**Project 7: Membership**

**Remy Brandriff**

**7 November 2016**

**CS212**

**Project Overview**

**Purpose**

This project introduced MySQL to our web development, in the form of using databases and tables to add membership features to our websites. Membership is a common feature on websites that serves many purposes—protecting data, maintaining session data between visits, special privileges, etc.—by allowing users to register with the site and storing their information.

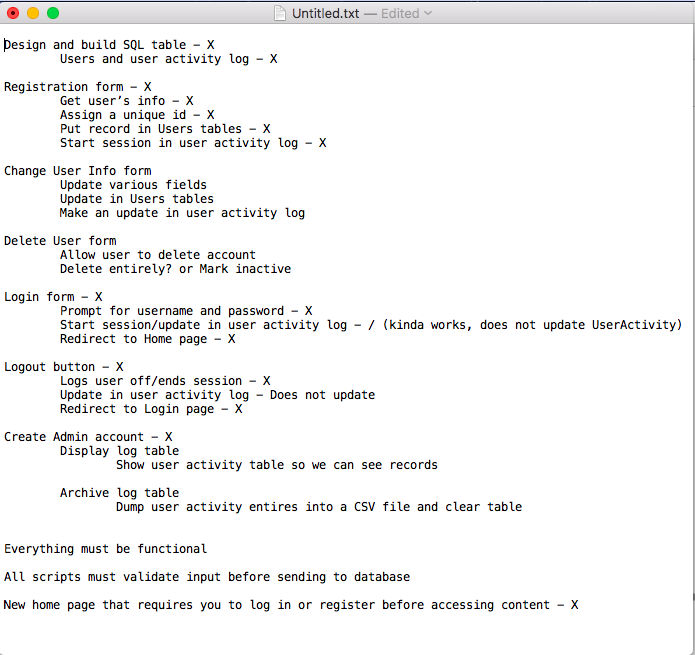
For this project, we needed to:

* Design and build MySQL tables for user records and user activity
* Create a registration form that gets user info, assigns a unique ID, and puts the record into the user table
* Create a form to change user information that allows users to change their account information and update it in the MySQL tables
* Allow users to delete their account
* Create a login form that prompts for a username and password, and redirects to the home page
* Create a logout function that ends a session and redirects to the login page
* Create an admin account
* Display the user activity table so we can see records
* Archive the user activity table into a CSV table
* Validate all scripts and forms
* Update all user activity in a table
* Create a new home page that requires the user to log in or register before accessing content

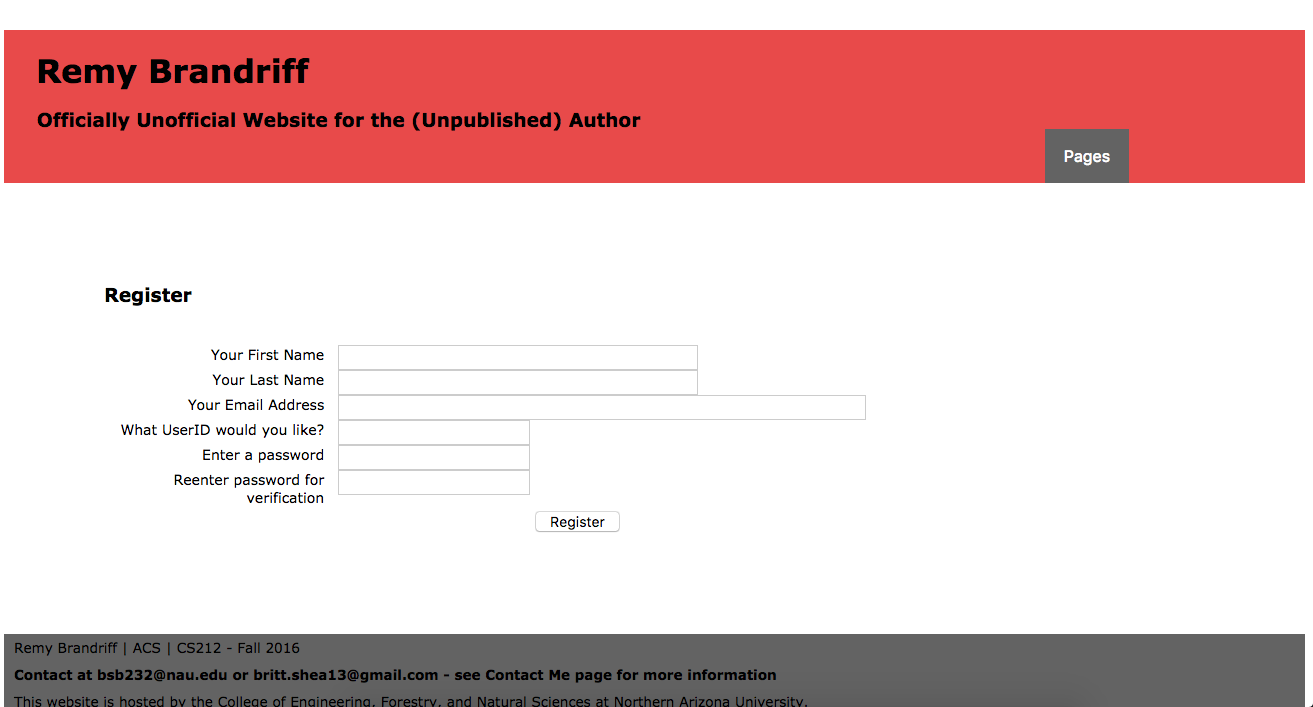
This was also an exercise in using MySQL with PHP, CSS, and HTML in a practical setting.

