issues in plugins are usually the reason why Wordpress sites get hacked so often.
Unless you are dealing with the most popular plugins, you might also face problems with lacking documentation or support.

Issue 2: Lack of important core features. This issue has partly to do with the issue mentioned above since it is also about plugins.
In my opinion, Wordpress is lacking many features that could be a part of the core package, instead of being available as different plugins. Some of these things that I am thinking about are localization and custom fields.

Issue 3: Security. Yet another thing that has something to do with plugins. Wordpress sites are infamous for being hacked often.
With the massive amount of plugins and other components available for Wordpress, it is not a wonder that there are a lot of security issues with the system.
In a 2016 Q1 report made by Sucuri, we can see that the plugins RevSlider, Gravity Forms and TimThumb account for 25% of the compromised Wordpress sites. The whole report which includes a lot of stats of not only Wordpress but other CMS as well, visit https://sucuri.net/website-security/website-hacked-report (brought 25.3.2017)

4.5 Target audience

Keeping in mind all the things that I have brought up in the previous sections of this chapter, I think that Wordpress is a good CMS for a user that wants to get a website up and running fast without having to think too much about technicalities. This is especially true if you are creating simple websites such as blogs, as Wordpress is

basically a blogging platform. It is also a good idea to use Wordpress if you are interested in selling custom themes to your customers.

I would avoid Wordpress if you are building a large website that needs to be highly customizable. Another very good reason to avoid using Wordpress is if you are unsure of the security of your website. If you are instead interested in building by hand and coming up with your own solutions, look no further than the next chapter where I introduce Craft CMS.

5 Craft CMS

The Craft CMS chapter will function as a brief introduction to the system. I will highlight its features, introduce Twig which is used as the templating language, and present a few examples. At the end of the chapter I will go discuss some of its issues, as well as the potential target audience.

5.1 Introduction

Developed by Pixel & Tonic, Craft CMS is a Wordpress alternative for development-oriented publishers. Unlike Wordpress with its themes, Craft CMS does not offer ready made solutions. It is also not a site builder such as Wix or Squarespace. Instead Craft CMS is built to be extremely scalable and offers multiple complex content management features natively.

At the time of this writing, the newest Craft CMS version is 2.6.2961. This is the version that is used for this thesis.

The upcoming sections will dive into the features and technicalities of Craft CMS.

5.1.1 Features

This section introduces the things that Pixel & Tonic market as features for their product. A few things have been left out on purpose in order to keep only the relevant things and to be able to compare them easier with Wordpress's own features.
The features that are not highlighted underneath are "One-click updating" and "Responsive CP". These are not relevant because Wordpress has these exact same features.

The list of different features are as following:

Matrix: The Matrix field is a special field that is designed to store repeatable data, or string multiple different types of data together. A Matrix field is made out of blocks, and each one of these blocks can include multiple other custom fields.

Live preview: Gives you the option to see how your page looks like before saving or publishing it. Your changes on the dashboard automatically refresh the live preview allowing you to manage your content and see it update in real time. This view is also shareable.

Localization: Craft Pro offers full localization for your content. What this means is that you can change your content depending on the user's language or territory.

Custom fields: Craft comes with 18 different fields that you can customize to your liking. These fields range from basic text fields to asset fields and multi-option fields.

Section types: Craft comes with three different section types for easier content management. These are a channel-type, structure-type and a single-type. A channel section is usually used for a news section, while the structure section can be used for your basic child pages and a single section for your homepage.

Entry types: Different sections can store different type of data which are defined by their entry types. Different entry types can have different fields attached to them, giving you the option to create multiple different page layouts with varying content.

Assets: Asset management with folders, subfolders, option to upload to different kinds of services such as a cloud service. An additional feature called "Image transformations" which help you keep your image sizes in check. These transformations can be used to rescale resolution, change quality or change how an image is cropped.

Users: User management with a permission system and a public registration system.

Categories and tags: For creating taxonomies

Relations: A relationship engine with four different relational fields that let you link entries, assets, categories, tags and users together.

Templating: Twig powered templating for customizing your templates and more.

Dashboard: A customizable dashboard for each user type. The dashboard can include things such as recent entries, quick link to post a new entry, or check the status of possible updates to the system or its plugins.

5.1.2 Example Craft CMS websites

For some examples of websites that are powered by Craft CMS visit the showcase article on the official homepage.

5.2 Setting up Craft CMS

In this section I will present a brief look on what the Craft CMS files and folders look like, as well as a short step by step guide on installing the system. I will also showcase the dashboard of the system and example code of a news landing page.

5.2.1 Folder structure

When you first download Craft CMS, you get a folder that includes the following two subfolders:

Figure 11. The two subfolders that reside inside the main Craft folder.

In my example, I have renamed the parent folder to “demo”. This is project specific and could be anything. When comparing to folder structure of Wordpress (see chapter 4.2.1) you can see a big difference in the structure. Craft CMS tries to keep things more organized with dividing everything into two folders.

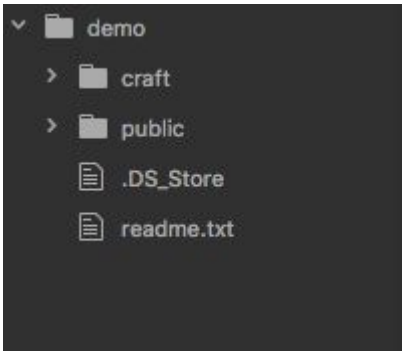

Figure 12. Example folder structure

The craft folder, is where all the functionality comes from.

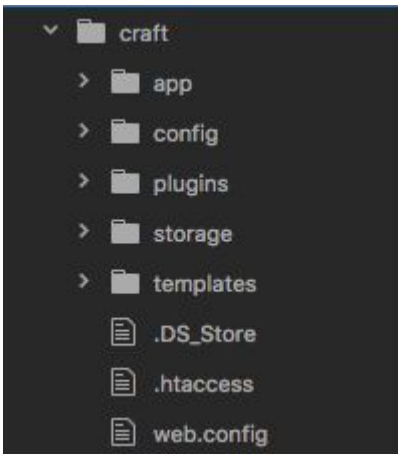

Figure 13. Contents of the "craft" folder

In this view, you can see five different subfolders. The app folder is the engine for Craft, and is a folder that should never have to be opened. This is the core folder for the CMS and is the one folder that should be replaced in case you would have to update it system manually.

The config folder is where different configurations for the system reside. Here you can find database credentials, license keys, general settings, a Redactor configuration file (rich text editor settings) among other things. A few of these files will get highlighted in the next section where I will go through the initial setup of the project.

The third folder, plugins, is where all of the different plugins that are used on the site go. By default, this folder does not include any plugins or other folders.
Up next is storage which functions as a folder for backups, runtime assets such as cache, logs and temporary files. This is also a folder that you should never have to touch. The most common reason for delving into this folder is to read logs when encountering bugs with the system, but this can also be done from the dashboard.

The templates folder is where the layout of the site comes from. This is where you create your custom templates and the structure of the website. This is an important

folder as a lot of your time goes into working on different templates and functionalities inside them. The templates folder looks like this:

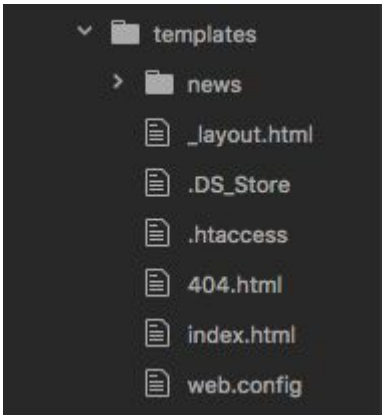

Figure 14. The templates folder

This is the default view of the folder, and comes with a few examples. There is example code for a single news article page, an error page in the form of 404.html as well as your typical index.html page. Depending on your way of working, all of these files and folders could be replaced or removed.

Jump one step backward and we are back at the root of the project. Let us take a look the the public folder:

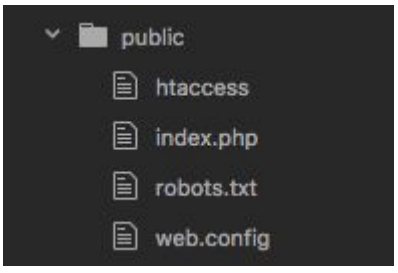

Figure 15. The public folder

The only file that is actually needed here is index.php which serves as an entry point for the website. The file points to the craft folder, and by default this is set one level above its own.

5.2.2 Installation and setup

Now that we are familiar with the folder structure of Craft CMS, let us take a look at how we get a project up and running. The following steps can be done in a different order.

