

Future Trends of Content Management Systems

Zachary Brueske

CONTENTS

I	Introduction	1
II	Background	1
II-A	What was the status quo prior to this new technology?	1
II-B	What was the pressing need addressed and catalysts that caused development and introduction of new technology?	2
II-C	What was the introduction phase (of this new technology) like? . .	2
II-D	What is the new status quo? . . .	3
III	Technical Analysis	3
III-A	Getting Started	3
III-B	Database	3
III-C	User Interface	4
III-D	Pages	4
III-E	Plugins	4
III-F	Themes	5
III-G	Template Files	5
III-H	CSS Files	5
III-I	Metadata	6
IV	My Project	6
V	Future Trends	6
V-A	User Interface Changes	6
V-B	Implementing New Languages . .	7
V-C	Mobile Development	7
VI	Conclusion	7
	References	7

Abstract—Content Management Systems are tools that make building websites exponentially easier. They came about because of how hard it was for some people to build websites because they did not know how to code them. Content management systems use simple user interfaces and implement modern technology to create the environment necessary for this. This has enabled many non-tech-savvy people to build their own websites and blogs and has expanded the Internet in ways that would not be possible without content management systems.

I. INTRODUCTION

In the late-1980s, the World Wide Web became open to the public and soon people began making their own websites to publish to the Internet. To create a website for your company or organization everything had to be coded by hand. You had to have someone who knew HTML and CSS to develop and create a website that functioned well. This process would take a lot of time and effort, with most of the sites being created this way being simple text. In the early 2000s groups of web developers began to create an easier way for people to build websites; a way that allowed them to not have to know everything about HTML and coding to create a spectacular website. This is where the first content management systems came about. Content management systems allowed a wider audience to build websites but still use complex items to their advantage through simple interfaces and well coded environments.

II. BACKGROUND

A. What was the status quo prior to this new technology?

The first website started being developed by CERN in 1989 and was released in 1991 over twenty years after the World Wide Web was first created. It was a basic site that was created using HTML, it told people how to get the code and create their own websites. It also gave information about who was behind the project of developing HTML and how people could help out with adding more to the web.

HTML, which stands for Hypertext Markup Language, was developed by Tim Berners-Lee at CERN. It was based on an existing language called SGML that was used on IBM machines at CERN [12]. He also developed a client program, essentially a web browser and editor, which was used to create and view HTML files [12]. This was one of the most important advances in the history of the Internet but no one seemed to care. The problem with it was it only worked on Apple's "next" computer, which never became a mainstream device [12]. What helped HTML and the World Wide Web (WWW) to take off was the release of a browser, called Mosaic, for Unix in 1992. Mosaic allowed people to view not just the line by line of text but also view colors, graphics and animations on the page, this along with the point and click ability with an integrated mouse made the browser

a huge success. Then in 1993 Mosaic was released for the PC which allowed Windows to gain much needed access to the WWW.

At the time websites were still very basic, with only being able to use HTML in the browser didn't allow much in the way of customizing a web-page. You could add pictures and colors with the new browsers but it was in 1996 when Cascading Style Sheets (CSS) was integrated into Microsoft's Internet Explorer 3 that websites began to change the look and feel to more advanced web-pages [2]. Everything was still hard-coded and done in an Integrated Development Environment (IDE) so the process of creating a website was slow with many files which required a great knowledge of coding to be created.

B. What was the pressing need addressed and catalysts that caused development and introduction of new technology?

With the web becoming increasingly popular people needed easier ways to develop websites and manage the content that was on them. This is due to the fact that there were so few people who knew how to code websites at the time because computers were just starting to become a common item in the household and also because all websites were still created manually using code, which is not an effective way of managing a website. When creating a website you have many pages and each of those pages have hyperlinks, a reference to data (or another page on a website) that the reader can directly follow by clicking on, to other pages. This means you end up with a complex network of these hyperlinks that all have to be managed made sure they all work and fit together in the website, not an easy task when done with just code. This led to the creation of content management systems (CMS). However, many of the first CMS's were just frameworks on which a CMS could be built.

