International Journal of Advanced Engineering and Technology
ISSN: 2456-7655; Impact Factor: RJIF 5.54
Received: 13-02-2019; Accepted: 16-03-2019
www.newengineeringjournal.com
Volume 3; Issue 2; April 2019; Page No. 74-80

Design and Development of Vlog Content Management System (VlogCMS)

Rasheed AA[1*], Maku BS[2]
[1] Department of Computer Science Technology, Federal College of Agriculture, Ishiagu, Ebonyi State, Nigeria
[2] Department of General Studies, Federal College of Agriculture, Ishiagu, Ebonyi State, Nigeria

Abstract
The high level of global adoption of Content Management System (CMS) as a web tool is undisputed because of its numerous advantages. These benefits are: users' friendly, ease of manipulation, content deployment and maintenance. Several inventions have emerged within this system; but video CMS (VlogMS) is rare. This is what gives impetus to this research. The aim of which is the design and development of a user-centered and secured content delivery VlogMS. Reviews of the most used CMS available were made. Basic concepts like: Video Blog, Content and File Management System; Security issues in CMS and Video Streaming were equally reviewed. The artefact was built using a Model – View – Controller (MVC) framework and the standard software architectural styles. The design, development, testing and evaluation of the software conformed to conventional standard as related to specific objectives of the research. Cross Site Scripting (XSS) observed in the comments section during testing was patched leading to the release of 1.1 version after the initial test run 0.9 beta version. Further improvement is recommended on the following area: "theme/template changer", Search Engine Optimization (SEO) configuration, plugin system, navigation manager and cleaner coding structure.

Keywords: design, development, vlog content management system (VlogCMS), content management system (CMS)

Introduction

As technology evolves, demands for media content over the internet rises. Consumers cannot seem to get enough of the content they have, so they search for more. With the increase in demand for media content, so also the increase in content distributors becomes inevitable. In order to meet these demands, platforms responsible for distributing these would have to upgrade to better services; and new media platforms to compete with these demands would come into existence.

Ownership of video blog has proven to be a challenging task to intending bloggers due to the technicalities that are involved in having a website to share thoughts and ideas they have. With time, they end up settling for third-party services that allow them share these ideas. Platforms like YouTube et cetera are often used. These platforms often impose restrictions which they have to abide by. When violated, their blog could be suspended or rendered inactive. Vlog Content Management System (VlogCMS) comes in handy to leverage these challenges.

CMS is an open source platform to manage contents with customized templates and plugins on the web or other devices. While this web tool is laudable to professionals, sub professionals and non-professional alike, it is fraught with numerous challenges. Veen (2004) acknowledged the following reasons as the cause for the failure of most CMS: outdated contents as a result of poor content management, uncontrolled content due to poor administration within organizations, security issues of online contents for both management and users and failure to provide user-centered design structure. In most cases, features are not user-friendly. Other challenges includes: mismanaged editorial process in determining what kind of contents gets to the users, content mix-up and editorial process within software architecture, threat of attacks due to the fact that it can be freely accessed by virtually every non-technical personnel. Besides, its source code can easily be understudied so as to ascertain its vulnerabilities by adversaries for eventual exploitation. This makes it easy to be compromised.

Vlog Content Management System [VlogCMS]) has never been created despite the popularity of CMS as web favourite. Since the debut of the first CMS on the internet called 'Enterprise Content Manager (ECM)' in 1992 by FileNet, till the release of WordPress in 2003 (which turns out to be one of the most influential CMS in the world today) and several others, the innovation never ended. This system has their peculiarities that distinguished them from others. With over 1200 CMSs existing with wide variety of features, stretching from consumer specific services like Article publisher system, portfolio management systems and e-commerce system amongst others; there would be no harm in innovating with VlogCMS, a video blog content management system as a member of the family.