Step 1, create a database:

After downloading the system, the first thing we should do is creating a new database. In my example I have created an empty database with the help of a program called Sequel Pro. For this task you can use the database management software of your choice. Next, we have to tell Craft how to connect to our newly created database. This is done in craft/config/general.php.

```php
<?php

/**
 * Database Configuration
 *
 * All of your system's database configuration settings go in here.
 * You can see a list of the default settings in craft/app/etc/config/defaults/db.php
 */

return array(

  // The database server name or IP address. Usually this is 'localhost' or '127.0.0.1'.
  'server' => 'localhost',

  // The name of the database to select.
  'database' => 'demo_craft',

  // The database username to connect with.
  'user' => 'root',

  // The database password to connect with.
  'password' => 'root',

  // The prefix to use when naming tables. This can be no more than 5 characters.
  'tablePrefix' => 'craft',

);
```

Figure 16. The general.php file which is used to connect to the database

The database name, “demo_craft”, is a name that I came up with for this demo and it could instead be anything else. As we created a new fresh database with no content in it, the credentials to connect to it are the defaults “root”.
The “tablePrefix” variable does not have to be changed, and keeping it as “craft” or something similar, makes it easier to find data relevant to the system.

Step 2, set-up a server:

The set-up of a server works in the exact same way as described in chapter 4.2.2 where I went through the Wordpress steps. The only important thing that you have to remember when setting up the server for Craft CMS is that unlike with Wordpress, the project has to be pointing to the public folder.
In the demo example, I have used a domain name called demo.craft.dev to connect to my website.

As of this writing, Craft CMS has the following server requirements:

- PHP 5.3.0 or later with safe mode disabled
- MySQL 5.1.0 or later, with the InnoDB storage engine installed
- A web server (Apache, Nginx, IIS)
- A minimum of 32MB of memory allocated to PHP
- A minimum of 20MB of free disk space
- A minimum of 1MB of database space

For full details of all the requirements, visit the official requirements article.

Step 3, set permissions:

At a minimum, Craft CMS has to be able to write to three folders:

- craft/app/
- craft/config/
- craft/storage/

The exact permissions are dependant on the relation between the system user that the PHP is running on, and whoever owns the actual files and folders. For specific permission settings see the documentation on installing.

Step 4, run the installer:

Navigate to the address that you created when you set up the server, which was demo.craft.dev in this demo example. If you see the background with the monkey you have managed to set-up the project correctly and are ready to go through the initial settings of the website.

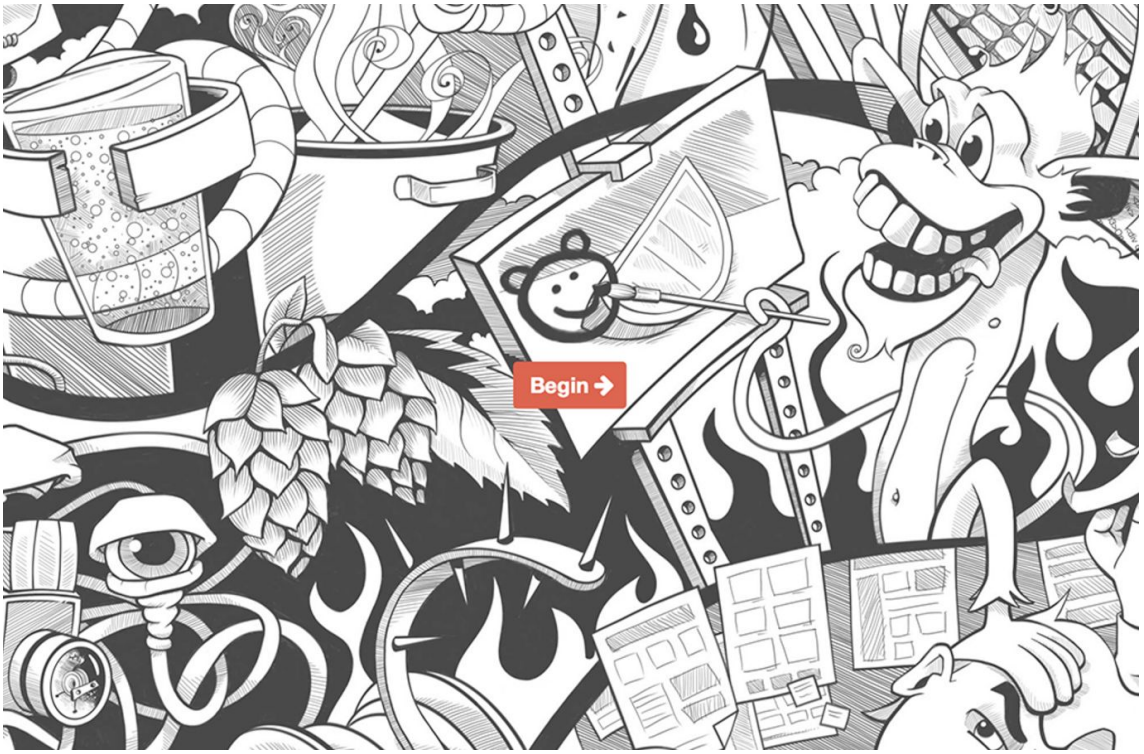

Figure 17. If you see this after going through the installation steps, you have succeeded.

The next steps take you through user creation and setting basic website settings such as the name and the address.

For additional installation resources, visit [the installing article].

5.2.3 Dashboard overview

This section will show how the dashboard, or backend, looks like for a typical new install of Craft CMS. I will only showcase a few selected parts to avoid too many images and less important things.

After going through the installation steps highlighted in the previous chapter, you will enter the dashboard of the system.

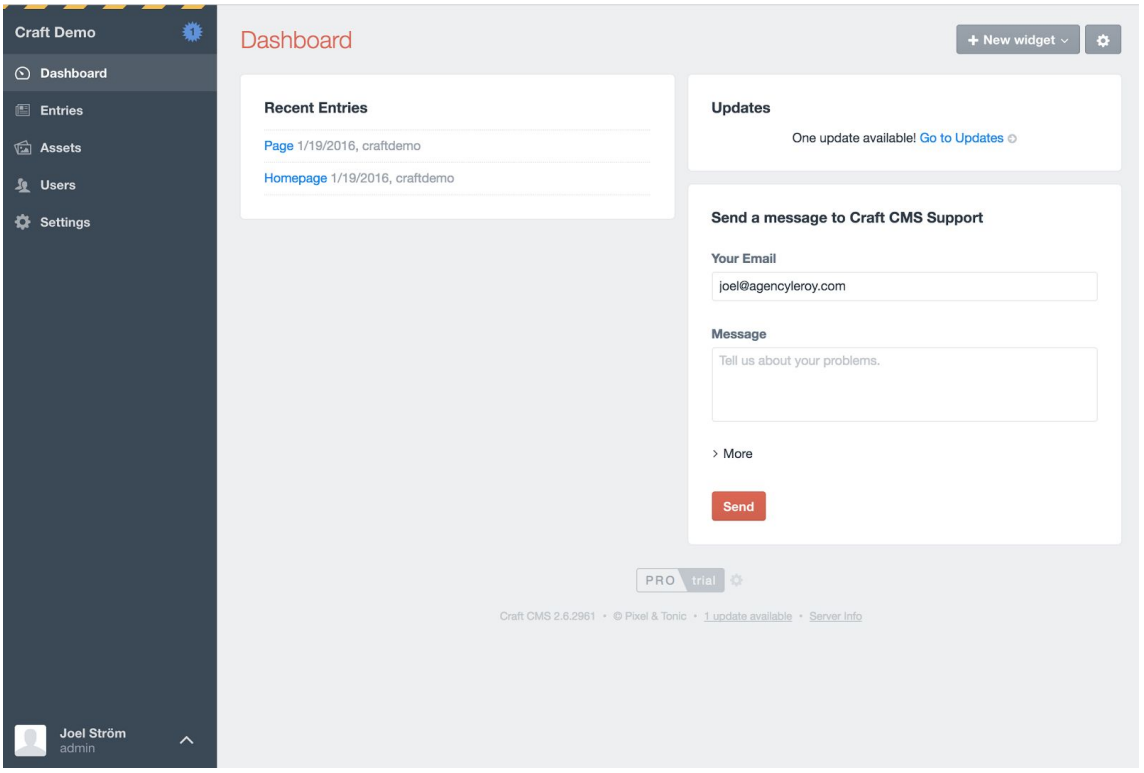

Figure 18. The first view when you enter the dashboard of Craft CMS

As you can see in figure 18, the first view when you log in to Craft CMS is quite plain. The first default view comes with a box for recent entries, updates and a quick portal for sending a message to Craft support. As I highlighted in the features section of Craft, it is

possible to edit this dashboard view and customize it with different widgets. Without any customization, this view does not offer much functionality, and usually serves the user as a landing page.

