TO: *Dr. Green*

FROM: *Elijah Rose (elirose)*

DATE: *7.19.2019*

SUBJECT: *CocoaRose CMS Web Application*

SUMMARY

Content Management Systems (CMS) are an increasingly popular tool for developing websites, handling the more technical aspects of website management leaving almost only content generation and design layout to the creator. Wordpress, one of the most popular CMS tools, (WordPress).

However, CMS's are still coded websites with HTML as their skeleton and CSS, JavaScript, and some server-side language at their core (alongside the various frameworks, plugins, libraries, and preprocessors associated with these languages). In order to better understand these systems, a rudimentary CMS blogging site will be built from the ground-up and with minimal frameworks. The following pertain to the intent of this CMS:

1. Sign-In/User System: Site visitors shall be able to log-in to have varying levels of privileges, based on a server-side userlist.
   a. Naturally, users shall be able to create a userprofile that is password secured.
   b. Users should be able to stay logged into their same profile throughout at least a session, if not between sessions, through persistent memory techniques.
2. Users of the requisite privilege-level shall be able to commit content – archive posts and pictures – as well as edit and/or delete content, which shall affect the site itself and be visible by other viewers.
   a. Users should be able to retrieve, edit, and commit posts in markdown format, or perhaps a designated derivative, which should then be converted into HTML for page rendering.
3. Users of the requisite privilege-level should be able to commit, edit, and delete comments pertaining to a post.
4. User-Generated content should be properly validated to ensure that inserted data shall not be harmful to the system overall.
5. The site should be accessible by most users, as assessed by the WCAG 2.1 Guidelines, at least to Level A.

While it is unlikely that this CMS will amount to much, or even be used by anyone for any purpose other than testing, it should serve as an example of the core features in almost any CMS or blogging site.

PROJECT INTRODUCTION

In honor of a friend and for lack of ability for creative names, the site is titled as "CocoaRose". The site serves primarily to create a rudimentary blogging site with CMS tools, such that a non-technical user can effectively add, edit, and remove content without viewing any source code or even necessarily being on the same machine as the server.

As this is a learning project, minimal frameworks and libraries will be used. JQuery, for instance, will be used as I understand the relation between some of its tools and the equivalent JavaScript, but PHP's Laravel framework near trivializes some elements such as a log-in user

system, hence while it is quite useful, it defeats one of the primary purposes of the project – to create a site from near scratch.

The project began in early June and has since undergone several revisions. The project and its goals can be found on GitHub (Seraphendipity, 2019), although as of yet the commits have largely been major file re-arrangement and large edits rather than the desirable "atomic commits" (Patterson, 2013).

Naturally, as this project developed alongside my knowledge/skills with web development, it began simply as HTML pages. CSS was also used extesnively to "pretty up" the pages. A large focus was placed on creating semantic HTML, removing styling and scripting from content pages. Later, even content was removed from the HTML pages, being moved to initially text files. In the initial versions, PHP simply streamed the HTML content and added tags in a very rudimentary markdown fashion, however after the discovery of the original (Gruber, 2004), the content is now stored in this fashion, being converted to HTML via simple batch scripting.

That aside, the basic layout and structure of the site is complete. The main pages being the homepage (a sampling of the site), archive pages (both the viewing of all posts a la card-style and the posts themselves), photos page (allowing the viewing of photos), and a contact form page (complete with form-handling). There remains a good-bit to be done, from bugs (text fade-in/fade-out not working as intended) to optimizations (archive reads require multiple file opening and closings, needs to be stored in a database or maybe large JSON) to security (form validation, client and server side) to features (displaying meta information of posts, editing and committing posts, uploading images, user system, etc.) A more detailed list of ideas and steps can be found on the aforementioned GitHub project (Seraphednipity, 2019).

I am not entirely sure how much can be realistically done. I have a quasi-decent idea on how the features will work and how they will be implemented. In order to create the core features, though, I may need to take shortcuts in a few implementations. For example, I am leaving image pop-ups to a JQuery extension/plug-in and may simplify the user/password system without as much security as may be properly needed. Additionally, though the styling will preferably be done in SASS/SCSS to create cleaner styling and allow separate pages of CSS (recall that you do not want to use raw CSS @import much (Avoid Using CSS Import)), I may need to leave it at the sort of messy CSS that it is now.

REFERENCES

Avoid Using CSS Import. (n.d.). Retrieved from Gift of Speed:
        https://www.giftofspeed.com/avoid-using-css-import/
Gruber, J. (2004, December 17). *Markdown*. Retrieved from Daring Fireball:
        https://daringfireball.net/projects/markdown/
Patterson, S. (2013, October 4). *Atomic Commits*. Retrieved from Fresh Consulting:
        https://www.freshconsulting.com/atomic-commits/
Seraphednipity. (2019, June). *CocoaRose Projects*. Retrieved from GitHub:
        https://github.com/Seraphendipity/CocoaRose/projects/1
Seraphendipity. (2019, June 21). *CocoaRose*. Retrieved from GitHub:
        https://github.com/Seraphendipity/CocoaRose
WordPress. (n.d.). Retrieved from WordPress: https://wordpress.com/