Team Members:
Tyler Dionne (tdionne2021@my.fit.edu), Kendall Kelly (kelly2021@my.fit.edu), Braden Corkum (corkumb2013@my.fit.edu)

Project Advisor:
Sneha Sudhakaran, ssudhakaran@fit.edu

Project Title:
Automated FIT Transcript Evaluator

Client:
Sneha Sudhakaran

Dates of Meetings with the Client for Developing this Plan:
Once a week every two weeks

Project Plan

Goal and Motivation:
This project is a web-application that will allow users to check whether their transcript credits from different universities can be accepted based on a specific catalog from the Florida Institute of Technology. The goal of this application is to create a more efficient and simple way to perform credit evaluations for the users. It is important that this application is user friendly and represents the information in a clear, concise, and accurate manner. The motivation behind this project is to assist students and faculty alike to be able to quickly compare transcripts from other schools with a specific catalog at Florida Tech so that they may have a better understanding of their current academic standing in a more efficient way.

Features

1. Two Tier User System: In our web-application there exists two user types; user and admin. The user can access the automated transfer credit evaluator software where they can input two files as well as make a user account. The admin can access the source of the website and see the backend processing. The key focus is to assure that the user account cannot access any files or resources on our web application that might give them insight into an exploit on the database or the web app itself. The admin account will be an example of an account that needs to be secured because it will have access to components of our software and web application that the user should not have any access to.
2. Two User Inputs: There exists two user inputs to our software which include a transcript and desired catalog to compare it to. The transcript file will include their transfer credits from another university. The catalog file will include a link to the page with the desired catalog from the Florida Institute of Technology website.
3. Evaluation Result Output: There exists one output from our software which is an output file containing results of the evaluation. The evaluation output will include detailed

results from the transfer credit evaluation algorithm which takes the two input files from the user on the web application and uses these files in an algorithm which examines the way in which the schools database will process the transfer credits from the other institution.

Novel Features

1. The evaluation is automated rather than manual. Typically, the systems already in place require staff to manually type in a list of the transfer credits to compare to the FIT catalog. Our application will only require an upload of a file with the transfer credits rather than having to manually input the transfer credits.
2. We are ensuring the security and integrity of our application by creating a two tier system which will only allow admin accounts to have access to sensitive information about/within the system.

Tools algorithms

1. The platform we will use to develop the web application is Replit. This platform lets us do everything in the browser, supports html and css front end, and has backend support for several languages ex. Python, Node.js.
2. For the database for our web application we can use MySQL or SQLite to implement the management and storage of user accounts/info.

Technical Challenges

1. We will need to create an algorithm to accurately match courses from other universities to courses at FIT. This algorithm will need to be able to properly evaluate course equivalencies.
2. We will need to implement the two tier system in order to ensure security in our web application. The users should not be allowed any unauthorized access to the source code of the website and our web app should be able to resist most common web vulnerabilities.
3. We need to implement secure file handling (upload, processing, and storage) in order to prevent any possible security vulnerabilities. For example, an attacker could upload an executable file that could result in some executable or malicious script that could be used to manipulate our tool or access sensitive information.