As with the newer versions of Wordpress, a navigation can be seen on the left side of the view.

The most important parts for an editor is the entries section as well as the assets section. This is where most of the content will be edited and maintained.

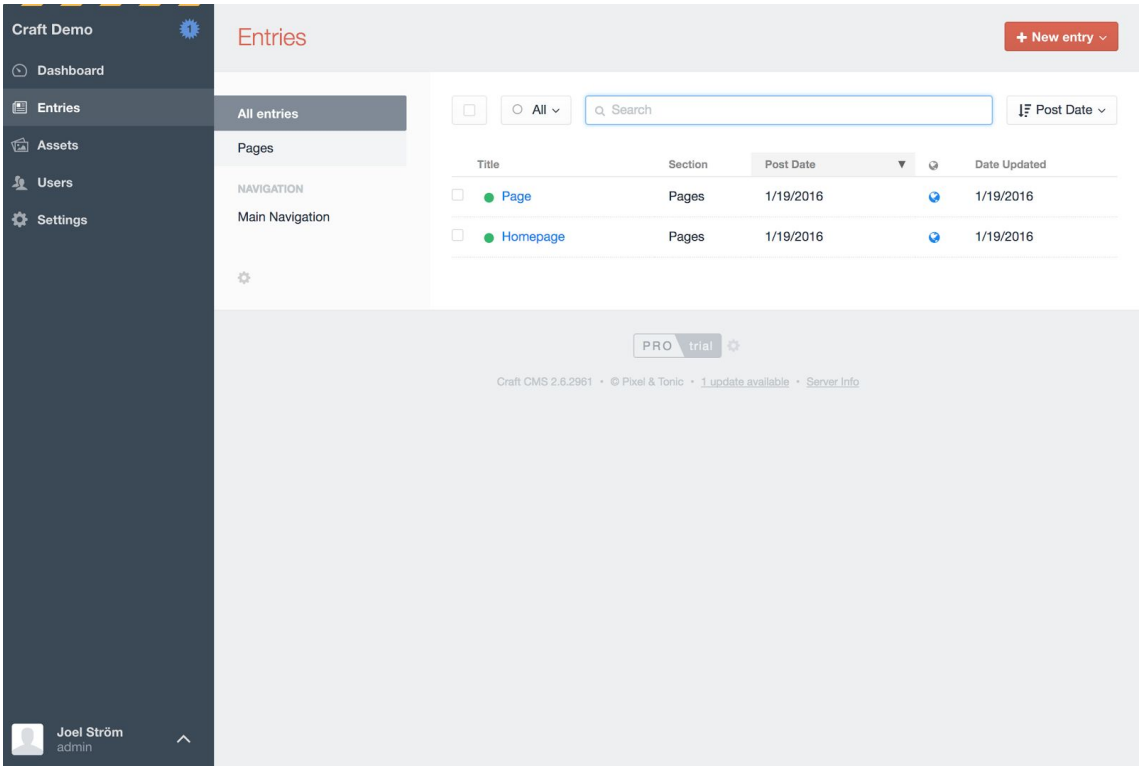

Figure 19. The entries view in the Craft CMS dashboard.

All of the pages, or whatever you choose to call your entries, will be maintained here. Basic filtering is available in this view, offering the user the option to search for a specific entry, or order by different things such as time created, time updated or ordered by name. In this example, by clicking on the “page” link, we get to the single entry view as can be seen in figure 20.

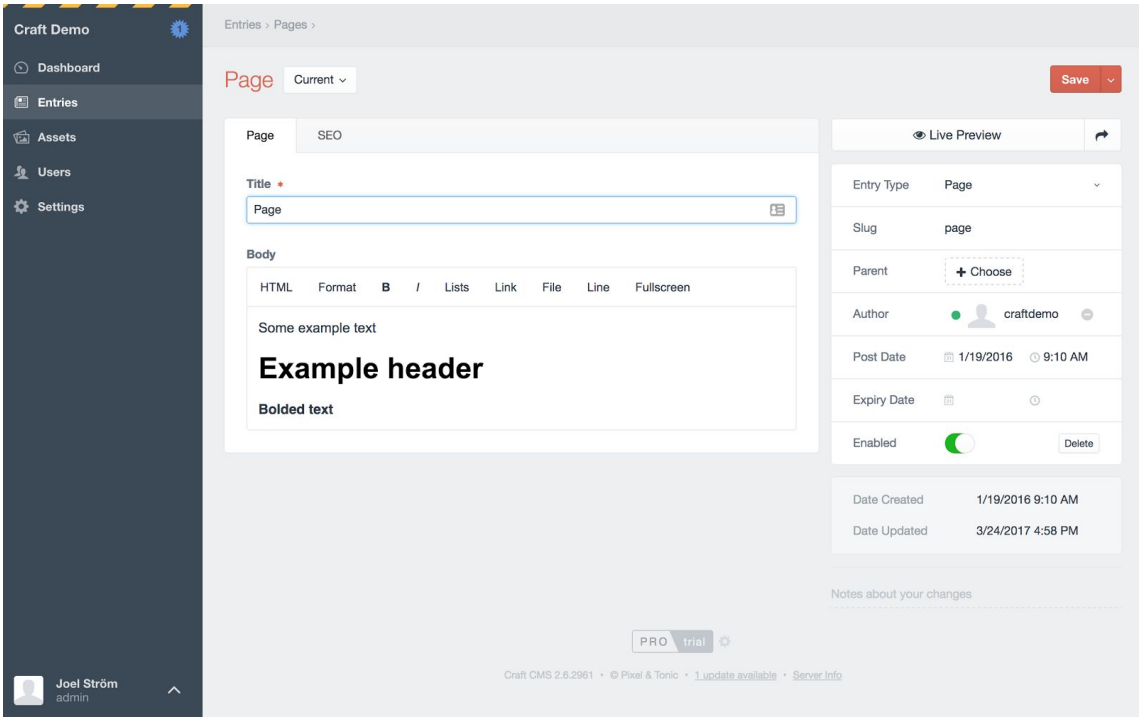

Figure 20. The single entry view in the Craft CMS dashboard.

This is where the editor changes the content of an entry. In the example above, you can see that this specific entry has a “Title” field that functions as the title for entry, as well as a “Body” field that functions as the main content field for the entry.

On the right side you can see the button for saving the entry and underneath it one of the features of Craft CMS, the live preview button. By clicking this button, the user will be able to see the current entry on the frontend, while still being able to edit the content and seeing the changes happen in real time.

Under these buttons the user can see important information related to the entry, such as which entry type the entry uses, the slug, possible parent entry and different options for the editor.

For a developer, especially during the development of the site, most of the time will be spent in the settings section. This is where you create fields, entry types and change other settings.

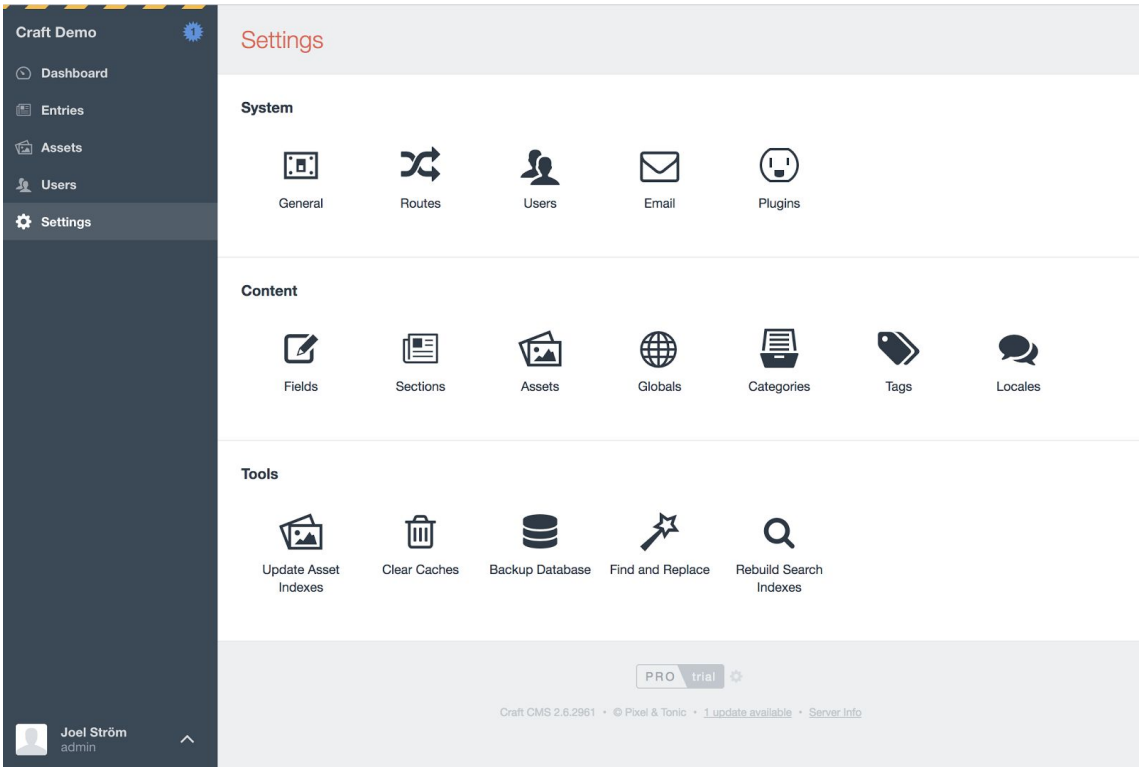

Figure 21. The settings view in the Craft CMS dashboard.