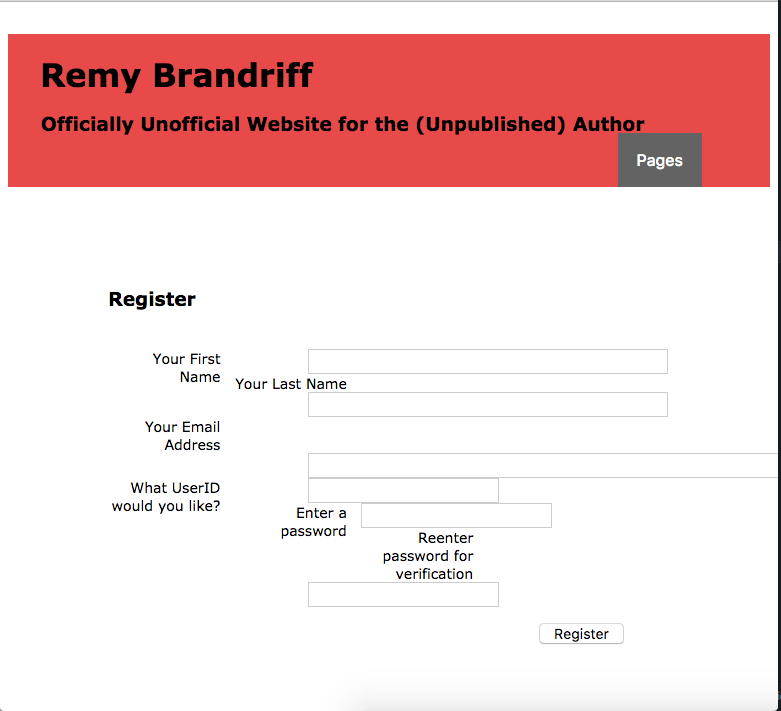
**Approach**

With the extensive requirements for this project, I made a list of everything I needed to do and what steps were necessary; I marked them off as I completed them, or updated the status of that task.