The first major CMS, WordPress, came about because of this need to simplify the creation and management of websites. It started in 2001 as a product called b2 cafelog, created by Michel Valdrighi, as a "classy news/weblog tool" that you simply had to type something in and hit "blog this" for it to be created as a page on your blog/website [1]. However, it was dormant by 2003 and development had stopped [11]. This led to Matt Mullenweg and Mike Little, forking the code for b2 and turning it into WordPress. Forking is the act of bifurcating a free software project, taking it in a new direction [11]. Mullenweg and Little wanted to take the good things about b2 and implement them with some improvements that they were worried b2 wouldn't have had, such as being forward compatible. This was the focus of Matt and Mike's first improvements to b2, now being called WordPress. In the late 90s a browser war

happened, much like today between Chrome, Safari, and Internet Explorer, so the makers wanted their technology to be able to be used on all browsers. They focused on HTML semantics and web standards to avoid issues between browsers and make sure web-pages created with WordPress would generally work better.

At the same time the second most popular CMS today was also being developed. This CMS is today known as Drupal, but when it was first created as a website used as the gathering place of a few University of Antwerp students after they graduated. These students shared news about new technologies, such as moderation, syndication, rating, and distributed authentication [3]. Drop.org slowly turned into a personal experimentation environment and was eventually released as an open source project to enable others to use and extend the experimentation platform and further improve its capabilities.

C. What was the introduction phase (of this new technology) like?

The launch of WordPress sparked immediate project growth [11]. Two days after the release of WordPress 0.7 (the first official release) they added another developer to WordPress, Donncha Caoimh who was the lead developer for b2++ another fork of b2. This allowed them to release WordPress 0.71 a month after the release of 0.7 which included many improvements but most notably a boost in performance speeds of 300% [11]. These quick improvements and the blogging boom of the early 2000s helps propel WordPress into the lead position of blogging software and easy website creation.

Plugins were added later on as ways to enhance the already usefulness of WordPress. The first plugin was the *Hello Dolly* plugin. All it did was randomly display a lyric from the Louis Armstrong song Hello, Dolly in the top right of the admin panel [11]. It was intended as a guide for others on how to make plugins and not really for anything serious. About the same time Matt Mullenweg created WordPress.org which gave the community a home and a place to share ideas and comments about WordPress. In addition to this it was a place to download the latest release of WordPress and keep up to date on the development of WordPress. It also served as an example of how one can make a simple to maintain website using WordPress, instead of just a blogging platform.

Drupal had similar success early on after it's initial release. In 2004 Drupal community members helped build "DeanSpace" for Howard Dean, a member of the U.S. Democratic Party running for the 2004 U.S. presidential election. "Dean Space" was a collection of 50 disparate, unofficial pro-Dean websites that allowed users to communicate directly with one another as well

as with the campaign [4]. After the campaign ended, members of Dean's Web team continued to develop a Web platform that could aid political activism by launching CivicSpace Labs in July 2004. This was the first company with full-time employees that was developing and distributing Drupal technology [4].

D. What is the new status quo?

Today WordPress is responsible for 23% of the CMS market and the vast majority of websites are no longer designed using just hard-coding [8]. Many new programs have come out that allow you to manage a website offline and then publish the pages to the web when done editing them, such as Microsoft Expression. The changes that WordPress has come up with have now made websites easier to create and maintain and it has influenced many people to create similar content management systems and online hosting services.

