Nick Petty
COP3813 Final Project Report

Purpose

WallPics's overall usage has not changed from assignment 8, it is still a photo sharing website. The primary functionality of the app is to allow users to create accounts and post pictures to a common wall. This is managed through two subsections: the sign in/sign up forms, and the photo sharing wall, taken from assignments 7 and 8, respectively.

The sign in/sign up forms remain largely unchanged from assignment 7. They still allow a user to create an account and sign into the app with that account. New in this version is the addition of Ajax to give feedback on username availability when creating an account. As an account is being created, the user database is checked to determine whether the input user name is available. This instant feedback makes the site a little more user-friendly. What makes the site user-unfriendly is the confirmation pages, but those could not be removed due to time constraints. An additional confirmation page was added for when an account is deleted, and it keeps with the theme of the sign in/sign up pages.

The photo wall has most of the new features implemented in this assignment. With a redesigned navbar, users have a dropdown menu with the new "account" option. The account management page allows users to change their password or delete their account. Password changing is somewhat secure as passwords are not exposed and the old password is required. The completion page is still very barren, but with the new navbar, it's easy to get back to the content. Deleting an account is now possible, but only when signed in as the account holder. This will keep the posts submitted by that account, however, as only the user table in the database is changed.

Accessibility and posting rights have been changed from previous versions of the app. The photo wall can now be viewed when not signed in, but attempting to take any action other than passive viewing will direct back to the sign in/sign up page. Assignment 8 required the username to be an input when uploading a photo, which is a potential avenue of abuse. In this version, when posting a photo, the uploader is unable to change their username and it is automatically attached to the post. A delete post option was considered, but not implemented due to time constraints.

With new features and more open accessibility, this version of WallPics is most able to provide users with a clean, simple photo sharing service.

Design

As with all my apps, the primary design focus is simplicity. All features and pages are built to be easy to use and understand, with minimal clutter and consistent styling and layout. User interactions are limited to posting photos and managing an account, but those interactions are tightly controlled and error-resistant. Every action can be done with equal ease on either mobile or desktop browsers. The two main components of the website have slightly different design, so I will discuss each separately.

The sign in/sign up forms maintain the styling and use cases from the previous assignments. They have a light gray background with blue buttons, and there is no other navigation system. Although it is possible to reach invalid pages through the address bar, these cases are protected against through session verification and redirection. With the change in photo wall access, an extra button was added to bring visitors directly to wall. The newly added Ajax feature does not change the form layout, as it is placed in the same location as feedback in the previous assignments. The account deletion confirmation page was added in this assignment, and it follows the styling of the sign in/sign up pages. All of these pages are based on Bootstrap's form styling, and are mobile-responsive.

The photo wall pages have had significant redesigns applied. While previous versions of the app used an inverse navbar with blue buttons, this version has a Bootstrap standard light grey navbar with a dropdown menu. The navbar on all relevant pages has been changed to this new design as well. I have always preferred the inverse navbar because it stands out from the rest of the page, but after using this new color scheme, I like the lighter feel it gives the website. The dropdown was necessary because adding more buttons would create problems for mobile browsers. On mobile, the navbar was nearly filled already, so adding more buttons was not an option. The user's name now functions as the dropdown menu button, and the app's name has been added for page branding. With the dropdown, users have quick links to the rest of the website's content.

Only minor layout changes were applied to the photo upload form. All form elements were centered except the file selection input. I could not find a working method to center this element, so it was left as is. The option to input a user name for a post has been removed, since this was a requirement I was opposed to in the previous assignment. Posts now automatically have the name of the user who created them attached. The extra filter, invert, has been carried over from homework 8 as well.

This assignment adds an account management page as part of the photo wall group. Because it is only accessible to users who have signed in, it was not included as part of the sign in/sign up pages. It uses the same form styling as the photo wall and upload forms, again taken from Bootstrap. Like the newly arranged photo upload form, this form is also centered for easier use with mobile browsers. The delete account button is red to warn that it has important implications, and there is a functional description above it. Changing the password is straightforward, and the confirmation pages adhere to the styling of the account management page. Although the confirmation pages here provide no links to go back, the dropdown menu is accessible for navigation.