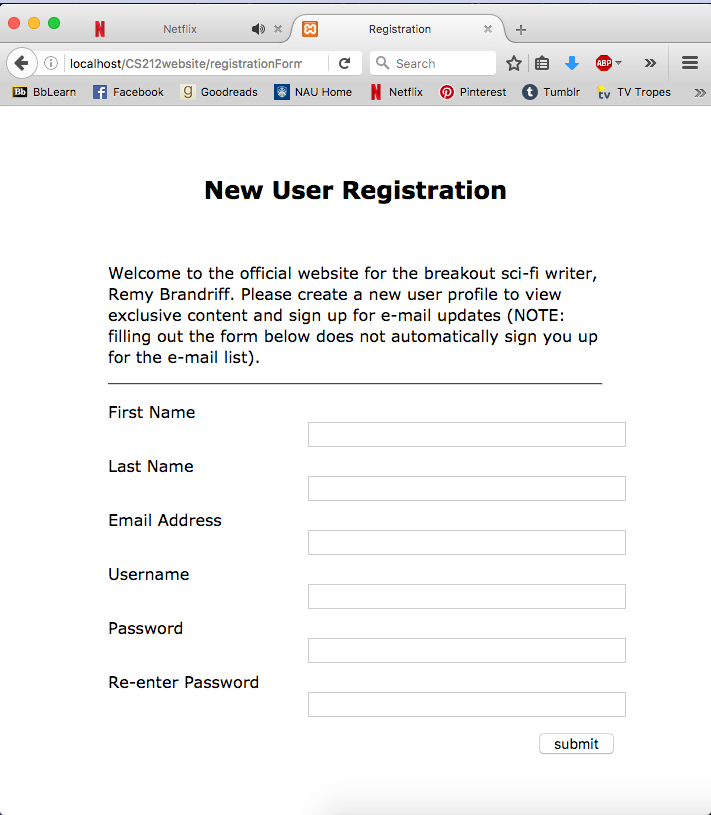


The first step was obviously registering users to the databases, and I relied heavily on the code provided to accomplish this. I took a while going through the code, trying to understand what was happening and how at every step, but I don’t think I ever fully understood it. The screenshot below was an early version of the registration form, before I fixed the formatting and added in validation.

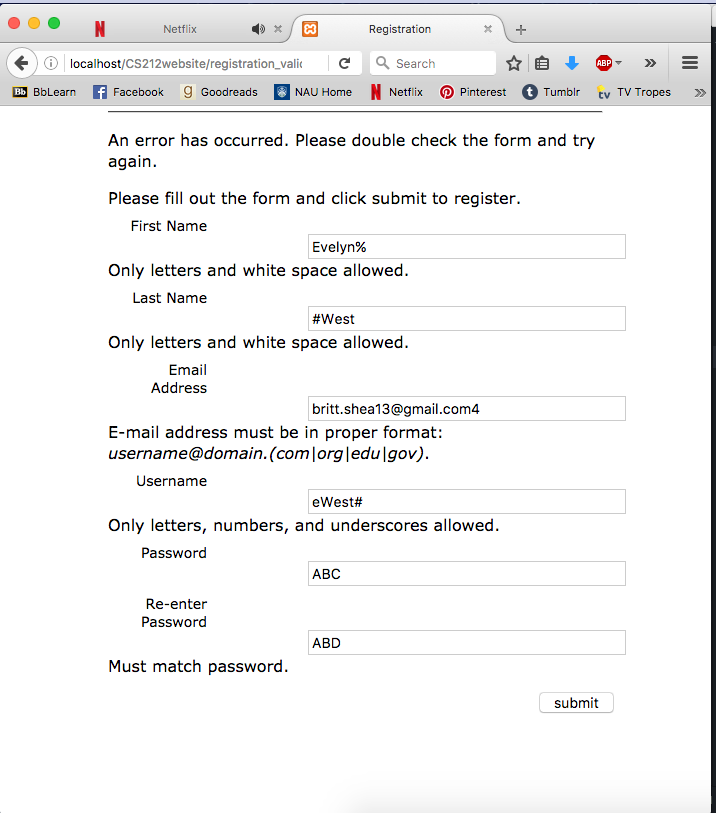




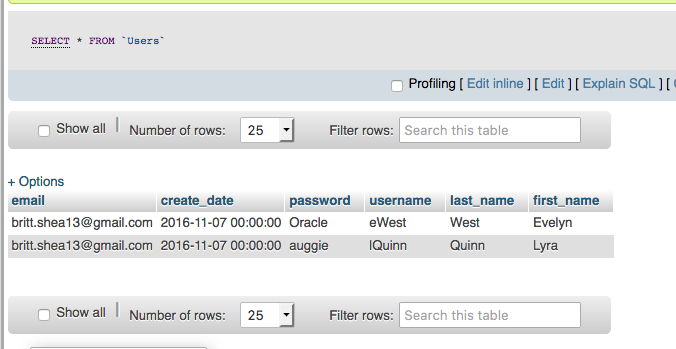
There was some weird formatting before I finally got that worked out, where the form warped when I changed the screen size. I later encountered this problem again when trying to