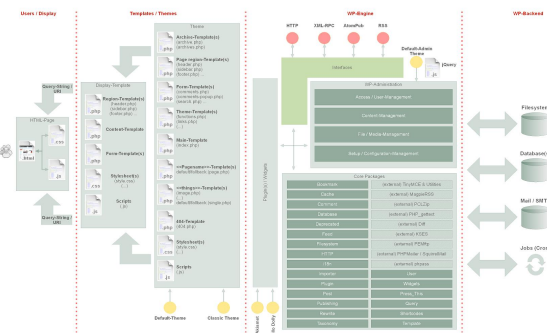
The WordPress community continues to grow with many new additions to blog, websites, plugins and themes that come out every year and will continue to as long as WordPress enjoys the majority share of the CMS market.

Today Drupal is a robust CMS and is estimated to be used by as many as 10 million websites and continues to gain many prestigious members, including Whitehouse.gov. Development members for Drupal now exceed 35,000 and new updates continue to be released with Drupal 8 on the horizon.

III. TECHNICAL ANALYSIS

All content management systems have a similar architecture which consists of a database that holds information and sends it to the various other elements when they need it.

Fig. 1. An example of the architecture of a CMS.



A. Getting Started

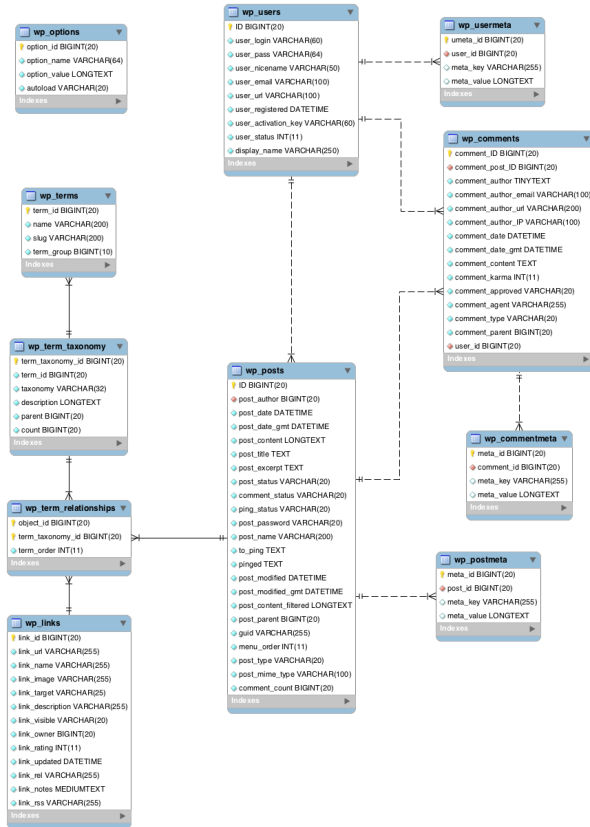
Getting a website set up on a content management system is a rather painless operation. Before you do

anything you must first establish your domain name for your site. You do this by first thinking of a domain name for your site. If its a blog most people would use the name of the blog or their own name for the domain. If the website is for a store or company then you can easily use that. Once you have a name, verify that it is not taken. Just go into your browser and search for the domain name you have chosen to see if a site pops up with that name already registered. Keep in mind there are many other domain extensions you can use, the most common are .com, .org, and .net. Finally once you have ensured that the domain name is not taken you must purchase it. This is usually done using a domain registrar such as Go Daddy or Register.com. Some of these companies will provide their own way to easily install WordPress on their servers, but if they do not you just need to upload the files to the host and into a directory called /wordpress and then run the install script. In the address bar in your browser you just need to type the following: `http://domain.com/wp-admin/install.php` and follow the on-screen instructions until WordPress has been activated. Setting up your own website in WordPress is just as simple as other content management systems. In fact they often brag about having a five minute installation. This does not include the time it takes to find your own hosting service and configure WordPress the way you want. The only thing you need to make sure the hosting service has is PHP and MySQL installed on their servers so you can actually use WordPress. Different content management systems use different languages as their building blocks. Drupal for example uses PHP, but if you do not like PHP you have many others to choose from that use different languages as their base code, such as JAVA or Microsoft ASP.NET.

