

2014

CMSVega: A Content Management System for the Layman

Roshan P.J. Nedumpurath
nedumpurath_roshanpaul@columbusstate.edu

Follow this and additional works at: https://csuepress.columbusstate.edu/theses_dissertations



Part of the [Computer Sciences Commons](#)

Recommended Citation

Nedumpurath, Roshan P.J., "CMSVega: A Content Management System for the Layman" (2014). *Theses and Dissertations*. 110.

https://csuepress.columbusstate.edu/theses_dissertations/110

This Thesis is brought to you for free and open access by the Student Publications at CSU ePress. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of CSU ePress.

CMSVEGA: A CONTENT MANAGEMENT SYSTEM FOR THE LAYMAN

Roshan Paul J. Nedumpurath

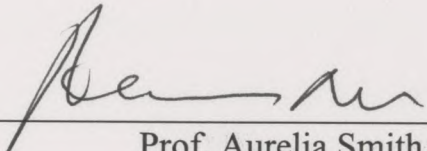
Table of Contents

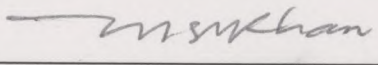
CMSVega: A Content Management System for the Layman 3
 by
Roshan Paul J. Nedumpurath 4

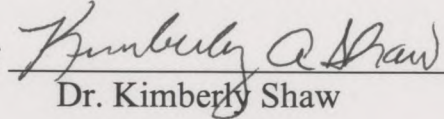
Project Overview 6
 Basic Configuration 6
 Page Management 7
 Modules 7
 Template Engine 7
 Security 8
 Deployment 8

A Thesis Submitted in Partial Fulfillment of Requirements of the CSU Honors College for Honors in the degree of Bachelor of Science 10
in 10
Computer Science 14
TSYS School of Computer Science 15
Columbus State University 15

Development Timeline 16
Blind Study 17
Results 17
Future Work 18
Conclusion 18
References 18

Thesis Advisor  Date 3-28-14
 Prof. Aurelia Smith

Committee Member  Date 3/28/2014
 Dr. Shamim Khan

Honors Committee Member  Date 3/28/14
 Dr. Kimberly Shaw

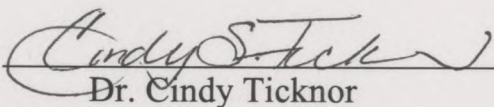
Honors Program Director  Date 3/28/14
 Dr. Cindy Ticknor

Table of Contents

Abstract	3
Proposal	4
Project Overview	6
Basic Configurations	6
Page Management	7
Modules	7
Template Engine	7
Security	8
Deployment Prerequisite	8
CMSVega Screenshots	10
Front-end view	10
Administrator View – Page Modification	11
Administrator View – Gallery Module Administration	12
Development Timeline	13
Blind Study	14
Results	15
Future Work	16
Conclusion	17
References	18

Abstract

Content management systems are becoming increasingly convoluted to use. Existing website creation systems require users without programming knowledge to scour documentation. A CMS should be intuitive, something a user can pick up and use right away. Simplicity is key when targeting layman users who have no web development knowledge. This project aims to create a simple and intuitive content management system, CMSVega, which users can pick up and use with minimal web development knowledge. A blind study was conducted to compare CMSVega against popular Wordpress, Drupal and Joomla systems. After having recreated a webpage using each CMS, participants were asked to pick their favorite, where 70% chose CMSVega. Participants gave CMSVega a rating of over 76% satisfaction in categories such as ease of use, time required to complete task, and achieving overall success of task. The outcome of the results suggests that users favor simplicity in applications.

called Microsoft FrontPage which allowed the use of a What You See is What You Get, or WYSIWYG, editor in order to develop a webpage (Microsoft, 1996). Even with this assistance, a user still needed to have a lot of know-how in order to publish a webpage. The first PHP CMS, PHP-Nuke, was released during the early 2000s. It utilized the PHP programming language and a MySQL database engine to run the CMS. It gained popularity partially due to shipping with many pre-installed modules, as well as having the ability to have add-on modules. The system also shipped with the ability to have customized themes. But what constitutes a good content management system?

