University of Idaho Scheduler

Spring 2020 (March – May)

The goal of this project was to create a better scheduling system for students at the University of Idaho. It is a system that recommends classes based on the user’s career goal. For example, if you want to be a game developer, the software will recommend 3D modeling and other related topics, while never recommending other interests such as network security.

The program allows a user to add a goal career field. From there the program recommends classes that are most fitting for that field. I used **Apache**, **MySQL** and **XAMP** to link everything and the bulk of the code was written in **PHP**.

This project was a fun change of pace from c programming. It is not a robust system, more of a proof of concept but it has a couple practical features. When a person is added to the database, they need their student ID and pick a goal industry. The database uses each student’s ID as a primary key and goal industries can be changed at any point. The image below is an example of the interface in use.

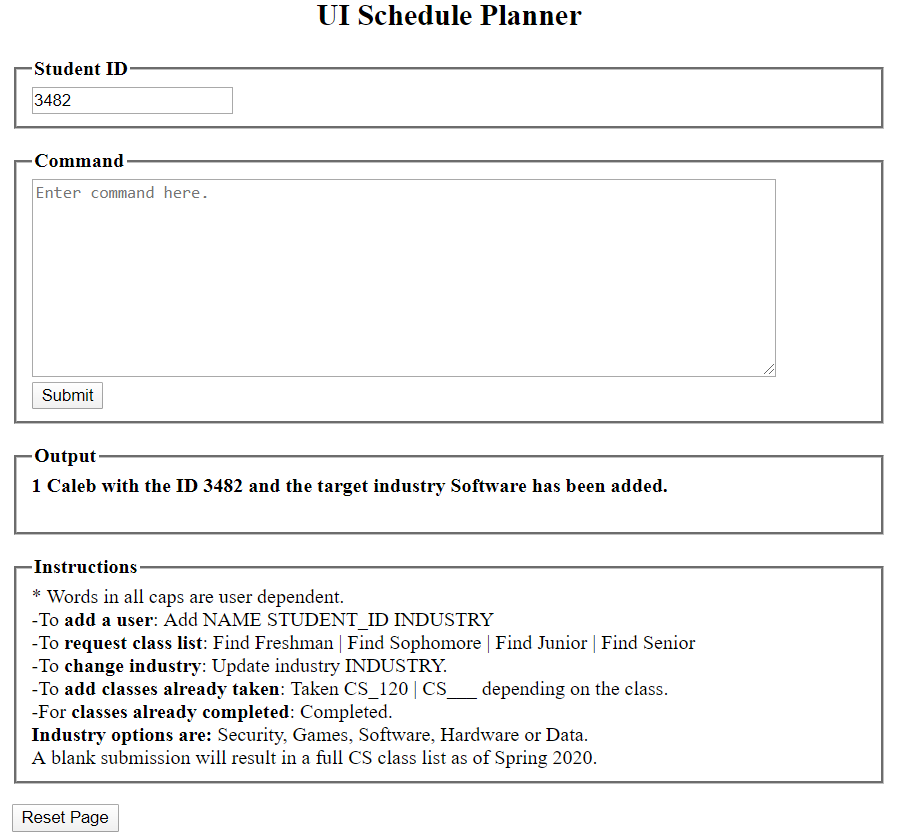
The code uses a combination of PHP, JavaScript, and SQL. I isolated part of the most interesting code snippet below. It is only a section of the SQL command I use.

This section of the command eliminates classes that the user has already taken.

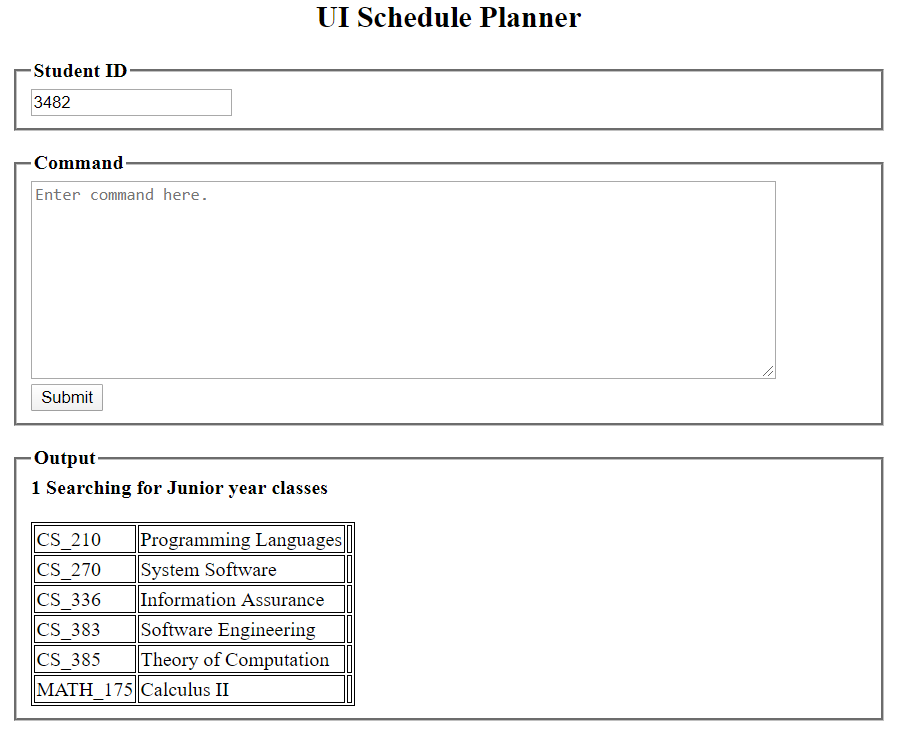
I also used Apache, XAMP, and MySQL.

This project was not code heavy, but the new language and tools

This project provided valuable experience working with new tools and a live database. Although it is not code heavy, I learned lots from my



The part I love most about this project is class recommendations. I find it incredibly satisfying to see class lists change depending on your interest and the classes you have taken. This program does not recommend CS II until you have taken CS I as well as other classes that the user has not completed prerequisites for. Below is an example of a recommend class list for juniors, not shown is the list of classes the user has already taken such as Calculus I and introductory CS classes.



This program is not secure beyond a student ID and a user could add nonsense classes or make mistakes in their commands. It is nowhere near a bulletproof system but it largely accomplishes what I set out to do.

This project was not a ton of coding, but it was a lot of new, new programs and new languages, making it a discouraging process. I had to manually add each class and twice I deleted the entire database forcing me to reenter it each time. Fortunately, by the third time I had learned some tricks to speed up this process.

This was also my first time using XAMP, Apache and MySQL. Getting everything to work together was a major hurdle itself and I had not written a line of code yet.

Looking back at this project I am incredibly satisfied with what I accomplished. The system is a fun visual of my accomplishments but most important to me is tackling so many new variables. At the beginning of this project, I really was lost, and it took some discouraging attempts to finally make progress.

You can check out the code [here](https://github.com/Caleb-Seely/UI_Schedule/tree/master). I recommend lines 199 -236 of *UI\_Schedule.php,* it is the SQL command I use.