Vlog Content Management System (VlogCMS) is designed as an open source Content Management System (CMS) to leverage on the stringent restrictions found in these third-party services. The project was designed as a CMS because CMSs are user friendly systems that anyone with basic knowledge of computer systems can conveniently manipulate. CMS contain an underlying architecture that enhances ease of design, content deployment and maintenance; without the need to crawl through the core system to effect changes which can be a complicated task. Also being an open source CMS, it gives advanced users with extreme coding skills the advantages to assert influence on the system features to suit their needs and add what they want or even take out functionalities that they feel is not exactly necessary.

It is on this premise that this study seeks to explore the design

and development of a Content Management System for Vlogs. The specific objectives of the research includes to:

a. design and develop a user-centered Content Management System Vlogs without restrictions on the content and level of Video blog that can be accommodated.
b. design a secured, user-friendly content delivery application.

Methodology

In managing version of the software, Software Versioning methodology was adopted. Software Versioning systems allows efficient sharing of a project, and hence eases team software development by allowing for reconstruction of the original design intentions and give subsequent variations in time (Romian et. al, 2005). This is done by assigning unique version names or numbers to every release. Within a given version number category (major, minor, patch), these numbers are generally assigned in increasing order and correspond to new developments in the software (Wikipedia, undated (b)). There are different Software Versioning methods that could be adopted. In this study, Semantic Versioning was adopted.

Controllers

The controllers are classes that interpret information provided by the model to the views. The following are controller classes with detailed information about each of them.

Admin Controller

This controller coordinates every operation performed by the administrator. The following are functions associated with this controller.

__construct():
This is a general method in PHP called the magic method. All libraries and models et cetera are declared in it so they would not have to be called every time a method is created.
add_category():
Receives input from respective field and inserts to database after create category button has been clicked
approve_comment():
Updates comment value is_approved to 1 after approve comment button has been clicked.
edit_account():
Updates account information with new information newly provided
images():
Retrieves images uploaded from specified directory
index():
Index method for the admin controller
login():
Returns database row in association with the information entered by user
logout():
Destroys sessions created when user logs in.
menu_navigation():
Displays views for adding menu navigation to headers
new_post():
Creates new posts and insert results to database
register():
Creates a new user with entered username, password and email
users():
Returns result of all registered users from database.

Home Controller

This controller handles items displayed on the main website including the Homepage, Categories, Post Details et cetera. The following are functions associated with this controller:
add_comment():
Inserts post comments into database from user inputs.
detail():
Displays information about posts and comments when post are clicked
index():
Displays the homepage when website is visited.

Install Controller

This controller handles the setup configuration of the system. Setup includes connection to MySQL, creating a database, creating tables, creating Administrator account et cetera. The following are functions associated with this controller.
index():
Initiates the MySQL connection page during the system installation.
database_creation():
Creates MySQL database after connection to MySQL.
tables_creation():
Creates tables into database when button is clicked
site_settings():
Inserts information from user input into database for system site settings
finish():
Displays the final process of the installation if installation is successful
delete_files():
Deletes all installation files after installation is complete.

Use Cases: Login Sequence Diagram

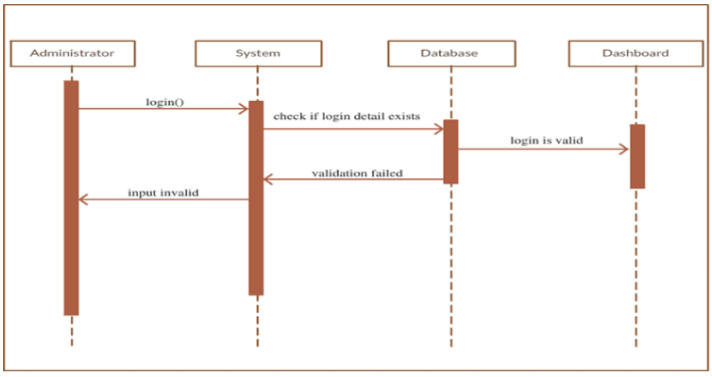

Fig 1: Sequence diagram for login

Requirements Gathering
In order to be able to know what end-user would want on the system, a questionnaire was designed to gather information in order to ascertain what end users might want. The Questionnaire can be found on the Appendix A.