Dr. Khalid S. Hussain, the Dean of Information Technology at King Abdulaziz University (2012), suggests that there are eight characteristics any good CMS should

have. The CMS should be (1) intuitive, (2) customizable, (3) extensible through modules. **Proposal** Content management systems, or CMS, have been around for the past few decades. At its core, a CMS allows a user with minimal knowledge of programming languages to create and modify websites (Ngoma, 2011). The problem with the popular CMS such as WordPress, Drupal, and Joomla is the learning curve to use it. The aforementioned three combined, dominate the open source category with a market share of 95.5% (BuiltWith, 2013). While these systems are robust and feature-rich, a layman user is encumbered with the amount of configurations and setup that is required in order to create a website (Di Maio, 2001). So the question arises: Could we create a CMS with the layman user in mind? A system that allows a user with limited programming knowledge to create a beautiful website.

(2) Initially, websites could only be created by programmers who knew the HTML programming language. Then, in 1996, Microsoft brought out a product called Microsoft FrontPage which allowed the use of a What You See is What You Get, or WYSIWYG, editor in order to develop a webpage (Microsoft, 1996). Even with this assistance, a user still needed to have a lot of know-how in order to publish a webpage. The first PHP CMS, PHP-Nuke, was released during the early 2000s. It utilized the PHP programming language and a MySQL database engine to run the CMS. It gained popularity partially due to shipping with many pre-installed modules, as well as having the ability to have add-on modules. The system also shipped with the ability to have customized themes. But what constitutes a good content management system?

Dr. Khalid S. Husain, the Dean of Information Technology at King Abdulaziz University (2012), suggests that there are eight characteristics any good CMS should

have. The CMS should be (1) intuitive, (2) easily customizable, (3) extensible through modules, (4) eliminate end user programming, (5) optimized for speed, (6) secure, (7) well documented and lastly (8) conforms to best web practices. Ashley Vereyken (2011), a Master's in Computer Science and Technology graduate, suggests in her thesis, some key goals to look for in a CMS. They include, separate content layout and design, central graphic design management by using templates, database driven content administration, and web forms based content administration where users without expertise in HTML can publish content.

In her paper "The Usability of Content Management Systems", Dr. Paola Di Maio (2001), a professor at Indian Institute of Technology, breaks down the user's experience using a CMS into six metrics. They include (1) user's end result of attempting the task, (2) how much time it took for the user to accomplish the task, (3) the user's happiness with the overall experience, (4) the precision of the user's success, (5) the user's opinion of the system's functionality, and finally (6) the quality metric which is the weighed calculation of the user experience. These usability metrics can be used as guidelines in attempting to achieve greater user satisfaction when creating a CMS.

This project will be to create a content management system using the aforementioned principles in mind, CMSVega. Once completed, a study will be conducted to measure the effectiveness of the CMS. The study will be modeled after Dr. Di Maio's metrics as guidelines. The end goal is to have a tool, which a layman user can use to create a website, without feeling intimidated by the process.

CMSVega is a content management system that is created with usability and simplicity in mind. An intuitive system allows the user to jump “right in” with a minimal learning curve. With focus given to simplicity, a “no fluff” interface allows the user to focus on what is important: their content. A user is able to create pages with a WYSIWYG editor.

Modularity and customization go hand in hand with a good CMS (Husain, 2012). A user has the option to add or remove modules of content, such as Google Calendar or Google Maps, from their website with a few clicks of the button. If a user wishes to post their wedding photos, they can do that with a gallery module. Modularity greatly increases the functionality of a CMS. Along with modularity, a user should also be allowed to modify the look and feel of their website (Husain, 2012). In CMSVega, a user is able to dynamically change from one template to another on the fly. If they so choose, they are able to hire a web designer to design a template for them to use. This allows for great customization and branding for a website.