The settings view features all the important settings that the user can work with. As you can see on the figure above, this view uses user friendly icons to display each part. This is a very useful small feature that makes developing the backend easier, since it is faster to find the right places without having to navigate through longer lists of items.

5.3 Code section

5.3.1 Twig

Craft CMS uses Twig as its templating language. Twig is a templating engine for PHP, with syntax that originates from Jinja and Django templates.

Unlike PHP which can be clumsy as a templating language at times, Twig tries to keep things simple with good readability as well as multiple functions that make things easier

for the developer. In this section I will demonstrate some of the ways that you can use to display your data in a Craft CMS template.

To display content in your templates, twig uses double curly brackets.

```
{{siteName}}
```

Displays the name of the website. (The name of the website can be set from the dashboard or hard coded into Craft's general settings.)

```
{{"This is some text"}}
```

Displays a text string

```
{{"This is some text that can be translated"|t}}
```

Displays a text string that can be translated.

There are multiple occasions where you would want to loop through repeatable content. Perhaps you have to loop through images, effectively creating a gallery, or perhaps you have to loop through the most recent posts on a blog page. This is where the for loop comes into play. Let us take a look at few examples of how to do a simple loop with the help of Twig.

```
{% set articles = craft.entries.section("articles") %}
{% for article in articles %}
  <div class="article">
    <h2 class="article__header">{{article.title}}</h2>
    <img src="{{article.image.first.url}}" class="article__image">
    <p class="article__description">{{article.description}}</p>
  </div>
{% endfor %}
```

Let us take a look at the loop a bit more closely.
First, we define a "articles" variable that we set to the Craft section that we want to pull our entries from. In this case, we've named the section "articles".

We then start the loop by using the “article” variable. This could be anything, but for best practice it is a good idea to keep the variable names logically correct. An alternative could be “single” or “single-article”.
In order to keep our loop structured and nice looking on the front-end, we write some plain HTML in the form of a div element that wraps the whole article. In each of these wrapping divs we output whatever content we have defined and filled in from the dashboard. In this example we have an articles section that contains single article entries, which contain a simple title field, an asset field for images as well as a text field for a longer paragraph. We output all of these by using the “article” variable that we defined at the start of the loop.

Conditional content is something that is relevant for any CMS. What conditional means in this context is that perhaps we want to show a “Log in” text if the user is logged out, but change the text to “Logged in” when the user is logged in. Or perhaps we just want to make sure that an image field has been filled in a single article post, otherwise we can choose to hide the whole section that contains it. In order to get a better understanding of this, I will demonstrate a few simple ways of checking for conditional content.

```
{# Check if the current user is an admin, if this resolves to true, display a link to edit the current entry #}
{% if currentUser.admin %}
  {{ entry.getCpEditUrl() }}
{% endif %}

{# Check if the featuredImage field in the entry is filled, if this resolves to true, fill the src attribute of the img tag with the url to the image #}
{% if entry.featuredImage|length %}
  <img src="{{entry.featuredImage.first.url}}">
{% endif %}
```

This can even be shortened to exclude the length filter. You can also check to see if the element even exists on the current page by using the “is defined” check. This is useful if you are using the same code snippets in many different places, or if you are developing

a website with multiple different environments that might have different code or functionality.

```
{# Checks if the featuredImage field even exists. Checking for only for length will result in a template error if the field is not correctly defined in the backend #}
{% if entry.featuredImage is defined %}
  <img src="{{entry.featuredImage.first.url}}">
{% endif %}
```

You can output different types of content with each template, and this is made easier with the possibility to check for a specific entry type:

```
{% if entry.type == "newsLanding" %}
  {# Output some content that is relevant for a news landing page #}
{% else %}
  {# Output something that is not relevant for a news landing page #}
{% endif %}
```

Comparison operators, ternary operators and other operators that we know from PHP are also available in Twig:

```
{% set variable = (x is defined) ? x : y %}
```

Some of the examples were brought from the Twig quick start guide article (brought 17.3.2017)

5.3.2 Example news page

To get a better understanding of how Craft CMS templates are built, I have written a small example to demonstrate how a news landing page could look like.
In the dashboard of Craft CMS, I created a new entry type under the “Pages” section called “news landing”. This new entry type finds its template under craft/templates/pages where I created a twig file called “newsLanding.twig”. In order

for the entry type to find the correct template in the files, they should both have the same names.

The news landing page code looks like this:

```
{% block content %}

  {# Fetch all the entries in the news section and loop through them #}
  {% set news = craft.entries.section("news") %}

  {% for newsArticle in news %}

    {# Here we output the content of a single news article #}
    <div class="newsArticle">
      <h1 class="newsArticle__title">{{newsArticle.title}}</h1>
      {% if newsArticle.body %}
        <div class="newsArticle__body">{{newsArticle.body}}</div>
      {% endif %}
    </div>

  {% endfor %}

{% endblock %}
```

As we can see, the code is still relatively readable, even with the added HTML tags. A few comments were added to better explain which step does what.

A whole new section was created for the single news. Unlike the pages section where we can have entries such as the news landing page, a contact page and other similar pages which only get created once and rarely edited, the news section is for all the single news. These get added and edited often, and it is also a good idea to have them in a separate section in order to be able to filter and order them easily. In my example, the news section is called “news”, while the entries that reside in it are called “news article”.

The single news articles that the news landing page is looping through looks like this:

```
{% block content %}
  <article>
    <h1>{{entry.title}}</h1>
    <p>Posted on {{entry.postDate.format('F d, Y')}}</p>
    {{entry.body}}
  </article>
{% endblock %}
```

In the Craft CMS demo that I set up, the code above with the default styling looks like this:

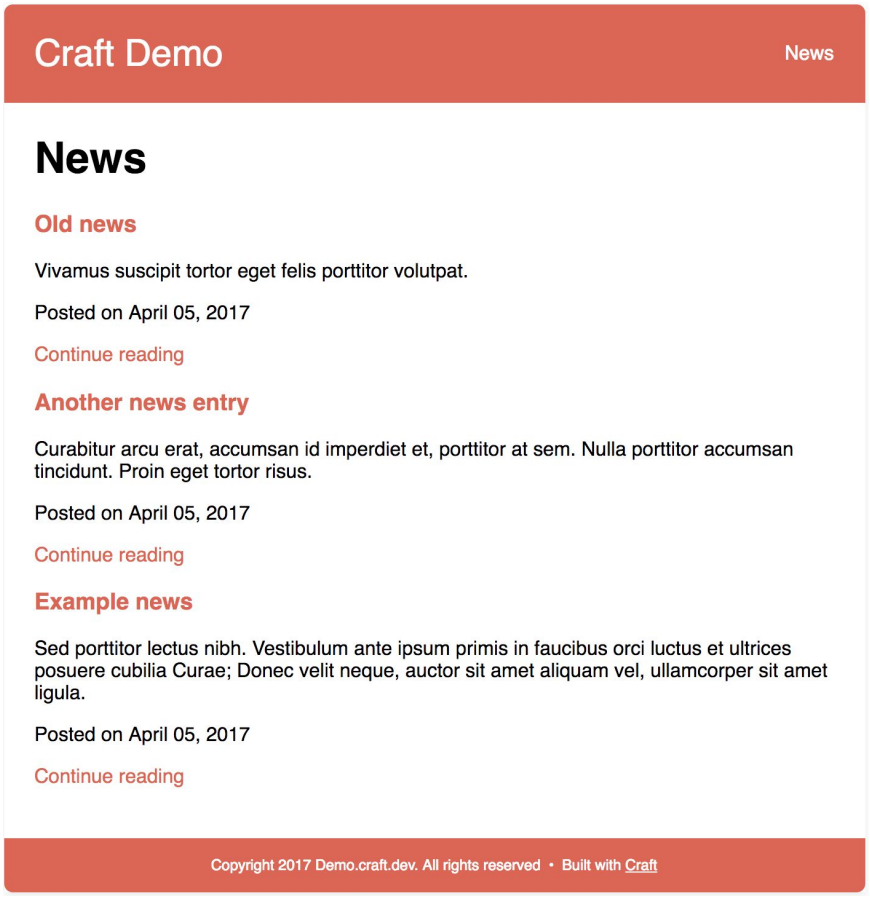

Figure 22. A news landing template looping through single news.

