

# Homework 2 — Individual Project Proposal

Collin Sinclair 107072664

## 1

My project is a tool to access CU Boulder's FCQ data in a more useful way than that provided by the default Tableau tool. It is accessible via a website (probably hosted on Heroku) where there will be a welcoming and easy-to-use UI.

Users can search for anything that identifies in the CU Boulder FQC data: a course number, instructor name, department, or college. They will be presented with a page full of all the statistical analyses that are available for the given entry, which may include seasonality (when the course is typically offered), instructor group (tenure-track, instructor, etc.), typical enrollment/class size, difficulty, projected hours of work per week, course/instructor overall, and metadata like response rate.

Additionally, users can create an account/sign in to store "schedules", theoretical lists of classes with grouped statistics, like total hours of coursework a week or largest/smallest expected enrollments.

## 2

I plan to use the typical HTML/CSS/JavaScript for the front end, with Bootstrap 5 for styling and Plotly for rendering plots. I'll use Python with Pandas as a server-side language for all of the statistical analysis and PostgreSQL as the database for all of the FCQ data. I will test the statistics and HTTP URL responses with PyTest by hand calculating statistics for a set of inputs and hard-coding them into the testing framework.

Technology	Familiarity
------------	-------------

HTML	Quite familiar (4)
------	--------------------

CSS	Familiar (3)
-----	--------------

JavaScript	Somewhat familiar (2)
------------	-----------------------

Bootstrap 5	Familiar (3)
-------------	--------------

Plotly	Not Familiar (1)
--------	------------------

Python	Very Familiar (5)
--------	-------------------

Pandas Very Familiar (5)

PostgreSQL Not Familiar (1)

PyTest Not Familiar (1)

### 3

The project won't work if it doesn't have

1. A front-end (HTML/CSS/JavaScript) — without this there is no interface or way for users to access the tool! Plus, the point of this is that it is more convenient than the current tool which requires being online since browser technology is the most consistent across devices
2. A way to display plots (if Plotly doesn't work for whatever reason, Matplotlib is a viable alternative — I'll have to generate the plots on the backend and display them on the front end; this will restrict interactivity but is better than not having plots, which are the whole fun of the project!)
3. A way to do statistical analysis (Python) with some data frame-like framework (Numpy arrays would suffice, but I don't anticipate having trouble with Pandas). Without these, I will not be able to provide the insights that the tool is designed to provide

### 4

I will require CU Boulder FQC data from previous years, which I actually already have and is publicly available as a spreadsheet [here](#). It is presented well in the spreadsheet and will read into Pandas niceley.

### 5

My front end (HTML/CSS/JS) will make HTTP requests to my backend (Python). I will use PostgreSQL as my database. The front end will be a website that gives the user graphical access to the features of the app. It will allow the user to instantly get info on a college/department/course/instructor, as well as make comparisons between these categories (each of which will be objects, see below). Further, the user will be able to sign in and store classes in their schedule, or pit different sample schedules against one another to see how their stats work out.

At its core, this app is about accessing and understanding data. The Python backend will be almost entirely working with objects - each of the large categories of data will be a class (e.g., colleges, departments, courses, instructors, class type, etc.) with its relevant fields and methods. There will also be a schedule class, where users can create theoretical schedules and get statistics on them (e.g., expected time to be spent on class every week). When the user enters a course number, for example, the front end will make an HTTP request to the backend

for the stats on that course. The Python backend will listen for this request and then fetch the relevant values from the database, perform statistical analysis on the data, and return a JSON object with values to the front end. Then the front end will parse the data object, draw the relevant plots and display values for the user.

Regarding OOP and polymorphism: the course class will act as a parent for the section object, and the college class a parent for the department class. All of these will be descendents of a data class, which will identify the objects as being representations of raw data straight from the FCQ. Additionally, there will be a schedule class that will represent the aggregate information for a theoretical schedule, and will primarily be used to make comparisons to other schedule objects.

My database will be relational, with tables for raw FQC data, user information, and stored schedules. The schedule table will share a one-to-many relationship with the users, and with the FCQ raw data.

## 6

The website landing page will have a large space to enter a course number / department / instructor for the user to use the service immediately, without needing to sign in. There will be a navbar at the top of the page that will allow the user to go home or see their saved schedules, as well as sign in or create an account. The page for a specific single query will display several plots to visualize the FQC data, as well as a summary in prose, with the option to compare the current course/department, etc. with another. See images at end of document — I couldn't get good formatting inserting them here.

## 7

### Homework 2 (February 27; 5 days)

Turn In A scaffolding of HTML pages laying out the GUI of the site. The user will be able to click around the site and visit different pages, but all content will be hard-coded and predefined.

Knowledge Needed HTML/CSS/Basic JavaScript, Bootstrap 5

Need to Learn Familiar with everything this week

### Homework 3 (March 11; 12 days)

Turn In	Database with raw FCQ data — users can query and see results. User authentication — users can create an account / sign in.
Knowledge Needed	Database (PostgreSQL), password hashing, HTTP requests (for auth), Heroku (for hosting the backend) + all of the above
Need to Learn	Specifics of PostgreSQL, implementing a secure password hash, HTTP requests, Heroku

## Homework 4 (April 8; 28 days)

Turn In	Python object structure and definition, statistical analysis. Users can use Python to get numerical statistical insights back based on queries to the FQC database. Users can sign in and save classes to a schedule, and make arbitrarily many schedule objects.
Knowledge Needed	Python/Pandas, statistical analysis, HTTP requests/responses for populating site fields from backend + all of the above
Need to Learn	Will already be familiar with all of the above (familiar with Python/Pandas, stats as of 2/22/22; HTTP requests as of Homework 3)

## Homework 5, Checkpoint 1 (April 19; 11 days)

Turn In	Plots on website based on stats from backend. Users will be able to enter a course/department/instructor etc and see visual representations of the data from the FQC results.
Knowledge Needed	Plotly + all of the above
Need to Learn	Plotly

## Homework 6, Final Due Date (April 28; 9 days)

Turn In Full working website with all kinks worked out. Edge cases tested, should be very difficult to break/encounter unexpected behavior. Finalize aesthetics/styling.

Knowledge Needed All of the above.

Need to Learn Hopefully, nothing by now!

## 8

To stay engaged in the course, I plan to treat the course and programming exercises as a separate endeavor from the project. This will ensure that I'm still getting experience with the subjects in class and putting beyond-sufficient effort into the PEs and lecture activities. This looks like answering questions in lecture and engaging in group work when it is time, as well as complete the PEs in their entirety and submit them on time.

# Images for 6

CU Boulder F&C Tool	Home	My Schedules	Log In / Sign Up
<div style="border: 1px solid black; padding: 10px; margin-bottom: 20px;"> Enter a Department, Course Num, Inst. <div style="border: 1px solid black; display: inline-block; padding: 0 10px;">GO</div> </div> <div style="margin-bottom: 20px;"> <a href="#">How to use</a> </div> <div> <a href="#">About</a> </div>			
Home   Help   Contact   © Collin 2022			