Basic Configurations

Basic site configurations for CMSVega includes options to edit the website title, upload a website logo, as well as configure Meta tags like description and keywords. The user is able to select the template to be shown on the website which loads from the database and is presented in a dropdown list. There is also an option to make the site active or inactive. Finally, a user can add their Google Analytics tracking code in order to have easy integration with the free Google Analytics website statistics software.

Page Management Along with templates, CMSVega has pages ending in an HTML. Next, a user is able to add or edit the pages on their website. Pages are what display the content that a user creates onto the website. Here, a user with the help of a WYSIWYG editor is able to add or alter the content of their website. Pages have other features such as a link name to specify what the page's link should be named, a title for the page, and a custom image to be shown before the content of the page. Each page has an active or inactive status which is useful when drafting pages not yet ready for public view. Finally, a rollback feature is available on each page. This can be used if a user accidentally modifies a page and wishes to go back to a previous version of the page. The WYSIWYG editor used in CMSVega is TinyMCE which is released as open source under the LGPL license.

Modules A few modules which are preloaded with CMSVega include a Contact Form module, Gallery module, and a Google Maps module. The contact form module allows a user to receive email messages from their website visitors. The gallery module allows a user to upload photos to create an image gallery. The Google Maps module enables a user to enter in an address and load a map of that address shown on the web page. Configurations for all of these modules are available in the administrative panel. There is an area where an administrator can change their name, email, and password.

Template Engine CMSVega has the ability to dynamically switch templates. The template engine supports dynamic loading of templates from the database. Developers have the ability to add templates to the system by uploading the resources files to the server and adding the

corresponding database entries. Along with templates, CMSVega has pages ending in an HTML extension to give visitors the feel they are viewing a static HTML web page. A feature on the Apache webserver called rewrite conditions is used in order to achieve this. Lastly, when logged in as administrator, there is an “edit” link under every page. This allows the administrator to quickly edit the selected page.

Security

Security on CMSVega is handled through PHP Sessions. The administrator credentials are stored in the database. The password is encrypted using PHP’s Mcrypt function using a dynamic hash. The hash is calculated with regard to the system file path of the CMS installation. Upon logging in, the system sets session variables in order to track the user’s administrative privileges. The system has a “forgot password” functionality which decrypts the password and emails it to the user. Form submissions is validated using JavaScript for client-side validation as well as PHP for server-side validation. File uploads are validated in order to ensure only valid file extensions are uploaded. Lastly, all MySQL transactions made by the system is “escaped” to prevent SQL injection attacks.

Deployment Prerequisite

CMSVega is created with an open source target deployment in mind. It is created to be compatible with a Linux, Apache, MySQL, PHP (LAMP) environment. The CMS is developed using a mix of HTML, CSS, JavaScript, and PHP. MySQL is used as the database to hold information. LAMP solution stack is completely open source and free to the public. This will ensure compatibility with the majority of cost effective web hosting solutions. LAMP environments are also optimized to work well with each other which

CMSVega Screenshots

will result in an optimized end product. To install, users will simply need to upload the files to their server and run the database script.