This short demo and showcase of the templating used in Craft CMS concludes the chapter. Everything shown here was created for demo purposes and many of the things

were simplified in order to better explain the major steps of setting up a website powered by Craft CMS.
Even though the example code looks simple, Craft CMS still follows the same simple style even on bigger websites.

5.4 Issues

In this section I will write about three things that I qualify as potential issues or drawbacks to the system. Every system has tons of smaller issues, but in this case I will focus on bigger problems, things that a new user of the system should know of.
A comparison between the issues of Craft CMS and the issues I wrote about back in the Wordpress section will not be made. Instead it is up to the reader to decide if these issues are a deciding factor between choosing one CMS over the other.

Issue 1: The pricing, a potential deal breaker for many users. A basic version of the CMS is free, but that version only includes one admin account and is mainly meant for developers who develop their site for their own use. However, as of this writing, the pricing for a Client version is 199$, which is a version which includes one admin account and one client account (an account that the client uses to edit the website). The Pro version, or the unlimited version, is 299$ which includes unlimited amount of users. This is a very relevant issue especially when thinking about many other CMS which are usually free or cheaper.

Issue 2: Small community. Because of the smaller community in relation to the Wordpress community, there are not that many plugins and other components available for Craft CMS. This also means that there are not as many discussions going on or help available.
Note: A small community can also be a positive thing, because this means that it is usually easier to find relevant information and the platform is not getting targeted by hacking attempts as often.

Issue 3: Twig. Some feel Twig has a steep learning curve, while some feel that some of the techniques used in the templating can also be overly simplistic and overly repetitive. Some of the logic and solutions can questionable. More about this issue can be read in the analysis section.

5.5 Target audience

Based on the previous sections in this chapter and on my own opinions I think it is safe to say that Craft CMS is a fitting CMS for a developer who does not fear to get his hands dirty. As one of the questionnaire respondents so cleverly commented, it is almost assumed that the user is a developer. This is an issue if we are comparing Wordpress and Craft CMS on a general level, but as this is focusing more on the developer's perspective I decided to leave this as a note in this conclusion, rather than to add it to the issues above.
If you are the type of developer that likes to create things by himself without relying too much on pre-defined tools and solutions such as themes and website builders, I would recommend trying out Craft CMS.

I would avoid Craft CMS if you are looking for a fast way to create websites. Creating a single theme for Wordpress and selling that to your customers can be very effective, and with small modifications to the theme you can easily get websites that look and feel different. I would also avoid Craft CMS if you are not prepared to design and build your own solutions.
Because of the licence fee, it might be a good idea to avoid using Craft CMS for small projects.

6 RESEARCH FINDINGS

6.1 Introduction

The following part will showcase the differences in the features of Wordpress and Craft CMS. All of the relevant features were introduced in each CMS's own chapters.

6.2 Feature comparison

In order to get a better view of the differences between the features of the each CMS, I decided to use a simple table. Each category either exists or it does not. The categories are in no particular order.

Disclaimer: Plugins and other addons that add functionality or extra features were not considered in this comparison. Instead, I decided to focus on the core features of the systems.

Table 1. A feature comparison between the two systems

Feature	**Wordpress**	**Craft CMS**
Users	Yes	Yes
Taxonomies	Yes	Yes
Relations	No	Yes
Asset management	Yes	Yes
Image editing	Yes	No
Image transforms	No	Yes
Live preview	No *	Yes
Localization	No **	Yes
Custom fields	No	Yes
Custom sections	No	Yes
Custom post types	Yes	Yes

Themes	Yes	No
Plugins	Yes	Yes
Comments	Yes	No
Importers	Yes	No

* Technically, Wordpress does feature a live preview mode, but you can not work with it in the same way as you can with the Craft CMS live preview which auto updates all of the changes. The Wordpress live preview feature is still practical to check how a page looks like before saving or publishing it.

** Localization in this case means that the content on a website can be different depending on the language and/or territory. String translations and translations of static things in the code is not localization.

As we can see from the table above we have 15 different features listed. Here are the results of the comparison:

Wordpress: 9/15

Craft CMS: 11/15

With a slight lead, a default installation of Craft CMS manages to offer more to the user compared to Wordpress.

7 ANALYSIS SECTION

7.1 Questionnaire results and analysis

The short questionnaire that was made for this thesis was answered by 21 developers. All of them have experience from both Wordpress and Craft CMS. The questionnaire was divided into three sections, the first one being about CMS in general, the second one focusing on Wordpress and the third section focusing on Craft CMS. It included 32 different questions and statements about the subject.
The questions were thought of together with the developers from Agency Leroy. A total of two different drafts were made before the final batch of questions. The questions were things that everyone involved agreed were relevant and important for the subject at hand. By having others help me come up with the questions I was able to create a better questionnaire.

In this section, I will go through some of the results of the questionnaire. To keep it shorter, I will avoid including every graph and question. Most of the questions will be presented in simple tables, with the question or the statement written above it. To make the tables faster and easier to read, I have highlighted the most popular answer with a green background color. Options that got no answers are left empty. The negative options are on the left side while the positive options are on the right (False/True, Bad/Good).

7.1.1 General CMS section

To get a better understanding of what kind of people were taking the questionnaire, a few questions regarding their general usage of CMS was presented. The questions do not directly tie to either Wordpress or Craft CMS, but are instead relevant for any CMS use.

A CMS is important for my daily activities at work (False/True)

Table 2.

			14,3%	85,7%

I have experience of X different CMS

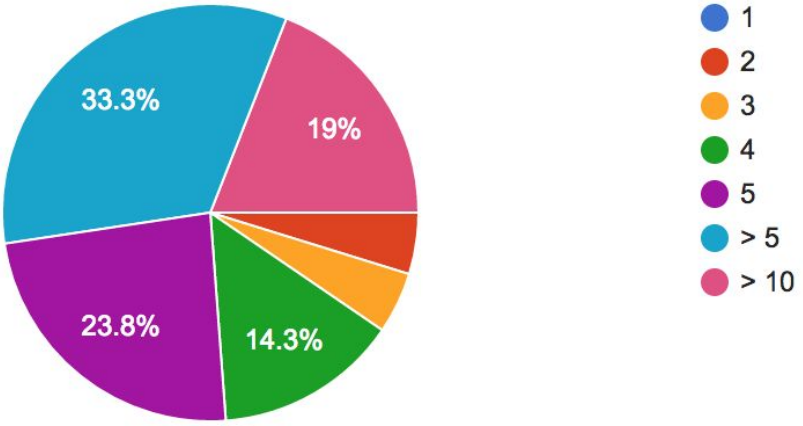

Figure 23. A graph displaying the answers to the question above

One CMS fits most projects (False/True) *Table 3.*

	19%	9,5%	42,9%	28,6%

My productivity can suffer depending on the chosen CMS (False/True)

Table 4.

		4,8%	14,3%	81%

As you can see from the figures above, the people who answered the questionnaire are clearly people who work with CMS daily, and the majority of them also have experience from multiple systems.

A staggering 81% of the respondents said that the chosen CMS greatly affects their productivity. The next sections presenting questions and statements for both Wordpress and Craft CMS will open up some of the reasonings behind this.

7.1.2 Wordpress section

PHP as a templating language is (Bad/Good)

Table 5.

38,1%	14,3%	9,5%	28,6%	9,5%

The learning curve of Wordpress is (Hard/Easy)

Table 6.

	4,8%	47,6%	38,1%	9,5%

The core features of Wordpress are (Bad/Good)

Table 7.

9,5%	33,3%	38,1%	19%	

The customizability of Wordpress is (Bad/Good)

Table 8.

14,3%	33,3%	23,8%	28,6%	

The overall performance of Wordpress is (Bad/Good)

Table 9.

19%	38,1%	42,9%		

The official documentation for Wordpress is (Bad/Good)

Table 10.

9,5%	19%	33,3%	28,6%	9,5%

It is easy to find help on the internet for my problems with Wordpress (False/True)

Table 11.

