

Jamie Nachbar (jtn26)
Senior Project Proposal
Advisor: Prof. Ronald Coifman
February 17th, 2022

For my senior project, I am proposing that I develop a new web application to replace the current system for managing the assignment, submission, and grading of questions in AMTH 160: Structure of Networks. I am hoping that this project will give me valuable experience in building this type of application, since I have no current experience with web development.

Background:

AMTH 160 is a course usually taught in the Spring. It is intended to be a course for those who do not have much mathematical maturity, and it covers topics such as graph theory, basic linear algebra, and basic probability theory. Homework assignments for the course are automated. Because of this, it is desirable to give students their own unique problem in an attempt to limit the amount of unauthorized sharing of answers.

The current system is built on top of Google Scripts, which provides an API for interacting with Google services. Currently, all information for each student is kept in a separate Google Sheet, one for each assignment. There is a script that users can run to query the spreadsheet to receive their personalized questions. They then use a Google Form to input their answers into the spreadsheet, and they run a second script to query the status of their answers. Grading is determined by formulas on the spreadsheet. Students typically have the opportunity to submit multiple times for a homework assignment and only once for a midterm or final exam.

Problem Statement:

Most of the issues with the scripts stem from the fact that Google services were never meant to be used in this way, especially Google Sheets. We are currently using Google Sheets as a database that individual users can add to and access, but this requires that each student have access to the spreadsheet. The students cannot view the spreadsheet without the link to it, but this is obviously sub-optimal security. In addition, students are required to accept a concerning set of permissions, including allowing the script to access and view the spreadsheets in their Google Account. There is unfortunately no way around this; in order to programmatically access the

scripts as each individual student, they need to give this permission. Of course, the script does nothing malicious with these permissions, but every year a few students are rightfully concerned about the wide scope of access the scripts have into their account.

Another issue is the usability of the project for students. Students need to manage multiple links for each assignment, and changing your answer to a single question in an assignment requires resubmitting the entire Google Form. Prior to my taking on this role, the previous maintainer of the scripts was attempting to refactor the project into a single Google Script that could handle providing the questions, submitting the responses, and providing the grades all in one page. However, there was a crucial problem with this approach. If the students are to edit the database themselves (as opposed to via a form), they need edit access on the spreadsheet, which means it is then visible in their Google drive account, which is unacceptable for obvious reasons.

In addition to the students, managing this current system can be onerous for the teaching assistant as well. The same manual actions need to be repeated every time for each assignment, things like updating the class roster, creating and deploying the scripts, copying the form, all things which do not necessarily need to be repeated.

The current system is also not very extensible. We are capable of rendering text that is personalized to each user, but if we wanted to include a more complicated or interactive assignment it would not be possible with the current system. In addition, managing complicated question generation and grading logic through a spreadsheet is more challenging than programmatically (at least for my generation of programmers!).

Proposed Solution:

For this project, I am proposing to rebuild this system as a unified web-app, running Flask on the backend and React on the frontend. I chose Flask because I am already quite familiar with Python and Flask has a rich ecosystem of libraries that I can use to provide additional functionality. I am going to build this on top of MySQL as the database system, and I will continue to use Google's authenticator to provide access and assist with account management. It is essential that the program is reliable, so the system will include robust failure handling and the automatic creation of backups for the database and any other auxiliary files.

The entire app will be containerized and deployable with Docker. A remaining question is what machines the application would be running on. Since this app will be running as a container, we can be quite flexible with our choice. In general, the app is quite

lightweight in terms of computation power required and requests made to it, but it is necessary that it always be available for students to interact with the system. Two potential ideas are hosting it in AWS on a lightweight EC2 instance, or just hosting it on the Zoo. However, the project is structured to be agnostic with respect to the hosting provider.

I am considering two key user-groups. The first group is of course the students in the course, who will be interacting with the system regularly in order to complete assignments, post and answer questions, and check their grades. My second group are the Teaching Assistants who will be using this system in the future. My goal is that while there should be documentation and the capability to modify or fix the system should something break in the future, TAs will not need to venture into the source code on a day-to-day basis and could use the web-app in the same way that the students did. There would be functionality like creating a new class, importing questions and students from spreadsheets, sending out announcements, etc. All in all, the system would be quite similar to Canvas, except with increased flexibility and the ability to procedurally customize content for each individual student. I would also explore integrating some features of the system with Canvas, such as grades. A top priority of mine is that the system is easy to use from both of these perspectives.

Deliverables

1. The primary deliverable is of course the application itself, which will hopefully be used in future offerings of AMTH 160.
2. The secondary deliverable would be documentation and tutorials surrounding how the website is built (and could be modified) and how it should be used from the perspective of future TAs managing the system.