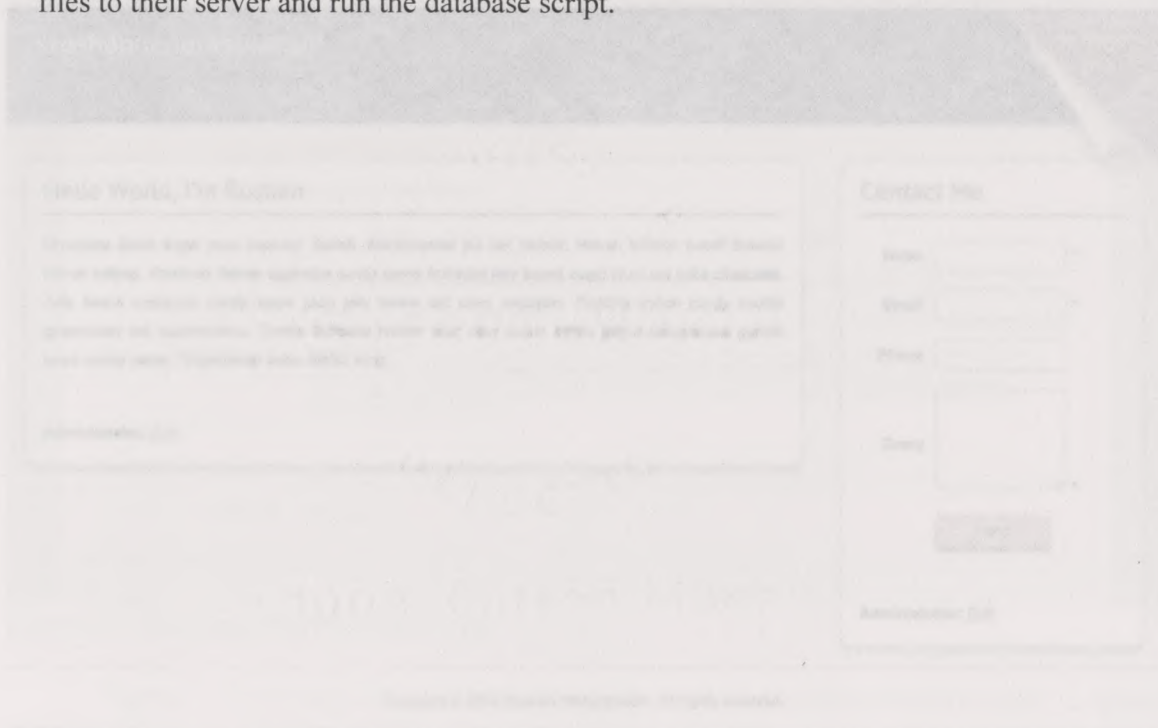


Figure 1: Front-end website

Figure 1 shows what a website user would see when visiting a website running CMSVega. The navigation bar shown on the top is generated based on the link order selected when creating new pages. The main page (left side of the page) and "Contact Me" sidebar (right side of the page) are loaded dynamically from the database and injected into the template. The template shown is created using HTML 5 and uses CSS 3 for styling.

Additionally, since this front-end view in this figure is that of an administrator, there are "edit" links under each page component. These edit links are a shortcut to edit the corresponding page in the administration area. An additional banner on the top right hand side also exists for users to quickly jump into the administration panel.