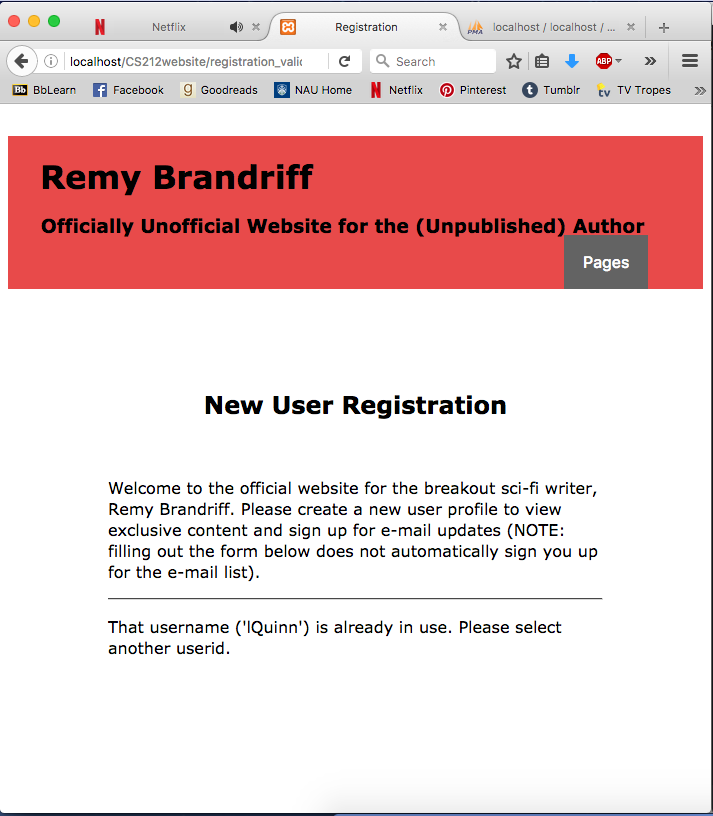


And the validation can be seen here, with a dummy user I created using a character from *The Eden Project.* Right after this, I changed the password fields from text boxes to password boxes; I used text boxes initially to test the accuracy of the validation. The previous projects involving the contact form were immensely helpful here.