B. Database

A content management systems database is the place where it stores everything that is involved in the website that database is for. This includes pages, comments, users who are registered on the website and many more things. Since the database is handled by the CMS, users do not have to worry too much about the structure or implementation of the database. The database takes care of the connections between specific parts of your website, for example if you change the theme of a page the database will connect that page with the specific theme you chose. You do need to know about the database of your CMS when you want to develop things for the CMS, such as a plugin. For this you might need to know how the CMS you are using stores its data and relationships between data in the database. The majority of CMS's use some form of SQL as their database, however some exist that use other databases such as Flat-File Database.

Fig. 2. An example of a CMS database and it's connections.

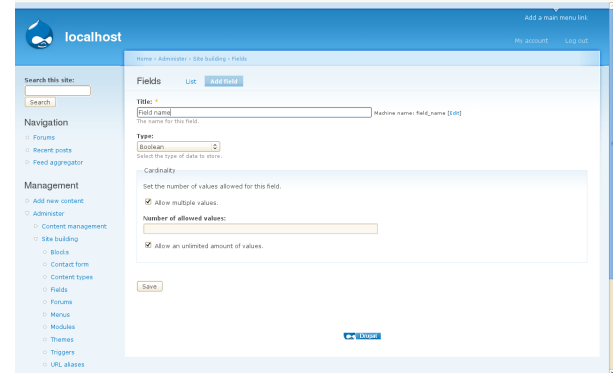


C. User Interface

The user interface is where all the behind-the-scenes things happen. It is where you create the look of your website and edit the content. It holds the settings, pages, and everything else you need to customize your site. When you first log into your websites administration panel it will show something similar to WordPress' At a Glance module. This shows you what is going on in your blog or website right now, right this very second [13]. It tells you how many posts have been made, how many comments users have added to the site, and the number of pages you have in your entire site. It also shows you a little bit about the version of the CMS you have installed. This includes the actual version number of the CMS, the name of the theme you are using, and other privacy settings you have enabled. Along the left sidebar of the user interface you can see the many other useful tabs to click on. The ones you will use the most are the Pages tab, the Appearance tab, and the Plugins tab. The Pages tab is where you create and edit all the pages on your blog/website. Any content you want on your site will have to be added using this tab. The Appearance tab is where you choose the theme you want to use, from any you currently have added to your

WordPress/themes folder on your host, and any changes that can be made to the color or appearance of the theme using the Dashboard. The Plugins tab is used to add new plugins you want to your site, activate new plugins you have installed, or delete plugins that are no longer needed.

Fig. 3. An example of a User Interface.



D. Pages

The most important part of a CMS is a way to handle all of the content you are going to put onto it because that was one of the initial reasons they were created. The most common way of doing this is using pages. A page is what everyone will look at when they open your website on the Internet. Pages hold everything from text to pictures and links to other parts of your website. The database of your CMS tells the page what plugins will be active on the page, what theme the page should use and populates the page with all of its content.

E. Plugins

One of the useful parts of a content management system is the plugins. CMS's are great on their own, but by adding plugins you get more out of your CMS and allow it to do almost anything. In a CMS, plugins are add-on programs that give the CMS ways to handle web content [13]. This means they cannot function on their own and need that CMS to work since they extend the capability of the CMS you are using. They can also be used to give your visitors an improved viewing experience by giving them things to interact with or participate in on your site, but only things you want to add. In WordPress there are currently over 34,000 [9] plugins to choose from and more are being created each time someone needs something new to be done. Plugins are developed by people in the CMS community using the code that CMS was created with. Each CMS handles plugins a little differently and each has their own name for them as well. WordPress calls them plugins, in