Administrator View – Page Management CMSVega Screenshots

Front-end view

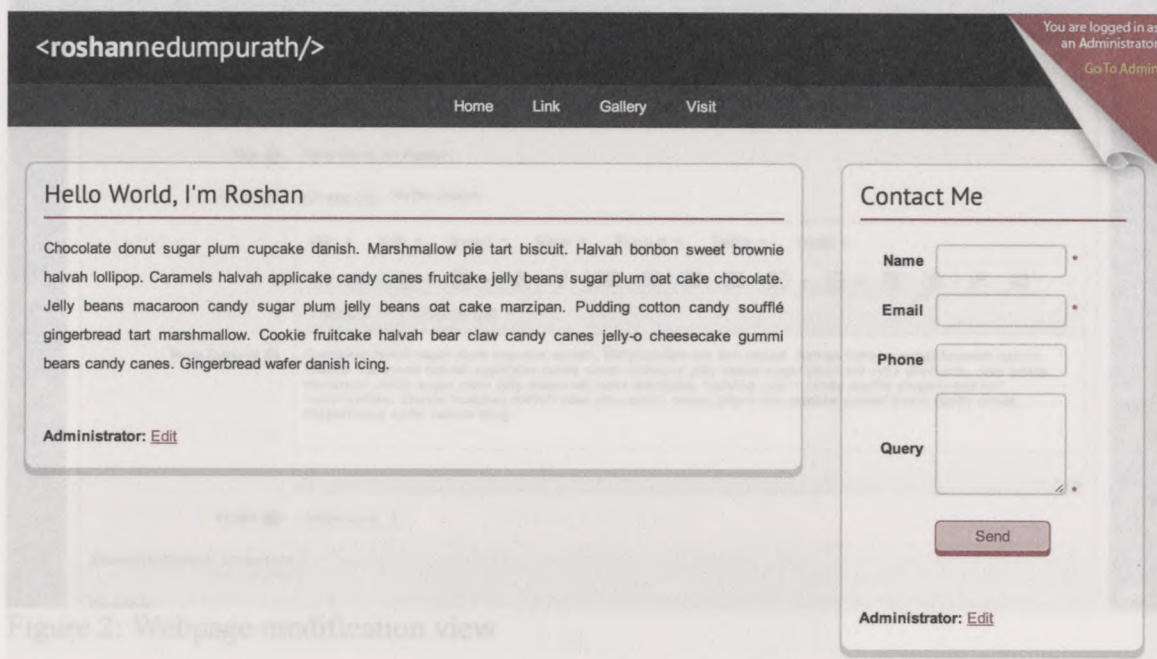


Figure 1: Front-end website

Figure 1 shows what a website user would see when visiting a website running CMSVega. The navigation bar shown on the top is generated based on the link order selected when creating new pages. The main page (left side of the page) and “Contact Me” sidebar (right side of the page) are loaded dynamically from the database and injected into the template. The template shown is created using HTML 5 and uses CSS 3 for styling.

Additionally, since this front-end view in this figure is that of an administrator, there are “edit” links under each page component. These edit links are a shortcut to edit the corresponding page in the administration area. An additional banner on the top right hand side also exists for users to quickly jump into the administration panel.