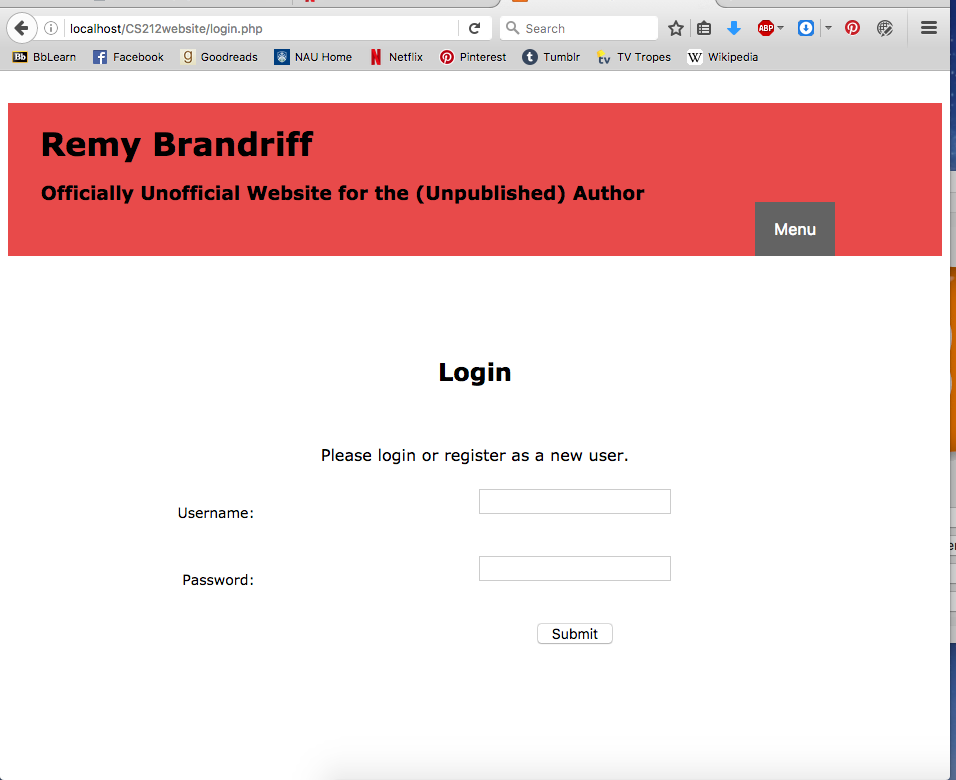


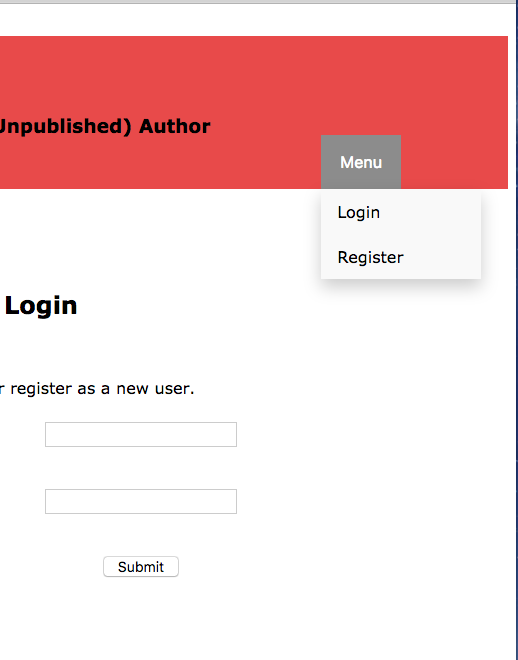
I also created a second user account, lQuinn (another character), and then tested how validation would handle my trying to use it again. I had difficulty updating the MySQL tables with signin and signout date/times in UserActivity, and the create\_date in Users. I realized while I was creating the tables for the CEFNS MySQL that I couldn’t edit signout in UserActivity because I accidentally created it as datetime instead of varchar(255).



