Milestone 4
CSCI 3308
Team Adrestia
04/08/18

Backend databases

Home page data: (Zihan and Natnael)

The whole team has discussed about the main course optimizing page layout and we drew and agreed on some features. I started working of the left side of the page that would be utilized by the user to choose a course of study and the following classes desired. Zihan and I where able to make this portion of the page partially work. With the help of Derek we were able to connect the html layout, using Django, to the database. After signing in the user will be taken to the main page, in which he will be able to make his selections, optimize his selections and get useful information about the courses. Even Though we have a lot of work ahead of us, we are on the process of making this main page functional.

Database for course: (Derek and Long)

While we originally chose MySQL for the database for our project we have started deploying it to heroku which runs postgreSQL by default. Fortunately using a different database in django is simply a matter of changing some of the settings. We currently have a database deployed to heroku and our app is connected to it. We plan on having three main tables, one is for users, this is created automatically by django's auth module which automatically handles user account creation and data. A table for all classes with their relevant attributes. This table is generated by preprocessing the data that we have from the fcqs and then uploading it the database on heroku. This is necessary in part because of the row restrictions of a free heroku account but also so that we don't have to recalculate attributes for courses everytime someone queries a course, they are all pre-calculated. Our third table contains the state information for each user, that is what courses they selected, how they are sorted, and various settings the user has picked. We have written scripts in python to preprocess the data for the courses using pandas, and then upload it to the database on heroku.

Database for user information: (Tenzin and Heinz)

We are using django and html as front end to get user data. For example, create account site gets user information like names, userid, password, email. The login site gets userid and password to login. When a user first accesses our website, he will be presented with the login page. This page has a username field and a password field at the very top. The user is also presented with other options, such as creating a new account or doing something if they forget their password. The create account site allows a user to create an account. It asks for a username, an email, and a password. Django has built in functions to check that the email is valid, that the passwords match, and that the passwords are valid. To handle backend data storage, we are using heroku tools to store information gathered from user input. The sign up page stores the information the user gave it in a database. We have also implemented encryption so that we are not storing sensitive data in plain text. The login page checks the input

that the user provides it against the same database and retrieves the person’s information if they are found. The database of heroku works like mysql which was helpful.