9,5%	4,8%	14,3%	38,1%	33,3%

The Wordpress dashboard from a developer's perspective is (Bad/Good)

Table 12.

23,8%	52,4%	14,3%	9,5%	

My overall score for Wordpress is (Bad/Good)

Table 13.

9,5%	42,9%	47,6%		

By counting the most answered options together and dividing them by the amount of questions (9), we can get an average score that gives an idea of the bigger picture. By counting the most popular answers together we end up in an average score of: 2,66.

7.1.3 Craft CMS section

Twig as a templating language is (Bad/Good)

Table 14.

	4,8%	4,8%	14,3%	76,2%

The learning curve of Craft CMS is (Hard/Easy)

Table 15.

	19,2%	42,9%	23,8%	14,3%

The core features of Craft CMS are (Bad/Good)

Table 16.

		9,5%	23,8%	66,7%

The customizability of Craft CMS is (Bad/Good)

Table 17.

		4,8%	28,6%	66,7%

The overall performance of Craft CMS is (Bad/Good)

Table 18.

4,8%		4,8%	61,9%	28,6%

The official documentation for Craft CMS is (Bad/Good)

Table 19.

	4,8%	14,3%	52,4%	28,6%

It is easy to find help on the internet for my problems with Craft CMS (False/True)

Table 20.

	4,8%	28,6%	42,9%	23,8%

The Craft CMS dashboard from a developer's perspective is (Bad/Good)

Table 21.

		19%	28,6%	52,4%

My overall score for Craft CMS is (Bad/Good)

Table 22.

		4,8%	33,3%	61,9%

The overall results of the Craft CMS section are very positive. As you can see, the majority of the answers are on the right side of the table, which in this case is the positive side.

The only question that got a majority response any lower was the question about the learning curve, where the majority responded 3 out of 5. This is most likely because in order to be able to use Craft CMS, you have to code almost everything by yourself as you can not use themes or plugins to set-up a site. Keep in mind that you also have to learn Twig, which nowhere near as popular as PHP.

The surprising thing is how many questions had the majority giving their answers to the rightmost column.

By counting the most answered options together and dividing them by the amount of questions (9), we can get an average score that gives an idea of the bigger picture. By counting the most popular answers together we end up in an average score of: 4,44.

7.2 Interview with experts

A short interview was held with two developers from Agency Leroy. The interview was held after the questionnaire, and some of the interview questions were directly related to the answers from the questionnaire. The idea behind having the interviews was to get help on analysing the results from the questionnaire as well as help on analysing other relevant questions that have been brought up in this thesis. By doing this I get a deeper knowledge about the subject and I am also able to present more trustful data.

The next subchapter will be divided into questions, answers and in some cases my own analysis.

7.3 Interview results and analysis

The first person that I interviewed was my colleague Hyein who I have been working closely with for the past year. She describes herself as a frontend developer and has two years of experience in the field.

The second person that I interviewed was my colleague Martin. Martin started as a freelancer in 2009 and was working mostly with logo design, web design and web development for various small companies. In 2012 he founded his own digital company Milk+Chocolate together with three partners. In 2014 Milk+Chocolate merged with Leroy, which is nowadays known as Agency Leroy. All in all he has been working as a professional for eight years.

Question: Briefly describe your experience with content management systems and why you think they are important

Hyein: I have experience with customized admin panels to provide different services. I have worked with Wordpress for a portfolio and I am also using it at work whenever necessary.

Craft CMS is the CMS that I use mainly nowadays. I think that a CMS is very important because this is where all the content of a website comes from. A static website is horrible when it comes to maintaining a website.

Martin: I have been working with content management systems since I started coding. The first website that I made was built with Joomla. If I remember correctly I have tried and worked with the following CMS: Joomla, Drupal, Processwire, CushyCMS, Concrete5, SilverStripe, CMS Made Simple, EZ Publish, Squarespace, ExpressionEngine, Wordpress and Craft CMS.
CMS are very important for me as I work with them daily. These days customers expect to be able to edit everything themselves, no matter if it is a small or large website.

Question: In the questionnaire that I held, the score for Craft CMS core features were great while they were relatively bad for Wordpress. What do you think the people who answered the questionnaire were thinking? What kind of core features are important for you as a developer?

Hyein: Craft CMS is a good choice if security is a concern. I think file management and the structure is easier and well thought out compared to Wordpress which can quickly become a mess with all the themes, plugins and customized features.
Craft CMS is well designed and structured as a whole. The readability of Twig is better than PHP, and therefore anyone who has some knowledge about programming will find it easy to use. The integration of Twig is powerful and makes it effective to develop with.

Martin: Wordpress core features are set and hard to tailor for the customer. If you do not like how something works or how it is named, you just have to live with it. You can of course change many things, but you have to code it yourself or use third party plugins that may not be as safe as they say they are.
There are not that many core features in Craft CMS, but they are built in the way that it is easy to tailor them to your customer's needs. You do not have to code during these

steps, it is just dragging and dropping. Craft CMS lets you focus on the frontend of the website.

Question: In the questionnaire, there was a question on how easy it is to find help on the internet. Wordpress got a good score while Craft CMS got a weaker one. What do you think is the reason behind this?

Hyein: In my opinion, it is simply because of the size of the community. While Wordpress is the most popular CMS in the world and has been developed for such a long time, Craft CMS's history is short. Thus, there are way more people that can offer help with Wordpress.
Also programming languages matter. PHP is one of the most common programming languages with many features while Twig is not familiar for many and is not even seen as direct competition for PHP.
Craft CMS can feel daunting at first, as there are no ready templates or themes as there is with Wordpress. Everything that you do with Craft CMS requires a developer, which is not always necessarily the case with Wordpress.
Another minor comment to this is that I think Craft CMS could have chosen a better name for their system, as the word craft is a very common name. This results in worse search engine visibility which can often be a bigger problem than what you would think of first.

Martin: A few things. Craft CMS is quite young, while Wordpress has been around for a long time. Wordpress is also open source and has a lot more users, while Craft CMS is not open source and has a lot smaller community.
The learning curve of Craft CMS is low, and the documentation is simple and short which means that people have fewer problems to post on the internet. Meanwhile, Wordpress has a lot of built in features and many plugins and themes that can cause a wide array of problems creating a bigger need of help everywhere.

Author comments: The smaller size of the community is a positive thing for me as well, as it is much easier to find relevant information and information of better quality. Hyein is correct in that Craft CMS can feel daunting at first, it really does assume a developer user in all cases, except for the content management part.
Martin has a good point in that the learning curve for Craft CMS is relatively low, and this can affect the amount of help that is asked for on the internet. Wordpress having features that are rarely used is also another point that has been touched on in this thesis.

Question: In the questionnaire, there was a question on how good the dashboard is from a developer's perspective. Craft CMS got an extremely good score here, while the score for Wordpress was bad. Why do you think this is the case?

Hyein: Craft CMS has a well structured, well designed UI and a clean admin panel compared to Wordpress and many other systems. The customizability of the backend structure is great and I think it is easy to keep everything well organized. Craft CMS also comes with options for full multilinguality. The few plugins that I use with Craft CMS are easy to maintain and they do not affect the structure of the navigation.
This is especially important when creating different content types, categories, sections and so on, as everything can easily be changed from one central location. This is not the case for Wordpress where plugins easily mess up the structure, and it can become a chore to navigate from one place to another to do any changes.

Martin: The dashboard of Craft CMS is clean and simple with all the control in the user's hands, and you do not even have to code anything to make it tailored for the customer.
While the dashboard of Wordpress is also somewhat simple and clean, especially with the newer versions, it also has many features and things that you do not even want in many of your projects. If you want to tailor the dashboard you have to code it yourself or depend on plugins which might now work as good as you would expect.

Author comments: The well designed dashboard of Craft CMS is something that I like a lot. You always know where to look for things, and it is easy to navigate from one place to another. There is nothing more frustrating than developing a big Wordpress site with multiple plugins, all of them having their own respective options in their own places. This seemed to be the general consensus of the people who answered the questionnaire.

Question: Why would you personally pick Craft CMS over Wordpress?

Hyein: As a developer, I want to build my websites myself. Craft CMS is easy to manage, develop for and keep it clean, and this is very important when it comes to maintaining my projects.
Personally, I like the fact that the community is smaller as there is less chance of bumping into false or outdated information. I also like the team developing Craft CMS as they are always listening to other developers and improving their product based on the feedback that they get.

Martin: It is just that much more simple to use, both for the developer and for the content manager or customer.

Question: What is a good target audience for Wordpress?

Hyein: For people who want to have a simple blog-type of website with no so many dynamic features or different content types. It also fits people who do not have to update or maintain their site. One of the big positives for Wordpress is that you can create a site with limited programming skills.

