Sofia Bautista
301292776
IAT 352 (D101)
04/04/2017

P03 Technical Report

This report should provide a report on the whole project. Consider feedback your TA gave you on the prior assignments and prepare the report describing all parts of the system.

Project Summary:

During the final stages of development, I observed that my web application *Cerulean* decreased in scope to focus on meeting the requirements of each iteration of the project, especially during the third and final one. In my project proposal, I initially had a messaging system, a way for students to view grades and teachers to generate reports, but in the end I had to compromise those features since they were superfluous and above the expectations of this course, so instead, I turned my attention to fully understanding and implementing course concepts like using AJAX/JSON and properly designing a logical database.

Nevertheless, I am pleased about the features I did manage to implement from my initial proposal, such as an admissions form, courselist page, welcome portal page, and individual course pages, as well as the micro-interactions within those pages like a dynamic events calendar, allowing users to filter courses for display, and pop-up modal boxes for login, editing account information, and enrolling in new courses as a registered user.

Changes to Initial Proposal in Final Development:

Moreover on the compromises I had to make in developing the final version of this project, I initially wanted to simulate course enrollment similar to adding products to an online shopping cart, but eventually it made more sense for both the user and to me as the developer to make it as straight-forward as possible. Course enrollment is black-and-white -- you either fill out an admissions form if you are a new user/enrollee or click the “Enroll in new courses” button found on the welcome portal (changed to “dashboard”) page.

As for the AJAX portion of this project, when I was writing my proposal, I had limited understanding of how AJAX, JSON/XML, and jQuery worked, so I was a bit lost on where to implement it. Near the second stage of development, I decided that I was going to use the Google Calendar API and embed a school events calendar on the static front page, but as I neared the completion of the third and final stage of development, I was informed by the TA that there needs to be a transaction between a PHP file generating school events data in a JSON format on the back-end side and dynamically displaying it through jQuery on the front-end side. Subsequently, instead of using a public Google Calendar to store school events data, I was able to generate a PHP file in JSON format which obtains school event data through a “school_events” table in my project’s MySQL database.

Sofia Bautista
301292776
IAT 352 (D101)
04/04/2017

Issues & Challenges:

In addition to the issues and challenges I described in my Project 2 Technical Report, during the third and final stage of development, besides the AJAX/JSON adjustments made to comply with my TA's suggestions, I encountered little to no issues and challenges in implementing the rest of my project's features. Personalization required little effort because I simply needed to make a few extra MySQL queries to obtain enrollment information through an *inner join* of the "courselist", "teaches", and "teachers" tables. The result of the MySQL queries were then interpreted through PHP to display course links with course title, description, and teacher on the dashboard page.

However, I faced many time constraints, mostly due to a mishap involving massive data loss while updating XAMPP on my local machine during project 2. This incident was a huge setback which required me to rewrite a lot of my project 2 code and allowed less time for me to work on project 3. I believe that if the data loss did not happen, I would have been able to implement the features I was forced to compromise, such as the messaging system, grades and reporting pages.

Possible Opportunities for Improvement:

There are many ways in which this web application can be improved in the future. Firstly, I prioritized functionality over aesthetics during this project, therefore I believe choices in colour scheme, font choice, and images needs work. Another way would be storing individual course files in a database rather than in local folders along with all the other .php, .css, .js, etc. files. I have not had the chance to properly research how this can be executed, but I have an educated guess that it requires the use of non-relational "NoSQL" databases through web stacks like MEAN (MongoDB, Express.js, AngularJS, and Node.js). Thus, if other course management systems on the market like Canvas, Moodle, Blackboard, and Desire2Learn are able to store hundreds of course files, then there should be way for me to integrate such a feature into my web application as well.

On a related note, allowing students to upload completed assignments and also storing those files in a database (which was in my initial project proposal) would have been a nice feature to add to my web application, but once again, I realized that it would most likely require me to use technologies outside the scope of this course. It would be good practice in the near future to take my back-end development skills to the next level by familiarizing myself with other web stacks like MEAN to implement such features.