Administrator View – Page Modification Administration

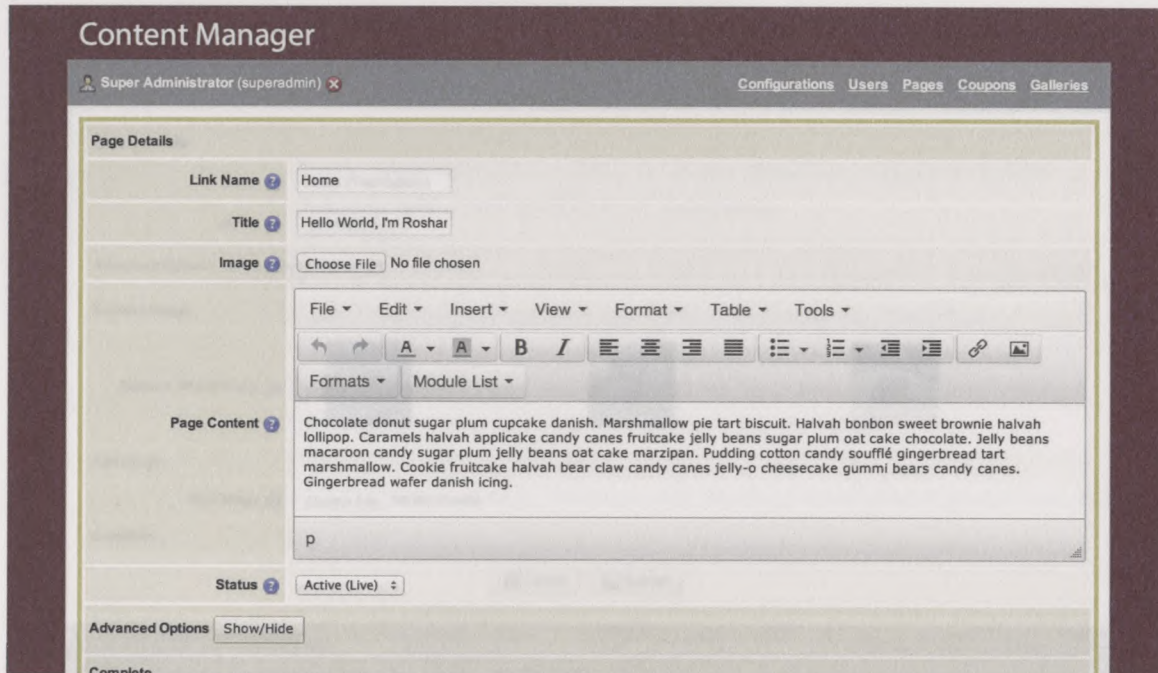


Figure 2: Webpage modification view

Figure 2 illustrates the user interface presented to an administrative user to create or modify a page. Users are required to enter a link name for the page, a page title, and the page content. The user must also select a page status which controls if the page is made live on the website or hidden. Lastly, a user can add an image to illustrate the page as well as pick from some optional advanced settings to further customize page.

Each module can be deployed on a page by selecting it in the Modules dropdown of the WYSIWYG editor. When a module is added or removed, the list is updated dynamically. This allows a user to easily create an image gallery and add it to a web page.

Administrator View – Gallery Module Administration

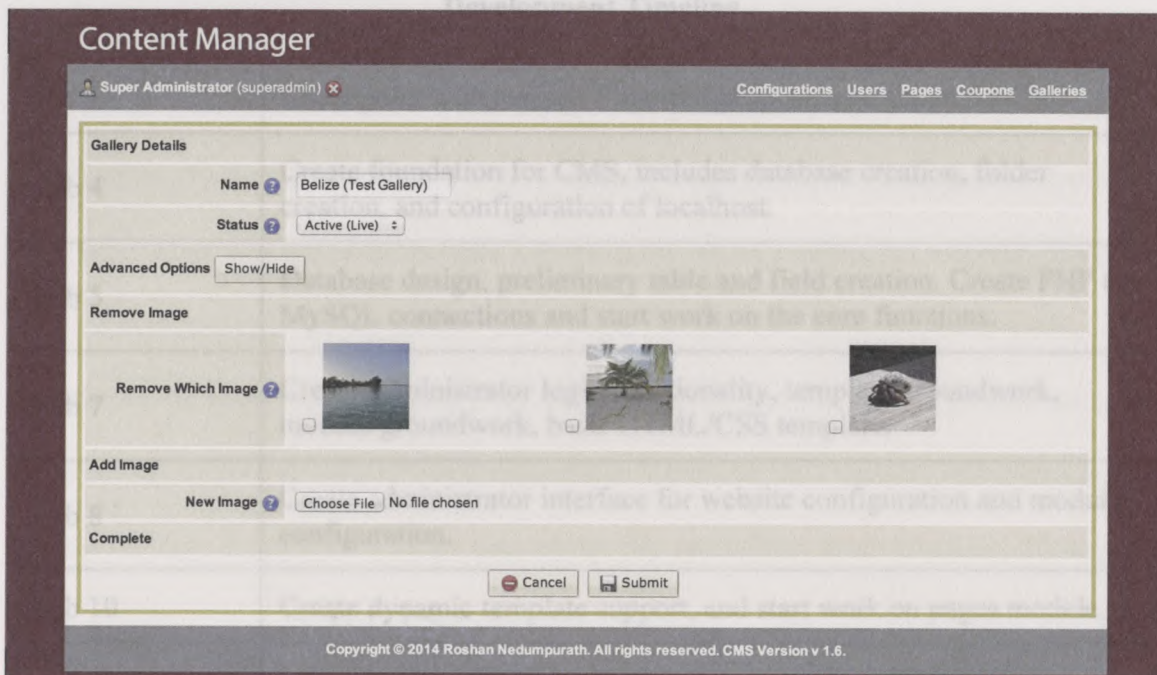


Figure 3: Gallery module administration page

Some modules used in CMSVega require an administration area. The gallery module is an example of this. Administrative users must be able to create new galleries, and add or remove pictures. This interface allows a user to upload files from their computers. The pictures are dynamically resized to preset or user specified dimensions. A Thickbox is used to present an enlarged version of the picture both in the administrative panel as well as on the front-end website. Like in pages, users can deactivate a gallery to hide it from the public view.