Martin: Looking from a developer's perspective the target audience would be developers that just want to use ready made themes and plugins and do not care how the code looks and works. To put it simply, it is for those who want ready made websites with little custom features.

Author comments: Quite simple yet good answers. It is true that setting up Wordpress and creating something that you can call a website is faster, but if you want full control you are better off choosing Craft CMS.

Question: What is a good target audience for Craft CMS?

Hyein: For people who want to create their website from scratch, and come up with their own solutions. This does require a deeper understanding of programming and related skills such as HTML, CSS and JS.

Martin: From a developer's perspective Craft CMS is good for those who want full control over the frontend and full control over the content structure. To put it simply, it is for those who want tailored websites that meet the customer's needs.

Author comments: Here we can once again see one of the main differences between the systems, in the end it comes down to which kind of developer you are and which kind of projects you are creating.

7.4 Thoughts regarding the interviews

All in all most of the answers align with my own research findings and results. The interview helped me getting a wider understanding of the subject, as hence was a success.
The answers that I gathered from these interviews played a role in my conclusions.

8 CONCLUSIONS

This section will function as a summary for the comparisons as well as for the thesis as a whole. The section will be divided into multiple parts summarizing the relevant parts.

8.1 Results

The conclusions and scores given below are a sum of the results gathered from the questionnaire. In the case of a draw, I also think about the results gathered from the interviews. I give a short explanation on each question before deciding a victor in each category. Everything happens in the same order as the results were presented in the previous chapter.

8.1.1 Templating

When it comes to templating both systems use same kinds of solutions, because both are based on PHP. However, Craft CMS is using Twig while Wordpress relies on traditional PHP. Templating is an important thing for both systems.
By looking at the answers on the questionnaire it is safe to say that Twig is considered to be the better templating language. Nevertheless, PHP is still very usable for templating.

The winner in this category is: Craft CMS

8.1.2 Learning curve

How hard is it to learn this system? A learning curve is dependant on multiple different factors. What kind of programming language is the software using, what kind of problems does it try to solve? What kind of help is available, whether it be in the form of official documentation, tutorials and guides, discussion forums or other community driven content? Many software related things can be relatively easy to learn, but hard to master.
The results were very even for both systems. Because of the huge Wordpress community and all its guides and courses I would say that it is a bit easier to get into Wordpress than Craft. This is especially true if you do not consider yourself a developer.

The winner in this category is: Wordpress

8.1.3 Core features

The core features of any system are important, but not every system is necessarily defined by it. In the case of Wordpress, the core features are very good if you are creating a website for blogging, but are otherwise very lacking.
The core features of Craft CMS are a bit more well rounded, and it does not include features that can be thought as useless by many.

The winner in this category is: Craft CMS

8.1.4 Customizability

Customizability is how easily we can create alternate solutions for our projects. Often times these solutions come in the form of plugins or in Wordpress's case, themes.
The potential to customize Wordpress is basically endless with the help of plugins, but this does not usually work out as planned because of bugs, incompatibility problems and security issues. If you have the skill it is recommended to use Craft CMS.

The winner in this category is: Craft CMS

8.1.5 Performance

Performance is dependant on multiple things. This comes down to the very functionalities behind the systems, but also solutions such as caching and image transformations to mention a few. By default, Wordpress does not offer as many solutions in improving the performance of your website as Craft CMS does.

The winner in this category is: Craft CMS

8.1.6 Official documentation

Documentation is important for any software, and CMS is no different. It is critical that there is a place where the user can find information about the software at hand. In this case, "documentation" points at the official documentation, not referring to any kind of unofficial documentation, articles, user discussions or other help forums.
The results were quite even, but in the end Craft CMS won. This is partly because while the documentation of Wordpress is good by itself, it can also be confusing because of the immense size of it. The layout and structure of the Wordpress documentation is not as well thought out as the Craft CMS one, and this is a very important factor to take into consideration. A documentation that is easy to read makes it easier and faster to work.

The winner in this category is: Craft CMS

8.1.7 Help on the internet

The size of the community is often a deciding factor in this category. Relevant channels in this category are forums, guides, courses and other helpful tools. The results of this category were very even.

The winner in this category is: Wordpress

8.1.8 Dashboard

From a developer's perspective, the dashboard is very important tool. This is where the backend of the website gets created, and where all its content gets managed. With the help of a well designed structure, Craft CMS succeeds in avoiding the clutter that plagues the dashboard of Wordpress.

The winner in this category is: Craft CMS

8.2 Score summary

The answers gathered from the Wordpress section were mostly neutral or negative, as could be seen in chapter 6.1.2. In the case of Wordpress, the average score for the most popular answer was 2,66.

The answers gathered from the Craft CMS section were mostly on the positive side, as could be seen in chapter 6.1.3. In the case of Craft CMS, the average score for the most popular answer was 4,44.

With the results gathered from the comparisons, questionnaire and interview we can see a clear difference between the two systems.

If you are looking for something new to try, I would recommend Craft CMS.

If you are looking for something that is fast to get up and running and something that it is easy to find help for, choose Wordpress.

All things considered, the winner of this thesis and comparison is Craft CMS.

8.3 Thoughts and further reading

With the help of this thesis, I managed to complete one of my personal goals which was getting a better understanding of both systems and how they compare to each other. Personally I am going to continue developing websites with the help of Craft CMS. However, Wordpress is still living with me because of having to do maintenance on older websites.

I would suggest to give Craft CMS an honest try and see how it fits your projects. For further reading on Craft CMS, visit its official website where you can download the project and read more about it from the documentation.

Whatever you think about the results of this thesis, I hope you decide to give other systems and solutions a go, instead of being stuck on using the same one.

9 REFERENCES

https://craftcms.com/docs/introduction (Craft CMS introduction, brought 19.3.2017)

https://codex.wordpress.org/ (Wordpress Codex, brought 19.3.2017)

https://straightupcraft.com/articles/twig-quick-start-and-twig-templating-key-concepts (Twig quick start and twig templating key concepts, brought 20.3.2017)

https://developer.wordpress.org/reference/ (Wordpress code reference, brought 19.3.2017)

https://wordpress.org/about/features/ (Wordpress features, brought 20.3.2017)

https://codex.wordpress.org/Function_Reference (Wordpress functions, brought 18.4.2017)

https://sucuri.net/website-security/website-hacked-report (Websites hacked trend report, brought 25.3.2017)

https://w3techs.com/technologies/details/cm-wordpress/all/all (Usage statistics and market share of Wordpress for websites, brought 23.3.2017).

The Elements of Content Strategy (Erin Kissane) (page 35)

http://www.comentum.com/what-is-cms-content-management-system.html (What is a Content Management System (CMS)? , brought 21.4.2017)

10 SWEDISH SUMMARY

10.1 INLEDNING

I detta thesis arbete har jag bestämt mig för att jämföra Wordpress och Craft CMS med varandra. Kraven för ett bra system är höga i dagens läge, och konkurrensen på marknaden är hård. Med tanke på populariteten av Wordpress, varför skulle man bestämma sig för att använda något annat system? Med hjälp av jämförelser mellan Wordpress och Craft CMS kommer jag att försöka svara på denna fråga.

Jag kommer att kortfattat beskriva vad ett CMS är och vad de kan användas för. Jag introducerar både Wordpress och Craft CMS för att ge läsaren en bättre helhetsbild av systemen. Efter det kommer jag att utföra en jämförelse mellan dessa system.

Jag går in djupare i systemen för att lära mig hurudana lösningar de använder, och med hjälp av praktiska exempel, min egen kunskap samt mina kollegors erfarenhet tar jag reda på de positiva och negativa sidorna av båda systemen. Med hjälp av dessa jämförelser försöker jag ge läsaren en orsak att pröva Craft CMS, istället för att fortsätta använda Wordpress som deras enda CMS.

10.2 MOTIV FÖR ÄMNET

Som en Online Media studerande och heltidsanställd utvecklare, spelar CMS en stor roll i mitt vardagliga liv. Före jag började arbeta inom branschen var Wordpress det enda systemet jag kände igen. Detta berodde på att det var det enda systemet som vi blev lärda i skolan. Jag tänkte inte så mycket på dessa system, de kändes endast som ett krav för en webbsida, ett onödigt steg mellan webbsidan och dess innehåll.
Då min egna erfarenhet som en webbutvecklare växte, insåg jag hur viktiga CMS är. Beroende på projektet i fråga kunde det vara en bra idé att välja ett specialiserat CMS. Oberoende hurdant projektet man arbetar med bör en utvecklare undvika att bli fast med samma system och samma lösningar, utan istället borde man leta efter nya lösningar och se fram emot att lära sig något nytt.