Drupal they are referred to as modules, while in Joomla they are called extensions. All have a large selection to choose from but the installations are all a bit different. For Drupal you must install each module manually. This means downloading the appropriate files for the module and placing them in the sites/all/modules area of the Drupal directory [3]. Joomla extensions require a bit more work on the part of user. You must first download the .zip folder for the desired extension, then once done, go into your administration section of your site to upload and install the extension [5]. Some extensions even have extra steps one must take after all this to get it installed and working properly. WordPress is by far the easiest to install plugins because you only need to search for a desired plugin and then click the Install Now button on the plugins page. This does not work for all plugins and sometimes you must install it manually the way you would for Drupal. One must be careful when installing plugins so they do not get overwhelmed with the sheer number of them and wanting to install many plugins thinking they will all add to their site. While it is true a few plugins will make your website much better, once you start installing too many the speed of your site begins to suffer. Just be sure to keep only those you need and are currently using to prevent your site from being loaded down with plugins.

F. Themes

A theme is a collection of files that creates the structure and design of your site [6]. It is the first impression a visitor has when they come to your site and can determine if they will stay or leave. All CMS's use themes to diversify the look of sites so not everyone has the same appearance in their webpages. Each CMS has many free themes users can install and use on their websites and most CMS's create a basic theme that is updated each year to get everyone started once they install that CMS and begin working with it. Since these themes are very basic, content management systems often have a large repository of user created themes for others to download and apply on their site. Joomla for example has over 8,000 free themes for its users to download and customize to fit their site [5]. Installing themes in WordPress is just as easy as installing a plugin. You only need to find a theme that would work for your website and click the install button. Once it is installed you only need to go to the appearance tab on the Dashboard and click activate on the theme. You are then able to customize some of the parts of the theme in the same area of the user interface. Some content management systems allow you to further customize the theme you have chosen with child theme, in Drupal these are called sub-themes. A child theme is a theme that inherits functionality from another theme. You are then

able to modify the child theme without interfering with the code or functions from the original or parent theme. This way any changes you make will not break your website and you are free to make changes or additions such as adding new menus or further tailoring the styling of the theme to fit your site. In CMS's, themes are made up of two types of files, template files and cascading style sheet (CSS) files. These can be for altering anything from the header of the page to a sidebar. These files are recreated in the child theme and can be adapted depending on what it is you would like to change. There is a slight difference in files between CMS's. For example WordPress does not use the .info files that Drupal does. In Drupal, a .info file is used to define everything that will go into your theme [3], such as meta data, style sheets, and template files. One file that WordPress has that Drupal does not is a functions file. This file acts like a plugin and is used to enable features in the theme [], such as sidebars and other menus.

Fig. 4. A quick comparison of the number of plugins and themes of the top 3 CMS's.

CMS:	WordPress	Drupal	Joomla
Plugins:	34,000+	28,000+	8,500+
Themes:	2,500+	2,000+	8,000+

G. Template Files

Template files are one type of file that makes up a theme. Template files control the structure and layout of your pages [14] using PHP code to tell the CMS what each part of the webpage should do and have in it. This includes the header of a website, the footer, and sidebars. Template files are useful because they add flexibility into the design of your website. You are able to customize each template file, usually using a child theme, to fit into your website how you want. You are also able to use multiple template files for different pages. For example if you have one group of pages that you want to add a new menu into just those pages, you only need to create a template file for those pages that implements the previous template files and adds the new menu you want. The rest of your pages will use the previous templates but the pages you select will use the new file to alter the way they look.

H. CSS Files

In a content management systems theme, the CSS files defines the colors, fonts, and how to display PHP elements [10] in your website or blog. These files allow for the precise control of the style of a template file or the overall look of a website by changing colors, fonts,

margins and padding and more to create a website that looks the way you want.