The design choices I made for this version of the app are limited by my time and programming skill. While adding more colors or layout changes may have been possible, I chose to focus on making each change work well with little complication. Use cases have been improved by allowing more access to the photo viewing section of the site, and the overall feel is lighter because of the reduced number of elements and colors.

Development

Tools

- Sublime Text 2 for coding.
- PHP for server-side data processing.
- MySQL for database management.
- 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.
- Favicon-generator.org for favicon.
- 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 began this website with all the files from the previous assignment copied into a new folder, but kept the same database and tables. Because this assignment and the previous assignment use the same table for posts, but have different folders for saved images, a broken link will appear in one version when a picture is posted in the other. This was fixed by updating homework 8 to save images in the project folder.

The first change made to differentiate this project from the previous assignment was the addition of Ajax to the sign up form. Before this, a user would not know if the name they chose was available until they submitted the form. With Ajax, the user is informed of name availability as they type. I used the sample code from chapter 26 of the textbook for this, but replaced references to the HTML Canvas with jQuery selectors. This was a section of JavaScript and a PHP file, which sent a query to the database each time a key was pressed. The code was functional, but threw errors because it was trying to set headers that were supposed to be set by the browser. After some research, I found the bad header functions and removed them.

The next thing to change was the navbar. I was quite happy with the inverse navbar and buttons I'd used for every other assignment, but with the amount of options I have now, this layout is no longer usable. From the Bootstrap components page, I switched to the default navbar with dropdown, and found it suited the website design very well. I had a lot of problems sizing the containers and getting them to work correctly on mobile and desktop, but eventually found the right combination of Bootstrap classes and CSS modifications.

Adding account management tools made me rethink a lot of my site's element placement. Specifically, I'd left a lot of things in their initially provided locations, which tended to be slightly to the left. With the account page, I decided to center my forms and buttons, which made the mobile and desktop site almost identical. From there, I revised my photo upload form to also have centered elements. Functionally, the account management page allows a user to change their password and delete their account. To change the password, the account creation code was modified to verify the old password, then save the new password over it. I was unable to use a Post/Redirect/Get system to prevent resubmissions on this form when the page is reloaded, but the business logic does stop the database from being re-updated. Account deletion was done by modifying the existing logout code to include a MySQL delete statement. The method is simple and doesn't have any effect on the photo wall.

With finals and projects in other classes, work, and family responsibilities, the time I could commit to this project was limited. Although it was assigned on November 16, I wasn't able to begin working on it until November 29th, and without the extended deadline, I would not have been able to do even this much. I had hoped to implement an administrator mode, or at least a way for users to delete their own posts, but I didn't have time to come up with an effective way to combine the JavaScript, PHP, and MySQL code. My first thought was to create a column in the users table that would indicate administrator rights, but then actually giving them an administration panel would require another page and complicate the functionality of the rest of the app. I decided instead to pursue a post deletion system, but didn't know enough PHP to manage the files on the server, and also could not rewrite my PHP/HTML code to add a button with that functionality. Eventually this administrative function was pared down the account administration page. I am, however, happy that everything works well and doesn't have any major vulnerabilities.

My work was split about 30% on the Ajax, 40% on redesign, and 30% on account administration. Even though the changes from the previous assignment are not enormous, I still spent about the same amount of time on this project as homeworks 7 and 8, about 15 hours. The main takeaway from this has been my understanding of Ajax, and how to use the Chrome developer tools to look at server requests and responses. I didn't learn much new with HTML, JavaScript, PHP, or MySQL, but I spend a lot of time reading up on Bootstrap's CSS. Specifically, there are a lot of ways to center elements in their containers, and Bootstrap provides classes for this. I think it's better to use existing the existing classes than writing my own, even if it’s just a few lines, and I did my best to find those classes in the API. And in the broader sense, that’s what the main thrust of this project was: reuse and extension of existing code to complete a task in limited time. I would like to have done more, but I think adding a few simple features that work well is better than adding a lot of advanced features that work poorly.