Development and Testing
This section introduces the system designs in form of screenshots so the reader can get a physical preview of the system.

Preliminary Design
Content Management System has two major interfaces: the CMA and the CDA. The CDA would serve as a central interface that can be accessed by only the system administrator and authors. These interfaces consist of features for publishing contents and managing the views of the front-end interface. The CDA currently has two access levels: Administrator and Author. Administrator can basically do anything on the system whereas the author has limited features unlike the administrator. These features include publishing Vlogs, adding users, editing users account and other functions.

Setup Configuration
Before the system can be used, administrator has to go through a setup configuration phase. Here MySQL connection was made; database created. SQL tables would be automatically created; site settings were defined with information like: Administrator login details, website information like website URL, website title, description et cetera. After this information has been successfully added, a complete installation page comes up where administrator has to click "Complete Installation". Afterwards, installation files would be deleted for security reasons. The following are screenshot samples demonstrating the installation process:

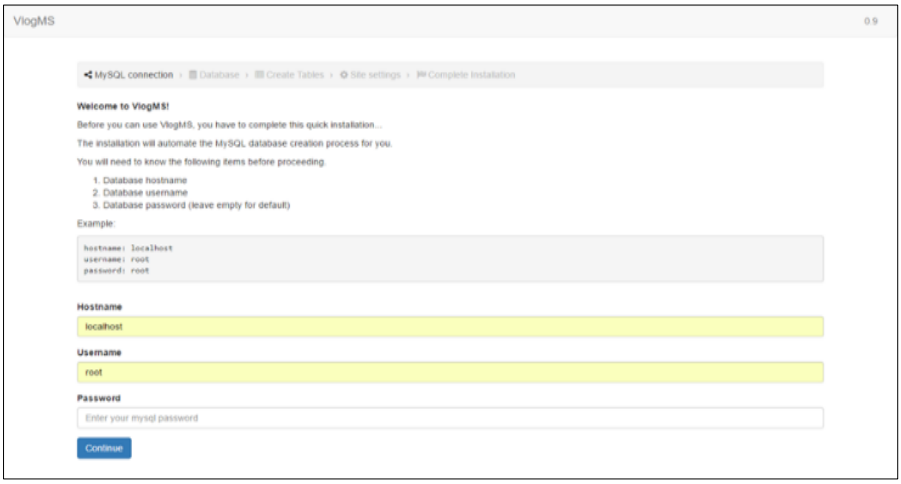

Fig 2: shows the first page of the installation process

When installing on a local system by default the hostname is localhost and username is root. Passwords are not usually specified on the local host by default so it can be left blank unless the user sets a password for the root user, then it has to be specified.

Fig 3: shows connection has been successful and user would be asked to enter a desired name for Vlog database

After connection has been successfully made user would be taken to the database creation page where desired name for database would be entered and database would be created automatically. If there is an error with this action database would not be created and user would be asked to retry until successful.

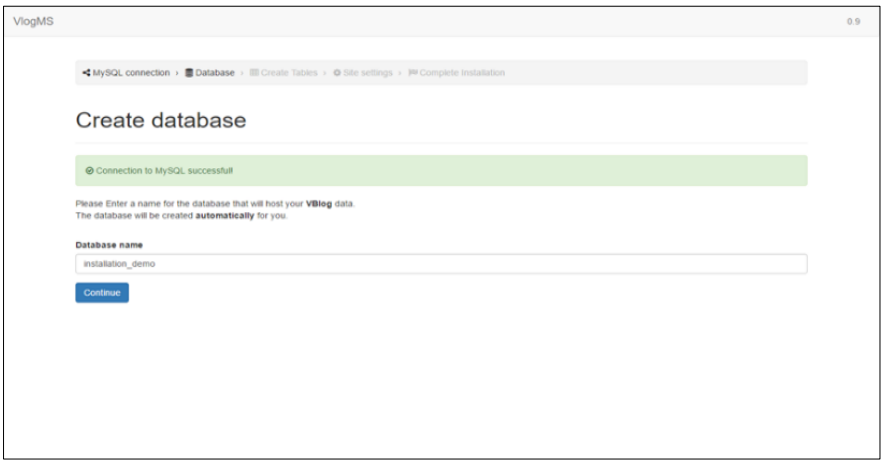

Fig 4: shows database creation was successful then user is asked to click continue to create required tables

After database creation is successful user is taken to the next phase; where "Continue" button has to be clicked to create tables required for the system to operate properly.