I. Metadata

Metadata is data that describes other data. Specifically for a CMS it reveals information about the created pages on a website. This could be something such as the author or creator of a webpage, the date that page was created or a file size in a download section. Administrators of a website may also add their own custom metadata tags to fit into other uses besides just information about a webpage. This data is sometimes not able to be seen by users who are accessing the site but is readily accessed by computers. Previously metadata was used as a tool for search engine optimization. Website admins would take commonly searched keywords or phrases and put them into a website's metadata to boost the hits it would get when people would search those keywords or phrases. This was considered unethical and many search engines now skip over webpage metadata when sweeping over a page when searching.

IV. MY PROJECT

My project has been ongoing for a few months now. This is due to the fact that it has become a part of my internship during the school year and over the summer. My project is the Trilinos group's website. It details all information about the Trilinos packages, information about the package developers, events with Trilinos, and much more. I initially worked on testing the requirements of the website on WordPress to see if it was a sound option to use as a content management system or if not should we switch to another popular CMS, such as Drupal or Joomla. What we were looking for in a CMS was one that could handle a large number of pages, currently over one hundred, and one that was easy to use and implement changes on. WordPress fit all of these things and I began getting to work on customizing the website to the desires of the many developers of Trilinos. I first worked with the overall organization and flow of the website. They wanted a website that was easy on the eyes and quick to navigate through so users and developers would be able to find what they wanted quickly and easily. My first attempt at this was through the use of a child theme. I tried to add new menus and control the look of the site to make it easier to navigate through the large number of pages. This tactic ended up working but in the process I found an easier way to improve the navigation of the site using a plugin that allowed me to add PHP code in a sidebar and using that to create a custom menu. This menu is used on all pages and depending on where you are in the site the list of pages in the sidebar expands accordingly. For

example if you are on a top level page it only shows other top level pages, but once you go into a child page, a package page for example, it expands to show the brother and sister pages of that page as well as any child pages of the page you are on. This helped to direct the movement of people to find things in the site easier. My current objective is to help the page creation of package pages move much smoother. While creating a page in a CMS is not a hard task filling it with all the information becomes tedious and time consuming. I am working with the metadata of pages to attempt to auto fill pages with the required information. This information is all about the packages and includes things such as the package name, the current release of the package, if it is in the latest release of Trilinos or not, and the contact email address of that specific package. The goal is to use an array that holds all the information in metadata and some PHP code in the body of the page that will autofill by calling the metadata using

```
wp_query
```

and other WordPress specific PHP commands. Each of these ideas have been tested on a sandbox site before being implemented on the actual Trilinos.org website. This is done to make sure everything works together before being deployed and avoiding a potential shutdown of the site if a certain part of something I am working on gets messed up.

V. FUTURE TRENDS

While con

A. User Interface Changes

One of the things on the top of the list of what CMS developers will be looking to change would be the user interface and the overall experience the admins have when editing their sites. This is currently easy on some content management systems, however others, such as Drupal, tend to be a bit more complex and take a lot more time to learn and develop the skills necessary to create your own website in the CMS. They hope to focus their efforts to make the CMS easier to use for all people. This would increase their user base and get more people involved in using their software. WordPress has already started to do this with their famous "one-click" installs for plugins, themes and WordPress itself. WordPress co-founder, Mike Little, thinks they still need to do more though. He says "As soon as you learn the basics in WordPress, it's all really easy, but until you learn the basics, it's actually not...I think this is the biggest challenge, because as more and more people use WordPress, essentially you'll have more people who are less tech-savvy, and it's important to make it easy for

them to come on board,” [7]. An increase in users would also mean an increase in developers for CMS’s. This means more people developing plugins and other add-on software to go along with the popular content management systems. These people will continue to drive the usefulness of CMS’s and create more opportunities for people to enhance their websites and blogs.

B. Implementing New Languages

On the back end there is the possibility of these CMS’s using more updated computer languages in the future. Not entirely replacing what the CMS is built with but as an addition to enhance the abilities of the CMS. One example of this is the implementation of JavaScript. With newer content management systems coming out and already using JavaScript to handle much of their content, older CMS’s such as WordPress and Drupal have yet to take advantage of this great language, but appear to be working on converting the code over. This was mentioned by Mark Jaquith, a lead developer of WordPress when he said ”we’ve been tackling many of the core WordPress experiences over the last 18 months and giving them intense JavaScript makeovers” [6].