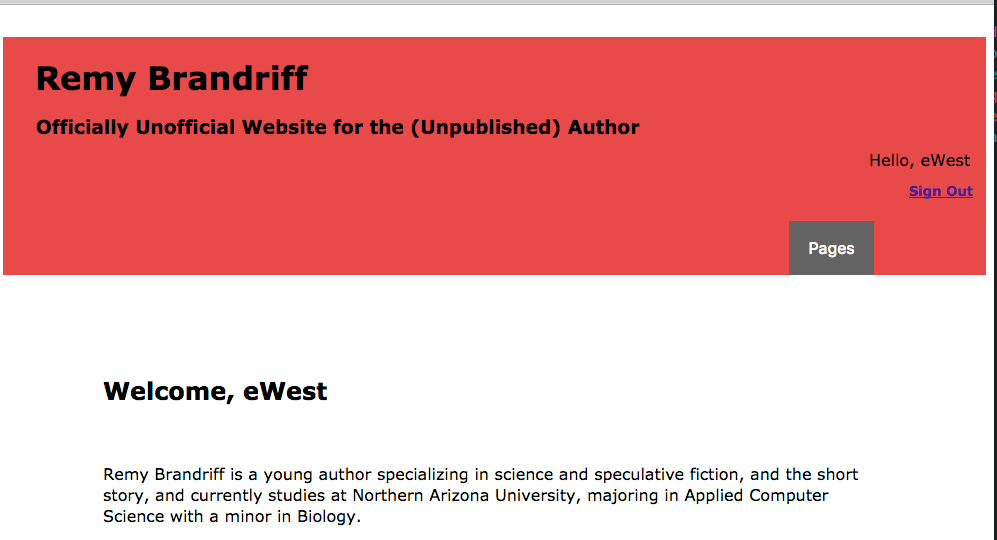


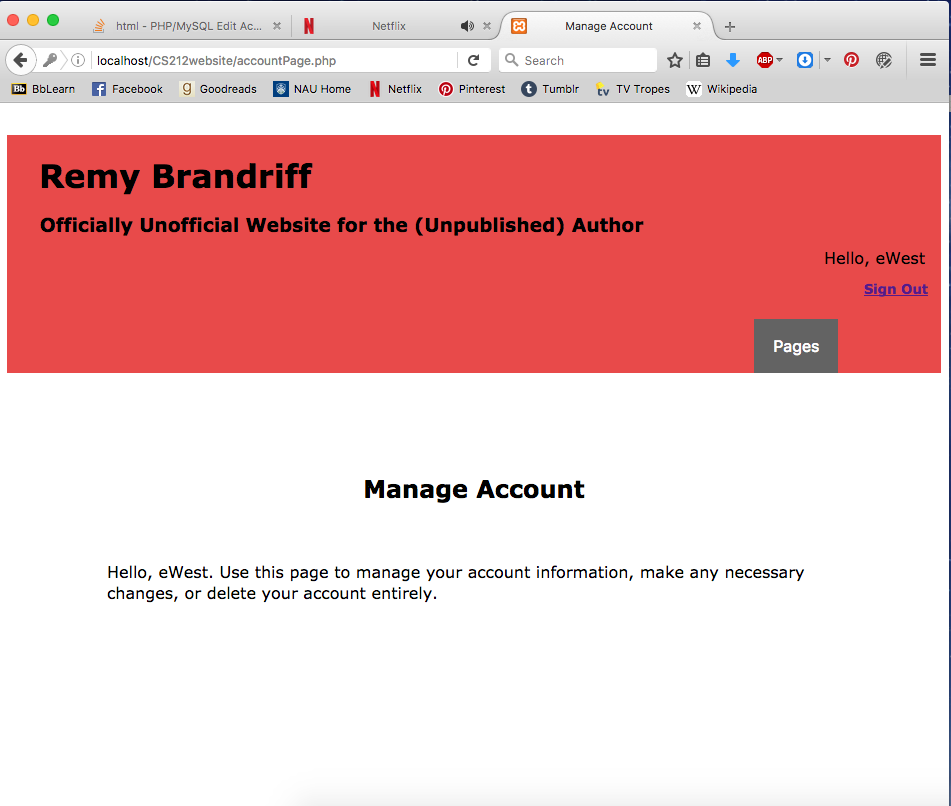
To create the login page and process logging out, I used [TutorialsPoint](https://www.tutorialspoint.com/php/php_mysql_login.htm) as a reference, especially to understand the MySQL side of it. I don’t know if I really understand now, or if I just understand the code I implemented, but the login works, and after I created that, I was able to force users back to the login page from anywhere on the site unless they were already in. Once I had login.php, logout.php, session.php, and config.php, I was able to redirect all other pages back to the login using *include(‘session.php’),* and I didn’t have to keep calling the database and creating a connection in each page because I could just *include(‘config.php’),* which was possibly the greatest discovery of this entire project.