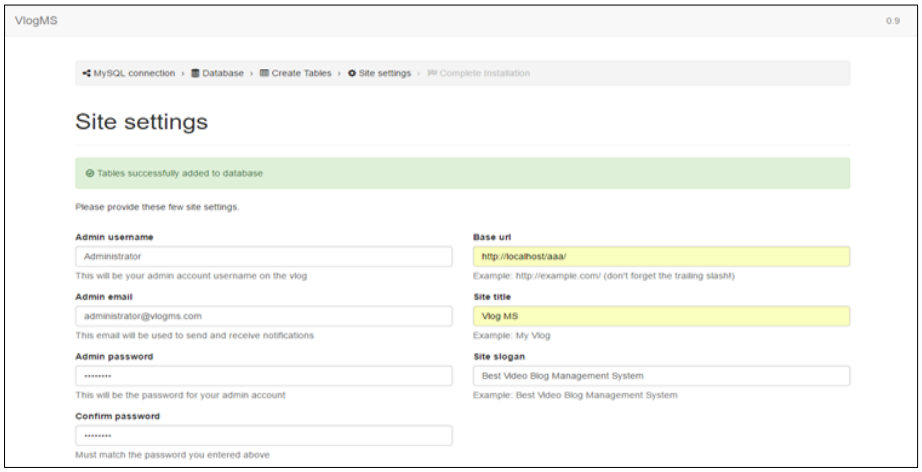

Fig 5: shows a sample how site settings are provided

On this page is where the administrator is created for the Vlog along with some basic settings for the site.

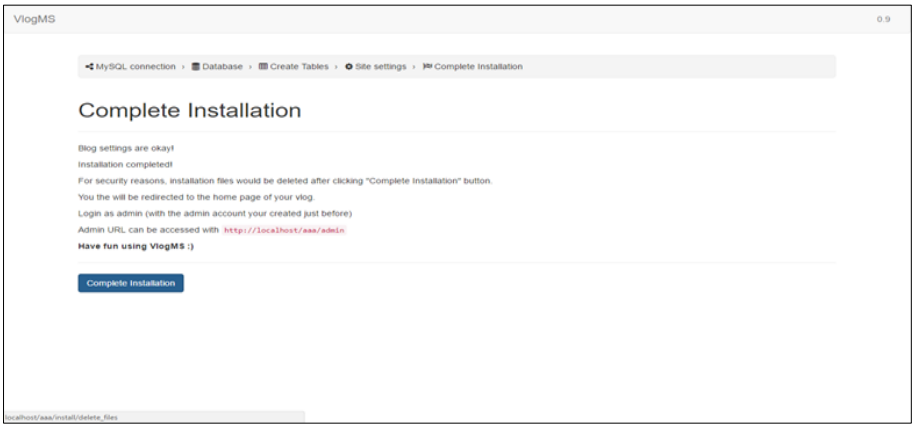

Fig 6: complete installation page

This is the final page of the installation process after clicking "Complete Installation". Files associated with the installation process would be permanently deleted so that someone with negative intentions would not have access to these files so as to use them to their own advantage as against that of the administrator. Although restrictions have also been made so these process cannot be repeated. It is just safe to delete the files rather than leave them to chance.

Login and Landing Page

There two landing pages: the first for administrators and the second for users. For administrators, the landing page known as "Dashboard". It is shown after their login has been verified. Although users' login has not been fully implemented on the system, Administrators can still add users and authors so they can post Vlogs. User registration would be fully implemented in later version of the system.

Currently only guest browsing is available.

