

Tyler Moak

COP3813

Intro to Internet Computing

The seventh and final project asked to create a website where users may upload images for other users to view. This was a very complex app that I designed from the ground up and I spend many hours and tears developing it. Some added features I never got to implement due to time constraints. I learned many things from this project and I'd like to share some of my experiences, frustrations, and moments of success. This is IMPORTANT: there are ONLY 2 PAGES the index.php and the wall.php. The wall.php allows to see posts as well as upload photos.

My project is organized into several key files: index.php, p7.php, session.php, wall.php, upload2.php, and logout.php. Logging in and registering happens on both the index.php and p7.php pages. Index.php is simply HTML markup with added session checking while p7.php does all the grunt work for most of this application. All calls to the server happen via client side AJAX scripts which eventually hit a method like this in php

```
if (is_ajax()) {  
    if(isset($_POST['signup']) && !empty($_POST['username']) &&  
!empty($_POST['password'])) {  
        echo registerUser($_POST['username'], $_POST['password']);  
    }  
    if(isset($_POST['login']) && !empty($_POST['username']) &&  
!empty($_POST['password'])) {  
        echo loginUser($_POST['username'], $_POST['password']);  
    }  
    ...  
}
```

This will check if the request is AJAX and that the specified url params are correct then it calls the necessary function to do the request. Separation of concerns is a major piece of my application. I want everything neat with high cohesion and low coupling.

The hardest part of this application is not so much the logic but the time it took to complete each milestone. Designing the database was the easy part, learning sessions took a bit to understand and implement but I was far from being difficult, but the hardest thing was figuring out uploading documents and integrating that into the wall page. I used an external library to handle uploading documents but this was only client side as I did the server side myself with the help of W3Schools.com This is the library I used for file input:

<http://plugins.krajee.com/file-input>

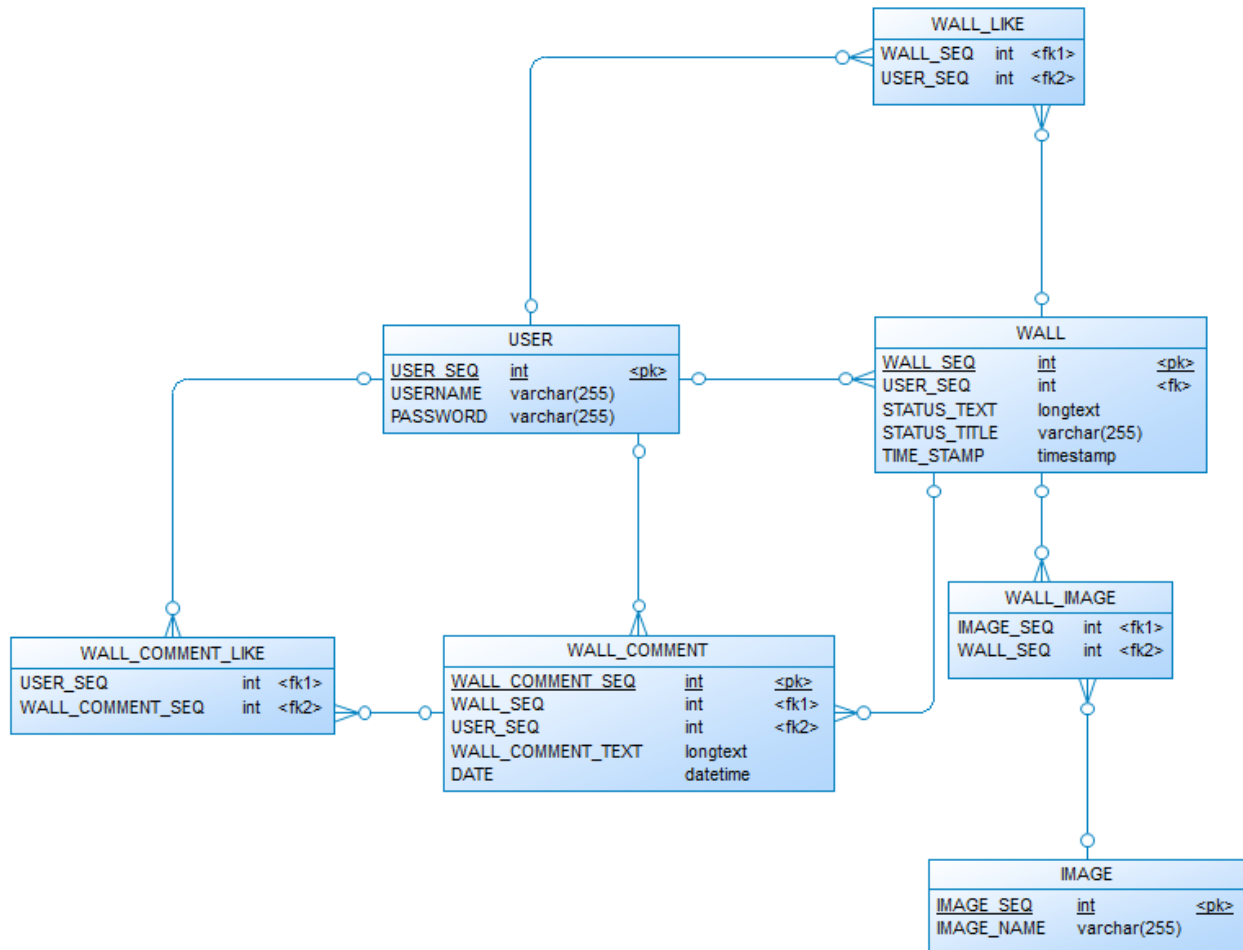
One thing to note is that I had to handle uploading files in a strange way. I created a temp folder for all uploaded pictures that would be moved to the uploads folder once the user actually posts the “status” This was due to multiple reasons that would require too much explaining to fully comprehend.

This is how I did my password salting and hashing for logging in and signing up:

```
$insertStmt= "INSERT INTO USER (USERNAME, PASSWORD) VALUES ('".$username."',  
            '".password_hash($password, PASSWORD_DEFAULT)."'");  
  
$selectStmt= "SELECT USER_SEQ, PASSWORD FROM USER WHERE USERNAME= '".$username."'";  
  
...  
  
if(password_verify($password, $row["PASSWORD"])){
```

There are some things I wish I could have done but just didn't have enough time to do such as editing posts, deleting posts, posting replies to posts, unliking posts as well as liking and unliking replies. I have everything set up in the database but it was just too much work to do in such a short time.

This is my database: Only the USER, WALL, WALL\_IMAGE, IMAGE, and WALL\_LIKE got implemented.



I used brackets for designing my site. I also used two Brackets extensions that let me code faster and more efficient: <https://github.com/dingdong-io/brackets.beautify.io>, <http://www.oslibrary.in/other-information-code/brackets-php-syntax-hint>. I really like the live preview option it has. Also its minimal interface is just right for me. User extensions were also a nice feature. I'm utilizing a github extension for brackets to have source control. I used Firefox for the majority of my testing as I could see how my webpage can be responsive on different screens.