Wooster CMS User Guide

By Austin Elliott

http://woostercms.com

Introduction

Wooster CMS is a basic Content Management System (CMS) developed as an independent study project at Eastern Michigan University. It provides basic CMS functionality such as managing tags, posts, and admins. This document serves as a user guide and project report and provides a thorough overview of the project and how to use the site.

The Public View

Anyone can access woostercms.com and view the posts and categories that are visible to the public. Public access only includes viewing the list of posts and tags and viewing the content of each post and tag.

Posts displayed on the homepage and on the categories page are sorted by weight, so posts with a higher weight will appear at the top, and posts with lower weights will appear below that. This allows the post author to decide how important the article is, and where it should be displayed on the site. Breaking news, for example, should have the highest weight of any article while it is still breaking news.

To view posts by tag, click on “View Categories” on the homepage. You can then click on any tag and see all posts that belong to that tag.

Logging in and Site Overview

To utilize the CMS’ full functionality, you must be logged in. All write access to the database requires user authentication. **To take a tour of the site’s full functionality, use the following account formation:**

Email: test@test.com

Password: test123

Once you are logged in, you will have access to new links on the navbar, including Tags, Posts, and Admins. None of these pages are viewable if you are not logged in.

Tags

Tags allow the posts to be categorized and organized on the site. To view available tags, click on the Tags tab. You have the ability to view, edit, and delete existing tags, as well as create a new tag.

Creating a new tag only requires a name and description. This will be public on the site. The Tag ID is not public information but is required for tagging a post (more info on this later).

Posts

Similar to tags, you can view all posts and are able to create, edit, delete, and view posts. Unlike the homepage, the posts page displays posts by most recently updated. This is to allow easier access to relevant posts.

Posts require a title, subtitle, author, tag, weight, and a value for visible, as well as the post's content. Posts with a higher weight appear before posts with a lower weight on the homepage. To add a tag to the post, you must enter the **Tag ID**, not the tag's name. It should be an integer (1, 2, 3, etc.). To see available tag ID's, view the tags page. Visible should be either 0 or 1. If 0 is entered, the post will not be visible to the public, but is still visible to admins on the posts page. If a 1 is entered, the post is visible both to admins, and to the public on the homepage.

Admins

Admins are able to view and create admin accounts on the admins page. Admin accounts include an ID, first name, last name, username, email, and password. Usernames must be unique. Because access to the CMS is limited, members of the public are not able to create an account and existing admins must create new accounts.

About the Project, Challenges, and Additional Features

Before starting this project, I completed two web development courses on LinkedIn Learning: PHP with MySQL Essential Training 1 & 2. This allowed me to brush up on my web development skills and learn how to securely implement a user login system. I spent about one month completing these courses, from Feb. – March 2022, and began implementing the project in March 2022. The project was completed in it's current form on April 24, 2022.

Site Security

Proper handling of user data is essential on today's web, and to this end, this project utilizes the BCRYPT algorithm that is built-in to PHP to hash and store passwords. BCRYPT is a slow and one-way algorithm that also salts every password. Because the algorithm is one-way, there is currently no way to unencrypt the passwords. Instead, when a user attempts to login, their attempted password is hashed and is compared against the hashed password stored in the database. This way, passwords are never stored or viewable as plaintext.

Protection against SQL injection is also crucial for any site on the internet today. One way to achieve this is to use the mysqli_real_escape_string() function in PHP. Any user-entered value that is entered into the database is passed through this function, which escapes any reserved characters in SQL. This way, when the SQL statement is executed, any malicious code entered by the user won't be executed.

Similarly, whenever something from the webpage is displayed on the page, it is first passed through htmlspecialchars() to escape any reserved HTML symbols that would otherwise be rendered to the page. urlencode() is also used for any values being passed to a URL as another layer of security.

Project Challenges

The first big challenge I encountered was styling. I don't have a good eye for aesthetics, and don't generally enjoy designing things, whether it's a website or a painting. But after watching some basic CSS tutorials and relying on examples of other websites, I was able to put together a somewhat organized layout that I was happy with.

Another challenge I encountered was getting the site running on a live web server. I had never gotten a site up and running live on the web, so it was an interesting opportunity to see how it's done. After sifting through docs and YouTube videos, I was able to troubleshoot common errors, such as properly connecting to the database being hosted on the server.

Additional Features

Due to time restrictions, I wasn't able to implement a couple of features I had originally intended to include. The first was a search bar that allows users and admins to search for articles or tags by name.

I also intended to include pictures in the CMS, allowing users to upload pictures and include them with articles. This is a feature I don't have any experience with, so I saved it to the end, and was unfortunately not able to complete it in time.

There are also some improvements I think I could make with the styling, and some other minor features like a back button, that I would like to add to refine the project more.