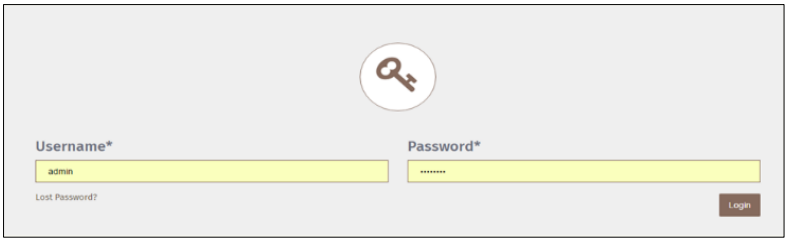

Fig 7: shows a sample of the login page

```php
$this->form_validation->set_rules('username', 'Username', 'required|alpha_numeric');
$this->form_validation->set_rules('password', 'Password', 'required');

if ($this->form_validation->run() == false) {
    // validation not ok, send validation errors to the view
    $this->load->view('themes/default/admin/login', $data);
}else{
    $username = $this->input->post('username');
    $password = $this->input->post('password');

    if ($this->admin_model->resolve_user_login($username, $password)) {
        $user_id = $this->admin_model->get_user_id_from_username($username);
        $user    = $this->admin_model->get_user($user_id);

        // set session user datas

        $_SESSION['is_blocked']     = (bool)$user->is_blocked;
        $_SESSION['user_id']        = (int)$user->id;
        $_SESSION['username']       = (string)$user->username;
        $_SESSION['logged_in']      = (bool)true;
        $_SESSION['is_confirmed']   = (bool)$user->is_confirmed;
        $_SESSION['is_admin']       = (bool)$user->is_admin;
        $_SESSION['is_author']      = (bool)$user->is_author;

        // user login ok
        if($user->is_admin == 1 || $user->is_author == 1){
            redirect(base_url('admin'));
        }else{
            $data->error = 'Only Administrator permitted';
            $this->load->view('themes/default/admin/login', $data);
        }
    } else {
        // login failed
        $data->error = 'Username and password do not match or you do not have an account yet.';

        // send error to the view
        $this->load->view('themes/default/admin/login', $data);
    }
```

Fig 8: shows the code associated with the login

The first two lines in the login code sets validation of login input from the users of the respective fields. On the first line, the username field is specified and is set to accept only alpha-numeric values; while the password field is only required. The first statement runs when the “login” button is clicked. The statement returns false if the validation does not correspond with the validation that has been set in the first two lines. If the statement returns true, the else block would run. Here the values inputted will be stored in respective variables. Then the nested if statement within the else block would run. The if resolves user login with the resolve_user_login method if this validation is okay, it check if the values entered corresponds with values in the database; then it returns the user unique along with the entire user information in the row. The values retrieved from the row are stored in session variables either as Integer, String or Boolean depending on the value being stored in the session. The nested if also has a nested if that checks if the user trying to login is either an admin or an author. If not, it returns a notice telling the user “Only Administrator permitted”. The last else statement runs when the resolve_user_login method returns false; which happens when username or password does not correspond with any value in the database.

Fig 9: shows sample of the dashboard

Headers
The website has a global header throughout the pages. It contains navigations to various pages as specified by the user. The default header navigations specified by the system are placeholders for values specified by the user, this includes the website title et cetera. Also, the search bar has been specified by default as header content. The other navigations and respective pages have to be created by the user in the CDA.

Fig 10: sample design of header content and navigations

Fig 11: shows how menu navigations can be added the header

Testing
Throughout the history of software/application development, no software is truly complete until the software is fully tested. The testing has to cover the following areas or more: functionality check, security test for vulnerabilities and also test to prevent regression. The main focus of testing was to ensure that the deployed system meets user standards for the operations described and to work as expected. Though, due to resource limitation and time, the system was not tested in large scalability when deployed by end users.
The VlogCMS system was tested by using peer review method. In this case, a 0.9 beta version was distributed to different individuals in which they had use the system to it peaks in different ways and then they reported issues they faced when using it and also areas where the system needs improvement. One major security issue reported during this test was Cross Site Scripting (XSS) in the comments section of which this security loop hole was patched and then a 1.1 version of the system was released.

