**Implementation Details**

Starting with the log in and account creation page, the log in function is just two forms: one for username and one for password. Next to these forms is a button that will post these two forms to the login.php. Below are a similar two forms, one takes a username and the other a password, and the submit button will also post these forms to login.php. However, these forms are used for account creation rather than logging in.

Inside login.php, it first gets the variables that were posted from the forms in login.html. If either the username or password is left empty, then it will return to the login page and display an error telling the user not to leave the fields blank. If the user tries to log in properly, then it will search through the database for any matches to the username. After finding the username in the database, it checks to see if the corresponding password matches the one input by the user. If there is no match, then it will return the user to the login page with an error.

If the user is creating an account, a query is made to the database checking for the username. If that username is found, then it already exists and cannot be created again. The user is returned to the log in screen with an error. Otherwise, the new username and password are added to the database and can be used to log in.

Once the user successfully logs in, index.html is shown. This page simple has 2 hyperlinks on it. One to create a new page, and one to view existing pages. When creating a new page, create.html is opened. This page has multiple forms and buttons on it for use in creating a web page. This page is reminiscent of assignment 3. When an element is added to the page, custom html code will be appended onto a variable “page.” When the user presses the “create page” button at the bottom, the title of their page, the “page” variable containing html code, and their username will be inserted into a table in the database.

Attempting to view existing pages will open up list.php. This page connects to the database and fetches values from the tables storing all the content, which are the user-created pages. Each page is listed by its title and the user that created it. The title is actually a hyperlink that opens up view.php and sends the custom html text to it from the database. View.php simply displays the page based on the html code that is sent to it.

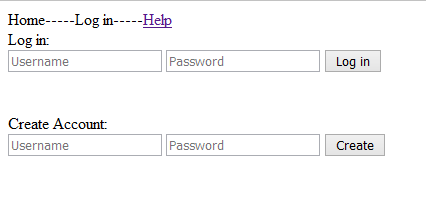
**Future Work**

One thing that I would really like to add to this project is *a lot* of CSS. I’m unhappy with the dull, basic look of it, but I am not comfortable with create CSS right now. I thought that it would look worse if I added some custom CSS with no real idea what I was doing. I would much rather learn CSS very well and make it look pretty, so for now it’s just plain.

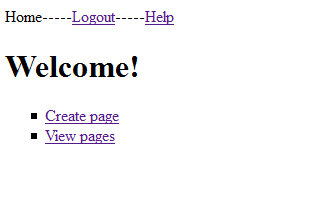
I would also like to add the functionality of editing or updating existing pages, rather than just creating a new one every time something needs changed. I thought about making multiple tables to store every element on a page by type, such as a table for headers, another table for sub headers, another for paragraphs, etc. This way a user could edit a particular paragraph without affecting the rest of the page.

**Screen Shots**

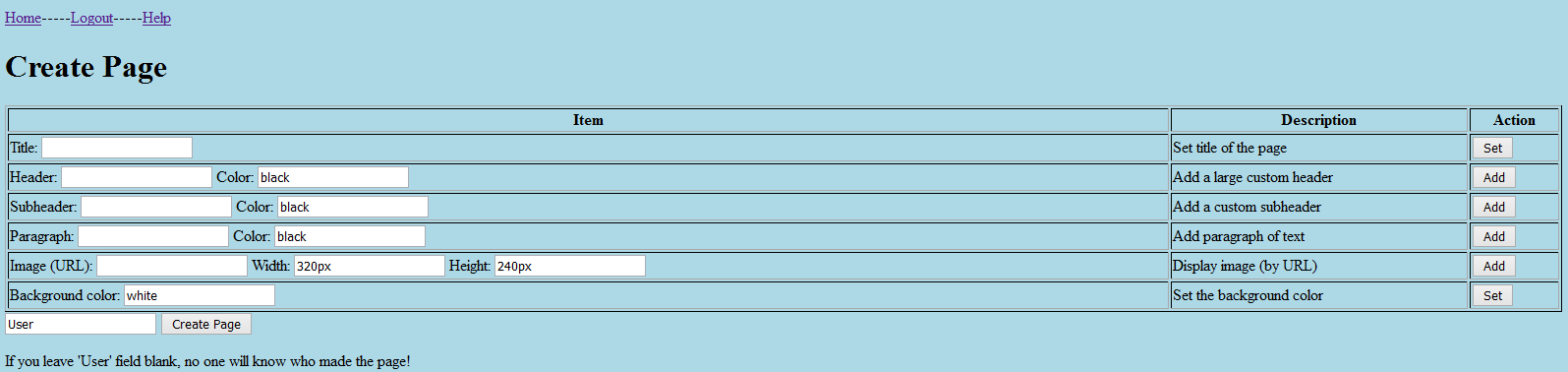
Login.html



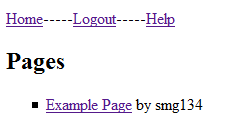
Index.html



Create.html



List.php



View.php (the custom page that is opened)



Contributions: Sean Gaffney – I did everything because I worked alone