

Nick Petty

COP3813 HW 8 Report

Purpose

“WallPics” is a fully-developed photo sharing website, giving users the ability to create accounts and post photos with filters and descriptions. With homework 7 providing the user account components, this assignment adds the photo sharing wall and photo upload form. The primary function of this website is to let users share pictures. To access this feature, an account must be created and the user must sign in. From inside the app, pictures can be uploaded and the uploader’s name, picture’s title, and uploader’s comments are attached. These fields are limited to 16, 20, and 140 characters, respectively. Besides uploading their own pictures, users can view pictures uploaded by others. This is the picture wall, and thus the app’s name “WallPics.” The wall displays pictures and attached text, with the addition of the time the picture was uploaded. Functionality is limited to creating accounts, signing in and out, viewing the wall, and posting pictures. If more features are to be added, post editing and deleting, and account deleting would be the most necessary. Overall, this website provides basic photo sharing with a simple interface.

Design

The wall and photo upload forms use the Jumbotron navbar with the Bootstrap-theme buttons. Posts are contained inside panels with a title bar, and have two Bootstrap grid columns for pictures and text. Because the account management forms from homework 7 used the basic Bootstrap styles, the Bootstrap-theme stylesheet was added. This gives the website a simple, consistent color scheme – white page backgrounds (light gray for sign in/up), black text and navbar, gray boxes, and blue buttons. Bootstrap stylesheets also allow for mobile-responsive pages, which is another primary design focus for this app. User interaction is tightly controlled, as only one or two options are given at a time, but the site is capable of handling abnormal actions (done by navigating the browser directly to specific files) by checking session status and redirecting. To make posting more interesting, three different filters can be applied to pictures and posts must have titles and descriptions. Posts must also have a username attached to them, which in the homework description must be user input. I’m opposed to giving this capability, as it opens abuse possibilities, so I made it not obvious that the “Name” field in the upload form can be changed. Use case issues from homework 7 are still present; that is, there are mostly-empty pages that link account management actions. These problems were not fixed because doing so would require re-writing the form and business logic.

Development

Tools

- Sublime Text 2 for coding.
- PHP for server-side data processing.

- MySQL for database querying.
- HTML and CSS for webpage layout and styling.
- JavaScript and jQuery for DOM manipulation.
- Bootstrap for template, stylesheet, and JavaScript.
- Textbook chapters 12 and 26 for PHP code examples.
- Starter code provided with assignment for core functionality.
- Favicon-generator.org for favicon.
- CSS-Tricks for filter usage.
- Cyberduck for SFTP to LAMP server.
- GitHub and SourceTree for repository management.
- Safari (OSX and iOS), Chrome, Firefox for viewing and testing.
- Nu HTML Checker for validation (no errors or warnings found).
- Stackoverflow and class discussion page on Blackboard for troubleshooting.

I thought this assignment would go quickly, since it was mostly done in the previous homework, but I was wrong. The core functionality was complete in a few hours; I was able to run the provided code with my previously-built account management form almost immediately. However, meeting the filename and code organization requirements meant enormous effort refactoring files and methods. Fixing formatting issues also presented a major challenge, as the “wall” table needed to be modified and much of the provided HTML design had to be re-done.

Besides refactoring and editing code, two primary issues took up a lot of my development time – filters and logout. To implement filters, I initially expected the pictures to be saved with filters applied and that I would just need to find some code to do that. But I realized that by adding the “filter” column to the table, then adjusting the CSS on the wall, I could meet the assignment requirements with the desired functionality. At first I did this by saving filters as an integer value, but found that the code is much simpler if varchars are used. The logout action was behaving very strangely – a user would be logged out if they clicked the “Sign out” button, and a confirmation page was shown, but no text notification was being displayed. This was especially frustrating since the code was directly from homework 7, which did not have this issue. After several days of investigation and fixes, I ended up re-writing a large portion of the logout.php file and its associated business logic. The problem seemed to come from using header.php for session verification and HTML creation, so those functions were copied into logout.php.

Although this assignment was very frustrating, I am happy that I was able to add a few small features. The first is a nicely formatted timestamp for when posts were created, via the PHP date() method. The second new feature I added is the “invert” filter, from the guide on CSS-Tricks. The last feature is small fix for text input. Because text is sanitized before going into the database, apostrophes are escaped with a backslash, but the escape is not removed when read back onto the wall. To make apostrophes appear correctly, I used the PHP str_replace() method when reading text from the database, as recommended by one of my classmates.

As far as bugs, I found that uploading a picture on mobile will fail if the user chooses the “take picture” option, this is probably because the filename and location of this kind of photo is not the same as one from a local disk drive. Most issues with forms being re-posted on a page refresh were resolved using a redirect after submission, but the forms from homework 7 were not fixed with this.

I’d estimate that the total time for this assignment was around 20 hours, a lot longer than I had expected. I’m certainly better at web development, but there were a lot of issues that I never would have been able to resolve on my own, and I don’t completely understand some of the code I used. These last few assignments have been rather overwhelming, and I’m not very happy with my work. Unfortunately, I simply don’t have time to make them better without taking time away from more important responsibilities.