Evaluation
The project was evaluated using the following objectives defined in the introduction section:
- To design and develop a user-centered content management system without content restriction.
- To design a secured, user-friendly content delivery application.

The project was analyzed on the basis of specified objectives.

User Centered Content Management System
"To design and develop a user-centered content management system without content restriction."
The project has been developed with a framework that uses the architectural structure of Model, view and controller to ensure interoperability and robustness when being handled by users.
The system uses model to communicate with data from users and interprets them using the controller and gives an output through the view. It is user-centered because issues' involving accessibility and content level has been eliminated. Content sections was clearly defined, user can easily browse through contents by simply searching, navigational elements are user-defined so there is no problems with navigations, scrolling through a lot of contents has been eliminated as there is pagination for easy browsing.

User-friendly Content Delivery Application
"Design a secured, user-friendly content delivery application".
The major challenge faced during the project is conceptualizing an innovative, secured and user-friendly CMS that can leverage the challenges inherent in the previous global releases. Inadequate time was a limiting factor in the design and development of the system since it is thesis based project. The project implementation was quite labour intensive. Several components APIs, libraries and architectures were utilized to ensure the system flexibility. The flexibility of the system has been illustrated in the 'architectural design and development' of the system. The deployed system demonstrates the operation of these components and how robust they are. Although the system is very simple, most features promised were achieved which makes it very unique.

Conclusion and Further Work
Key Accomplishments
The VlogMS was able to accomplish the following achievements:
- A framework for video blogging activities
- A responsive user interface for handheld devices
- A user-friendly content delivery application.
- A secured system.

Future Work
This project was the first to fully implement video blogging on Content Management System. Future improvement would be endless. This section covers possible areas that are likely to be implemented in future for best effectiveness and efficiency.

The following are list of possible improvement:

- Improvement of the "Theme/Template Changer".
- Enhancement on the Search Engine Optimization (SEO) Configuration.
- Better plugin system
- Advanced Navigation Manager
- Cleaner coding structure et cetera

Conclusion

Content Management System (CMS) as an open source web tool has revolutionized the internet world. However, VlogMS has never been invented. This project was the first to implement a system of this nature. The system however fulfilled the intended objectives. Also, the artefact has reached a state where it can be distributed for end user's consumption. The system operates with a good level of efficiency and its advantages have been fully outlined.

References

1. Andrew FL. A Review of HTTP Live Streaming, 2010.
2. Arpit D, Satinderpaul SD. Implementation of Streaming Audio and Video for Online Web Based Courses Using Mysql Database Server and Php, 2000.
3. Boag P. 10 Things To Consider When Choosing The Perfect CMS. SMASHING MAGAZINE. Archived from the original on 2009-05-05. Retrieved2014-07-07, 2009.
4. Bobby K, Luke G. HTTP Live Streaming as a Secure Streaming Method, 2012.
5. Brian R. Attacks on open source call for better software Design, GCN, 2014.
6. Dejan V, Igor L, Ana B. Comparative Study Based on Open Source Content Management Systems Mambo and his Fork – Joomla and Elxis, 2013.
7. Drupal Update Manager and update status. drupal.org, 2011a.
8. Drupal Drupal security team. drupal.org, 2011b
9. Drupal Drupal security RSS feed. drupal.org, 2011c.
10. Drupal PHPTemplate theme engine. drupal.org, 2009a.
11. Drupal XTemplate theme engine. drupal.org, 2009b.
12. Drupal Security advisories. drupal.org, 2009c.
13. Drupal Contributed modules. drupal.org, 2009d.
14. Drupal How does Drupal compare to Mambo? Discussion thread. drupal.org, 2005.
15. Educause 7 things you should know about YouTube, 2006.
16. Franc K. Media Streaming Over the Internet, 2002.
17. Francis US. Moving Media Storage Technologies: Applications & Workflows for Video and Media Server Platforms, 2011. P. 381.
18. Gehani N. The Database Book: Principles and practice using MySQL. 1st ed., Summit NJ, Silicon Press, 2006.