C. Mobile Development

One last thing that content management systems have yet to really take advantage of would be the use of mobile devices. I do not mean being easily viewed on mobile, since there are many themes in WordPress and the other CMS’s that handle mobile sites well. What I mean is mobile applications that work with WordPress and allow users to edit content on their WordPress site from their mobile device. This is seen as very important because there are some countries, such as China or India, where the majority of online use is done from a mobile phone or tablet. Since people are constantly on their phones this would be a great investment that would increase usability of a CMS greatly and probably increase the amount of users. The vision of WordPress is ”to start a blog and engage with the world of blogging 100% from the mobile device”. [6] This would allow users to blog on the go and update content when it is most fresh in their minds. The current WordPress mobile application is receiving mixed reviews from users. They like some of the aspects about it but they feel many of the things in the application seem useless. However, since the mobile application is open source just like WordPress developers are able to fork the project and edit out things they do not want in the mobile version. This doesn’t seem like it would be the best way of handling things since you are splitting your users into different groups, even though this would improve some aspects of the application. They should instead listen to the concerns

of the current users and base their updates on what the majority of people want.

VI. CONCLUSION

As the Internet continues to grow so will content management systems. This is because of the continued effort to make them easy enough for general public and not just computer science, web designers. To do this they will improve the current technologies and implement new ones, such as simpler user interfaces, new coding languages and mobile applications that allow users to access their websites or blogs. Today content management systems are used on around 40% of websites and continue to receive hundreds of thousands of downloads a month [9] [3] [5]. With more people using CMS’s the web will continue to grow as it has exponentially as it has the past few years, currently having over 1 billions websites registered.

REFERENCES

- [1] b2.
- [2] Cascading style sheets. http://en.wikipedia.org/wiki/Cascading_Style_Sheets.
- [3] Drupal homepage. <https://www.drupal.org/>.
- [4] Drupal wikipedia. <http://en.wikipedia.org/wiki/Drupal>.
- [5] Joomla homepage. <http://joomla.org>.
- [6] Mark jaquith. markjaquith.com.
- [7] Mike little: Wordpress specialist. <http://mikelittle.org/blog/>.
- [8] Usage of content management systems for websites.
- [9] Wordpress homepage. <http://wordpress.org>.
- [10] Brian Bondari. *WordPress 3 Plugin Development Essentials*. Packt Publishing, 2011.
- [11] Siobahn McKeown. *WordPress: Freedom, Community and the Business of Open Source*. WordPress, 2013.
- [12] Johnny Ryan. *History of the Internet and the Digital Future*. Reaktion Books, 2010.
- [13] Lisa Sabin-Wilson. *WordPress for Dummies*. John Wiley & Sons Inc., 2014.
- [14] Rick Shreves. *WordPress 3 Cookbook*. Packt Publishing, 2011.

APPENDIX A

The courses that helped me the most in this project were my early computer science classes, such as CSCI 150 and CSCI 160, and a couple of my upper division classes, Parallel Computing and Data Mining. My early computer science classes helped me with the coding that was needed for developing the parts of my project that I could not just use a plugin for. An example of this would be when I had to create a custom sidebar that would expand specific sections when on certain pages in the website or the code necessary to populate pages with metadata. This problem solving aspect came from my previous experiences in 150 and 160 where we had to work independently to solve the lab problems that were assigned to us. Parallel Computing and Data Mining were a factor in my ability to write a scientific report and implement more descriptive language rather than ”flowery nonsense”. This allowed me to develop a straight

forward presentation and paper that communicated the information in a concise way that was easy to understand to someone who was not an expert in what it is that I have been working on.