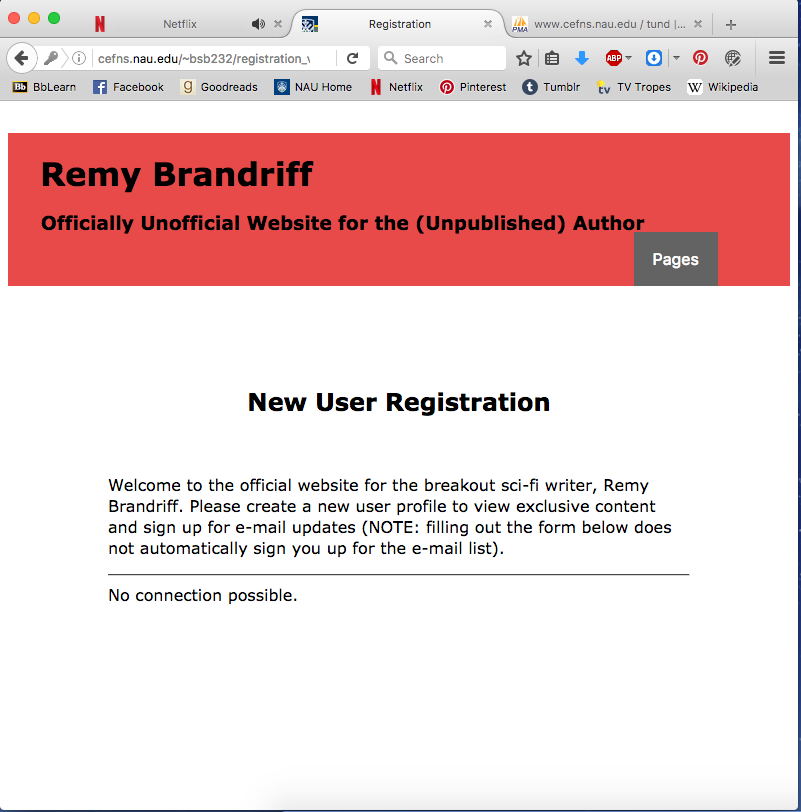


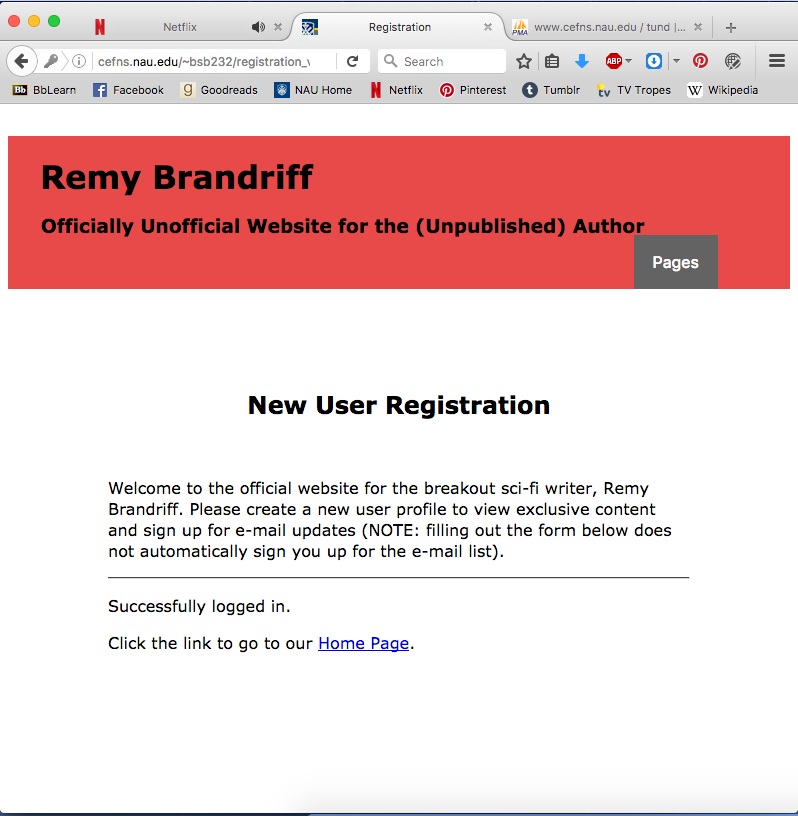
Once I had a user logged in, I decided to customize the website for them, even though it was only including their username in the header with the logout button; eventually, I’d like to turn the name into a hyperlink to the Account page, instead of having an option in the menu for it. At this point, though, the Account page doesn’t do anything (because I didn’t finish that part in time for the due date), so I haven’t fully completed that.

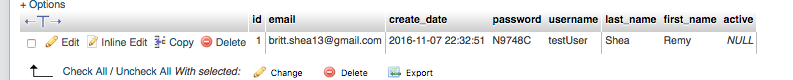




When it came to uploading the site to CEFNS and working with those MySQL tables, I had a little bit of trouble. At first, there was no problem—the login page showed up first instead of the home page, and I could only navigate there or the registration form. The registration form performed all validations correctly. My issue was connecting with the database; I realized that when I switched over the database to CEFNS in config.php, I forgot to change it in the original code we were given by the instructor for user registration. After I fixed that, I was golden.