Craft CMS var ett system som jag bekantade mig med då jag började arbeta på Agency Leroy. Det var första gången jag arbetade med något annat CMS, och för första gången kände jag mig intresserad i ett CMS. Jag tyckte att Craft CMS hade briljanta lösningar, speciellt i jämförelse med Wordpress. Men det måste ju finnas något väldigt fint med Wordpress, eftersom den är det populäraste innehållshanteringssystemet. Vad som fortfarande förvirrar mig är hur det finns så många webbutvecklare och kunder som väljer Wordpress för sina webbsidor och projekt, utan att ens tänka på andra lösningar.

10.3 MÅLET MED AVHANDLINGEN OCH DESS MÅLGRUPP

Meningen med detta arbete är att jämföra Craft CMS och Wordpress. De saker som jag kommer att forska och presentera är:

- Vad är CMS
- Introduktion för både Craft CMS och Wordpress
- Jämföra båda systemen och presentera resultat
- Analys av svar på ett frågeformulär och expert intervju

Målet med arbetet är att:

- Få en bättre förståelse av båda systemen
- Försöka bevisa att Craft CMS är ett bättre system
- Ge en orsak att pröva Craft CMS

Målgruppen för arbetet är:

- Utvecklare, designers, innehålls redaktörer, kunder och studenter inom branschen
- Andra personer som har att göra med webbsidor och innehållshantering på webben

Jag antar att läsaren har grundläggande kunskaper i teknikerna som används i skapandet av webbsidor och webbapplikationer.

10.4 ARBETETS BEGRÄNSNINGAR

Arbetet kommer att jämföra och analysera Craft CMS och Wordpress. Med hjälp av jämförelser och praktiska exempel kommer jag att försöka ge läsaren en bättre förståelse för båda systemen.

Jag kommer inte att förklara grundläggande koncept för webbutveckling.

Detta arbete kommer inte att fungera som en guide för systemen i fråga.

De kodexempel som demonstreras är nödvändigtvis inte fall av så kallade “best practices”, utan kod som fungerar för mig personligen.

Arbetet kommer inte att handla om redaktörens eller frontend användarens upplevelser.

Arbetets fokus ligger i att förklara skillnaderna mellan systemen i fråga, och allt detta baserar sig starkt på utvecklarens behov och perspektiv.

10.5 RESULTAT

Till följande skriver jag kort om resultaten av resultaten från frågeformulären och intervjuerna. Baserat på dessa resultat väljer jag en vinnare i varje kategori. Ifall

resultaten är väldigt jämna använder jag mig själv av min egna samt mina kollegors gemensamma kunskap om ämnet. Jag skriver en kort beskrivning av varje kategori före jag presenterar vinnaren.

10.5.1 Templating

Båda systemen använder liknande lösningar i sina templates, och själva koden är också väldigt liknande eftersom båda använder sig av PHP. Craft CMS använder sig av Twig, och Wordpress använder sig av traditionell PHP. Hur man bygger sina templates och strukturen för webbsidan är viktigt för alla projekt.
Genom att analysera resultaten är det enkelt att se att Twig är bättre språket då det kommer till templating. Traditionell PHP är fortfarande en bra och användbart språk.

Vinnaren i denna kategori är: Craft CMS

10.5.2 Inlärningskurva

Hur svårt är det att lära sig systemet? Inlärningskurvan är beroende av många olika faktorer. Vilket programmerings språk används, vilka problem försöker den lösa och på vilket sätt? I form av officiell dokumentation, guides, diskussionsforum och annat användarskapat innehåll har man tillgång till? Många problem som är relaterade till mjukvara kan vara lätta att lösa med hjälp av dessa verktyg.

Resultaten för denna kategori var väldigt jämna mellan båda systemen. Eftersom Wordpress har en otroligt stor användarbas finns det flera guides och kurser som ger användaren många bra inlärningsmöjligheter. På grund av det påstår jag att det är lättare att lära sig Wordpress. Detta är sant speciellt ifall man inte är en webbutvecklare.

Vinnaren i denna kategori är: Wordpress

10.5.3 Grundfunktionalitet

Grundfunktionaliteten är en viktigt sak för alla system, men alla system är nödvändigtvis inte definierade av den. I Wordpress fall är funktionaliteten av en ny installation bra om du vill skapa en enkel sida för bloggandet, men relativt svaga ifall du vill skapa något mer avancerat.
Craft CMS funktionalitet är på en högre nivå, och kräver inte lika många plugins för att kunna göra saker. Craft CMS har inte heller lika många funktionaliteter och verktyg som kunde anses som onödiga i många fall.

Vinnaren i denna kategori är: Craft CMS

10.5.4 Editerbarhet

Editerbarhet är hur lätt man kan skapa alternativa lösningar i systemet. Oftast kommer det fram genom plugins eller themes i Wordpress fall. Möjligheterna till att ändra på funktionalitet och lägga till nya saker är nästan oändliga i Wordpress, men oftast stöter man på problem då det gäller version hantering, uppdateringar, inkompabilitet och säkerhet. Ifall man har förståelighet och kunskap är det bättre att använda sig av Craft CMS som inte är lika beroende av plugins.

Vinnaren i denna kategori är: Craft CMS

10.5.5 Prestanda

Prestandan av systemen är beroende på flera faktorer. Det är frågan om funktionaliteterna bakom systemen, men även tekniska lösningar som till exempel hantering av bilder och deras optimering samt cacheminne.
Wordpress innehåller inte många lösningar för att förbättra prestandan av en webbsida, medan Craft CMS har flera.

Vinnaren i denna kategori är: Craft CMS

10.5.6 Officiell dokumentation

Officiella dokumentationen är en väldigt viktig sak för all mjukvara. Det är kritiskt att de finns ett ställe där användaren kan hitta pålitlig information av mjukvaran i fråga. I detta fall är det den officiella dokumentationen som talas om, inte användarskapat innehåll som till exempel Wikipedia.
Resultaten var relativt jämna, men som vinnare valdes Craft CMS. Orsaken bakom detta är att även fast dokumentation av Wordpress är bra, kan den samtidigt vara väldigt förvirrande på grund av hur stor den är och hur dåligt många saker är presenterade. Dokumentationen av Wordpress är inte lika bra uttänkt, vilket är en viktig sak då man letar efter information. Lätt läsbarhet är viktigt för att kunna arbeta lättare och effektivare.

Vinnaren i denna kategori är: Craft CMS

10.5.7 Hjälp på internet

Storleken av användarbasen är oftast avgörande faktorn i denna kategori. Relevanta hjälpmedel i denna kategori är diskussionsforum, guides, kurser och andra verktyg. Resultaten var väldigt jämna, men på grund av att det finns mycket mera information om Wordpress var det lätt att välja en vinnare.

Vinnaren i denna kategori är: Wordpress

10.5.8 Dashboard

Från en webbutvecklares perspektiv är dashboarden ett viktigt verktyg. Det är här var sidornas struktur och deras funktionalitet blir skapade och definierade. Med hjälp av en väl planerad backend struktur lyckas Craft CMS undvika kaoset som plågar Wordpress dashboarden.

Vinnaren i denna kategori är: Craft CMS

10.6 SAMMANFATTNING AV RESULTATEN

Resultaten som samlades från Wordpress delen var huvudsakligen neutrala eller negativa. Medeltalet för Wordpress var 2,66.

Resultaten som samlades från Craft CMS delen var huvudsakligen på den positiva sidan. Medeltalet för Craft CMS var 4,44

Med hjälp av resultaten samlade från jämförelserna, frågeformulären och intervjuerna kan vi se en tydlig skillnad mellan systemen.

Om du letar efter något nytt skulle jag rekommendera Craft CMS.

Om du letar efter något som är snabbt att komma igång med och något som det är lätt att hitta hjälp för skulle jag rekommendera Wordpress

Med tanke på resultaten är Craft CMS vinnaren i detta arbete.

10.7 TANKAR OCH FÖRSLAG

Med hjälp av detta arbete lyckades jag få bättre förståelse för båda systemen och hur de jämför mot varandra. Personligen kommer jag att fortsätta utveckla sidor med Craft CMS. Hur som helst kommer jag även att arbeta med Wordpress eftersom jag ofta gör uppdateringar och mindre ändringar till äldre sidor som använder sig av detta system.

Jag föreslår att du ger Craft CMS ett försök, och ser hur den skulle passa dina egna webbprojekt. Besök deras officiella hemsida för att ladda ner systemet och för att bekanta dig med dokumentationen för att komma igång med Craft CMS.

Vad än du tycker om resultaten av detta arbete, hoppas jag att du kommer att pröva andra system och lösningar, istället för att bli fast i samma gamla arbetssätt.