Raymond Tang
rt2ck
Richard Li
yl4tc
CS 4753 ECommerce Website Assignment 3

Overview

For this assignment, we sought to create the end to end functionality. We implemented the core course matching functionality under *select.php.* This was done with AJAX and will sort results. It will also only display other majors at first, and hide the resulting data. If the user is logged in, it also automatically loads results based on the user's declared major. This is because we felt this information to be most valuable to a registered user.

Decision process

We opted for an AJAX implementation call from PHP because it would not require the user to reload the page in order to view the data. The code for this is available in *majorMatch.php* It also allowed us to load the page much more quickly because not all the information is presented to the user and is only loaded as needed. We also chose to sort the more relevant information to the top in order to provide the user with a better and clearer match. In addition to this, we obscured the actual listed courses for each match in order to reduce the amount of information presented at one time to the user. To do this, we utilized javascript and html. The code for this is found in *Scripts/select2.js.* We placed all the javascript into separate files in order to have cleaner and more modular source code.

Issues

We ran into minor issues when implementing AJAX, Javascript hiding, Sorting, and the core functionality. However, we were able to overcome all of these issues. We felt that in the future, we would elect to use a framework or bootstrap in order to abstract away from minor errors that could have been otherwise more easily avoided. We also wrote this with emacs and vim which did not give us any feedback in terms of minor syntax errors. An IDE would have been helpful for this. However we were able to sufficiently overcome these errors.

Technicalities

To do all of this, we gave the admin account a form to input an add courses. We then utilized one database called Courses which stored CourseID (CS4753), CourseName (ECommerce), Majors (Computer Science). If the course had more than one major depending on it, it would list it as a comma separated list in the same field. This was then exploded using php into another array. In *majorMatch.php* we created major objects which stored the number of courses it had, the name of the course it had, and the courses' ID. We iterated through and presented this information in a table field. We also utilized the major's number of courses field in order to sort with the php usort command. We created a basic custom comparison sort to sort the majors from highest to lowest.

The AJAX implementation proved slightly difficult at first as we were not sure where the issue lied. It turned out to be several issues, which we iteratively solved and made better as we tested utilizing Chrome's Web Console.

Future

For the future, we do not plan on making this a business, and we intend to keep this a free to use site. Simply because we are all College Students and this was a tool built to enable other students to succeed and do well. This was also a tool built to learn some web technologies and how to implement them. Looking back, if we chose to pursue this topic again, we would probably utilize a web frame work such as Ruby on Rails, Python Django or CakePHP. This would significantly reduce development time as abstraction would make the site more aesthetically pleasing and enable the code to be more flourished and elegant.