Mark Wency P. Yambao
2012-76554
CMSC 198 AR-9

Introduction

I worked for Communication and Information Technology (COMMIT), a division of the UPLB Office of Student Affairs (OSA). Their key focus is to maintain the UPLB OSA website (http://www.uplbosa.org) and help students who need assistance in their clearance, student assistantship, etc. They also provide technical support to the other divisions of OSA and keep the Information Technology Hub in top shape.

OSA COMMIT is mostly composed of graduates from UPLB and has few student assistants. Everyone is professional even though they are friendly by nature. They meet after work to play bowling or eat dinner in which they invite me. Generally, it is a nice working environment. OSA conduct events where all divisions gather to eat and socialize.

My responsibility in OSA COMMIT is to develop computer systems or system components, design and maintain office databases, and provide technical support to students and staff. My primary focus in COMMIT is to develop a new design for the OSA website and improve some existing functionalities. COMMIT required me to use Materialize Framework and Fat-Free Framework. The project was done mostly in JavaScript and PHP (since Fat-Free is a PHP framework).

Methodology

The first task given to me is to create a web application for a job fair event. It takes information from the attendees and saves it in the database. At my first try, I programmed in php and JavaScript to connect and save information to the database. I suggested if we can use Firebase for the data storage but since there is no internet in the venue of the event, I used WAMP for the web server and phpMyAdmin for the database instead. After the initial trial, I created a Fat-Free Framework and MVC version of the web application for the job fair event. I used the F3 Framework, AJAX, and JSON to connect and save information to the database.

After finishing the attendance web application, I was tasked to create a program that deletes "malicious" files from a USB when said device is plugged. File extensions such as .sh, .ini, .dll, .bat should be removed from the device. At first, I programmed in C where there is a function that recursively checks the extensions of files in a directory and its subdirectories. It then removes files that is .sh, .bat, .ini and .dll and reports how many files were deleted. I utilized dirent.h. However, the C program I created was a dead end since it was hard to get the path file from a newly inserted USB disk. I then used batch file to constantly check if a removable drive exist, if yes then delete all files with the extension .ini, .bat, .dll, and .sh. The program traverses all directories recursively so it can find all files. If there is no removable drive, repeat the batch file. The batch file is hidden and can only be deleted through the task manager.

My next task is to create the new layout for the OSA website. My primary goal is to make it as responsive as possible and utilize Materialize Framework. After a few revisions from the comments of my supervisor, I finally came up with a good design. There is also a page where a form is located. The user can input information to the form and save it to the database.

After creating the layout, I need to incorporate Google authentication so user can log in on the OSA website I created. User can only use up.edu.ph. If other account is used (like @gmail.com), prompt the user then automatically logout. After a successful log in using a valid account, a session is created and the Google id is searched in the database. If the Google id does not exist, redirect to update information page. User needs to fill up the information form and can submit the form which then saves it to the database (also saves the Google id). If a user already has an information in the database, the update information page is preloaded with the user's information. User can also log out which clears the session. I used the Google API and their JavaScript functions for this task.

After implementing the Google login, I also included a login system where the user can create an account by logging in using his/her up.edu.ph Google account then later on use the username and password of the OSA account for logging in instead of the Google account (user can choose to log in using his/her OSA username or Google email to access his/her account). Inserted a change password input form in the account information page. Password input form will change if the user is new or not. If new, then username and password is required. Else, user can opt not to change his/her username or password. Also made a function to check if a username provided by a user (either new or not) already exists. For the password, I used MD5 and salting.

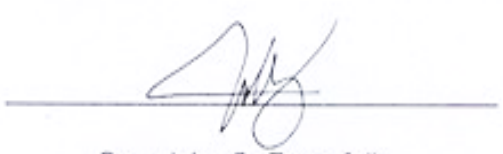
My sixth task is to create a function in JavaScript that will check if a user's given Facebook ID or username in the account information form is valid. User won't need to login in Facebook, just verify if the id or username is existing. I used graph.facebook.com to check if an ID belongs to a Facebook user. For the username, I used PHP to visit the given username's Facebook page then used get_contents and looked for the page's title, then check if the page title is not "Page Not Found".

I also implemented an image uploader just like in the OSA website. Image is chosen by the user then saved on the local directory. The file name is randomized and file is checked if it has a valid extension. It also saves the path file to the database.

My final task is to create an admin account and admin page. I inserted a new table for admin. I also created a login system for use of the admin only. The login system is located on a page for the admin (restricted access for normal users). The purpose of the admin page is to approve valid pictures uploaded by the user. I successfully created a system where an admin can approve or reject an uploaded picture. The user can only have one pending picture. If the user uploaded another picture, the old is overwritten by the new. The uploaded file name is randomized for security purposes. If approved, the uploader is notified that the picture is approved and the picture is set as the user's profile picture. If reject, the picture is moved to the rejected folder. Previously rejected pictures are loaded on the admin approval page.

Conclusion

I can say the most important thing I learned is to listen to other's constructive criticisms. It is usually hard for me to take criticisms but I managed to listen and apply suggestions that made what I'm doing better and efficient. Another thing I learned is to never depend on others. We weren't spoon-fed here in COMMIT so I had to research and learn concepts new to me. It was really effective because at the end of the day, I managed to learn things I know I can use in the future. Last but not the least, I learned how to speak up. Voicing out my thoughts seems scary at first, but I managed to learn to say my own opinion and suggestions. Overall, working here in OSA COMMIT is an enriching and an experience I'll never forget.



Czar John G. Demafeliz
Supervisor