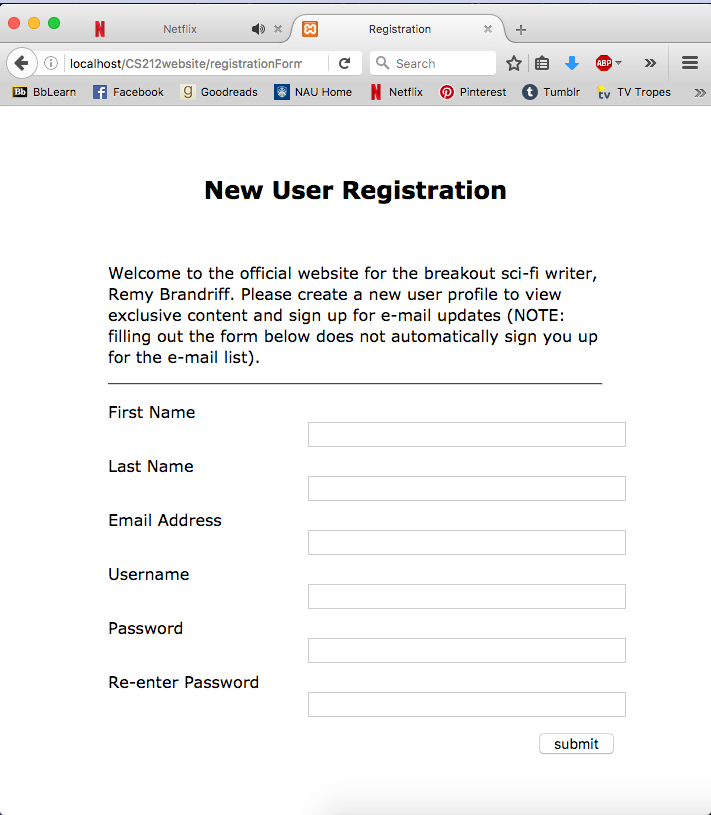




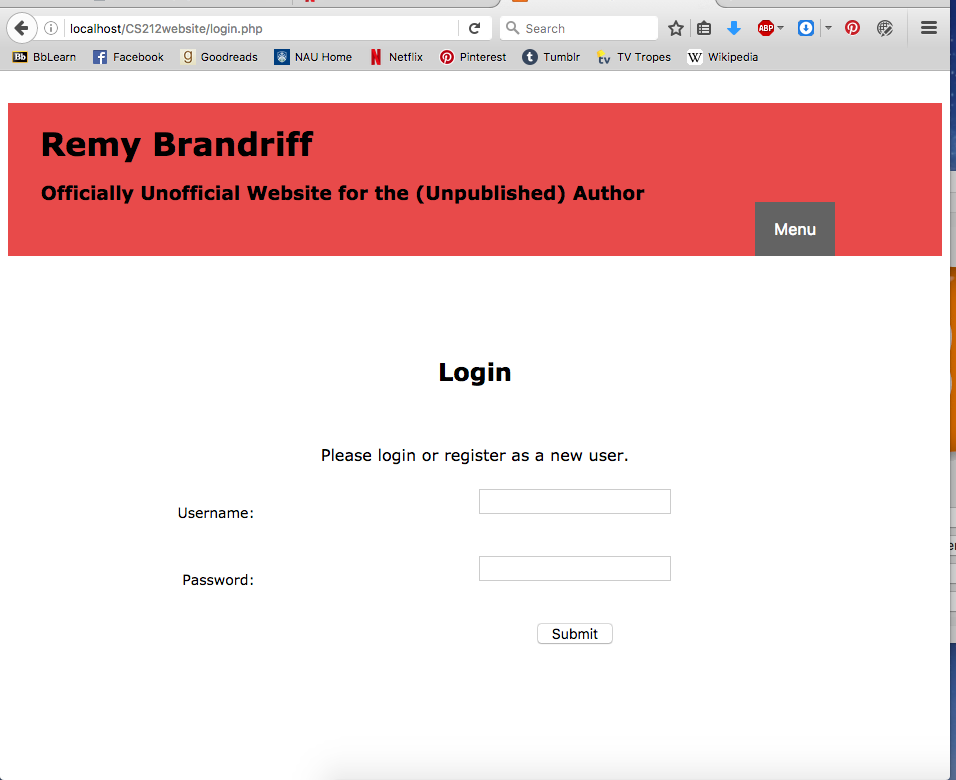
**Results**

I was not able to implement every requirement outlined in the project assignment, due to time constraints and a general lack of understanding of working with MySQL and tables. I’ve included screenshots of what I did accomplish.

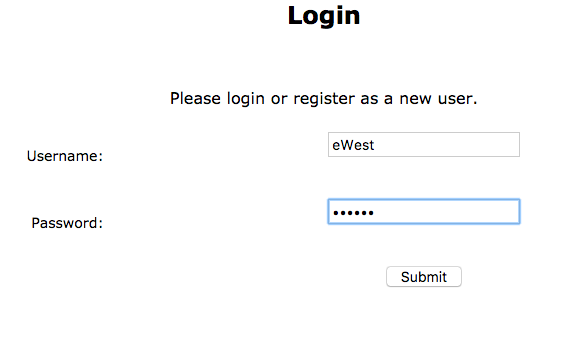
**Sample Output**



The new user registration form, in a minimized screen (like half the regular screen size)



The login page, the first thing people see when they try to get into the website



Also a close up on the login field, showing how it interprets user input for the eWest user, and showing the difference between the text and password fields

**Source Code**

The source files have been included as deliverables on BbLearn.

**Conclusions**

I’m proud of what I accomplished, given that it was my first time working with MySQL and databases like this. It was definitely a good experience, and I look forward to finishing the project after turning this in, because I’m not about to just leave this half-finished. But the main requirements for this project was being able to register users and log them in and out of the site, and regulate what they could see based on this status. I accomplished that—maybe not in the most elegant way, but it works.

There were multiple reasons I didn’t finish on time, but chief among them, I was focused on a project for CS200 for the first week, and then spent the second week trying to understand MySQL. But I’m pleased with what I did accomplish, and that I didn’t leave the unfinished pages open for the user to see, but gave them a reasonable page to see with a “This page is under construction” message. I got more done than I thought I would, and while I’m not happy with not finishing, I did what I could. I will continue to work on this and finish up the loose ends before Project 9.