Each module can be displayed on a page by selecting it in the Modules dropdown of the WYSIWYG editor. When a module is added or removed, the list is updated dynamically. This allows a user to easily create an image gallery and add it to a web page.

Development Timeline

Jan 31	Checkpoint with mentor. Finalized proposal and scope for CMS.
Feb 4	Create foundation for CMS, includes database creation, folder creation, and configuration of localhost.
Feb 5	Database design, preliminary table and field creation. Create PHP to MySQL connections and start work on the core functions.
Feb 7	Create administrator login functionality, template groundwork, module groundwork, basic HTML/CSS template.
Feb 8	Create administrator interface for website configuration and module configuration.
Feb 10	Create dynamic template support, and start work on pages module.
Feb 11	Integrate TinyMCE into CMS and save resulting HTML into database. Create history table for revision rollback support.
Feb 12	Use jQuery to create form validations across the CMS. Ensure all MySQL queries are made SQL injection safe.
Feb 14	Create gallery module, contact module, map module, calendar module. Create support to send email via Google servers.
Feb 15	Integrate module support into TinyMCE. Create administrator forgot password and user info update functionality.
Feb 17	Finish up front-end design and create CSS for administration panel. Add icons to buttons for user friendliness.
Feb 19	Finishing touches, package and send to mentor.
Feb 21	Meeting with mentor, go over functionality and code.
Mar 14	Meeting with mentor, show final code & paper.

Blind Study

In order to measure the effectiveness of the newly created CMSVega, 10 participants were selected for a blind study. The participants were selected to have no programming background, and not to have known what a content management system is. Each participant was asked to recreate a given print-out of a formatted webpage using the four content management systems, CMSVega, Wordpress, Drupal and Joomla. After completing the task, the participant was asked to rate the content management system in the areas of (1) ease of use, (2) time required to complete task, and (3) achieving overall success of task. A score of 1 in an area meant the participant was extremely dissatisfied while a score of 5 in an area meant extremely satisfied. Lastly, the participant was asked to pick a content management system they would prefer using to manage their website, if they were to administer one.

My Amazing Study Abroad Trip

This is a short post created to share information about my study abroad trip to Belize last year.

Activities included:

- Hiking
- Snorkeling
- **Research**

Find out more: <http://studyabroad.columbusstate.edu>

Figure 4: Task given to participants to recreate

Results

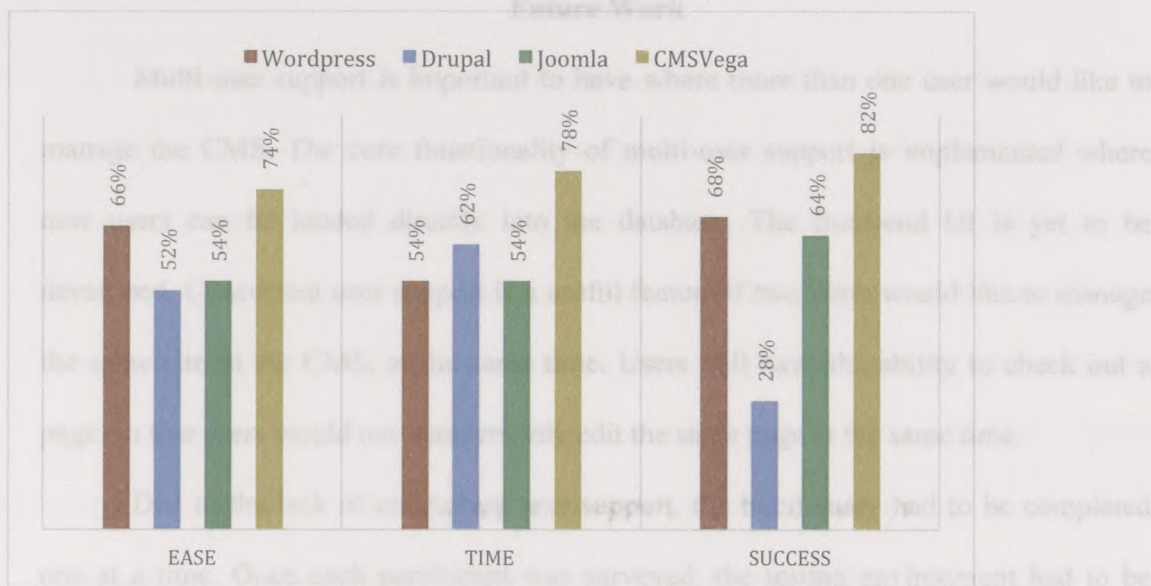


Figure 5: Average scores of survey results

Results indicate that participants preferred using CMSVega over the other options in the study. On average, participants considered to be 82% successful in completing the task. Participants considered the time invested to complete the task on average to be 78% close to what they expected. Lastly, participants rated CMSVega a 76% in the category of user friendliness. Wordpress came in second place, and Joomla close behind. Drupal was rated to be least successful, primarily due to the lack of a built-in WYSIWYG editor. For daily website management, 70% of participants chose CMSVega.

Future Work

Multi-user support is important to have where more than one user would like to manage the CMS. The core functionality of multi-user support is implemented where new users can be loaded directly into the database. The front-end UI is yet to be developed. Concurrent user support is a useful feature if two users would like to manage the same site in the CMS, at the same time. Users will have the ability to check out a page, so two users would not inadvertently edit the same page at the same time.

Due to the lack of concurrent user support, the blind study had to be completed one at a time. Once each participant was surveyed, the testing environment had to be rolled back manually for the next participant. Due to the time investment for each participant, only ten were able to be surveyed. Once concurrent user support has been implemented, a greater sample can be surveyed to achieve a more accurate representation of the results.

At its current form, CMSVega requires the knowledge of FTP and MySQL database import functionality to install. While this cannot be completely avoided, an easier method can be created where the skill involved in installation can be reduced. A setup package can be created where users can enter a few parameters for automated installation of the software package.

Lastly, code redundancy exists in the module administrator area. This can be reduced so all modules use the same administration structure. The CMS currently does not utilize OOP design patterns or stored procedures. These can be implemented for a more efficient end product.

References

- BuiltWith. (2014, March 07). Open source technologies web usage statistics. Retrieved January 25, 2014, from <http://trends.builtwith.com/cms/open-source>
- Burzi, F. (n.d.). Short history of PHP-Nuke. Retrieved March 06, 2014, from <http://tldp.org/HOWTO/PHP-Nuke-HOWTO/history.html>
- Di Maio, P. (2001, May). The usability of content management systems. Retrieved January 25, 2014, from http://www.academia.edu/687776/_The_Usability_of_Content_Management_Systems_
- Husain, K. S. (2012). Performance evaluation process of the content management system: Case study of Kau Marz system. Retrieved January 26, 2014, from [http://www.idosi.org/mejsr/mejsr11\(1\)12/20.pdf](http://www.idosi.org/mejsr/mejsr11(1)12/20.pdf)
- Microsoft. (1996, April 08). Microsoft announces FrontPage 1.1 with new pricing and availability of free beta on World Wide Web. Retrieved March 03, 2014, from <http://www.microsoft.com/en-us/news/press/1996/apr96/frntpgpr.aspx>
- Ngoma, S. (2011, September 17). Functional usability and flexibility of web content management systems. Retrieved January 25, 2014, from <http://www.congovision.com/WCMS-NEW.pdf>
- Vereyken, A. (2011). Analysis, design and implementation of a news and event content management system (NECMS) for the University of North Carolina Wilmington's Cameron School of Business. Retrieved January 26, 2014, from http://www.csb.uncw.edu/mcsis/complete/pdf/Ashley_Vereyken_Capstone_NECMS.pdf

