

## **SI 206 Data-Oriented Programming**

### **Project Name: APIs, SQL, and Visualizations**

#### **Project Objective:**

Demonstrate the ability to:

- Create a fully-working program without any scaffolding (starter code)
- Create and modify tables in an SQLite Database
- Utilize APIs (including researching methods)
- Utilize visualization software (including researching options)
- Document your code
- Work in teams of 2 to 3 people (**no exceptions!**)
  - Working in groups is a skill that is highly valued

In this project you will gather data from one or more APIs and optionally a website (with Beautiful Soup) to answer questions such as what is the effect of weather on crime or is there a correlation between the average yelp review and the average income in an area.

#### **Deliverables and Submission Process:**

1. **You must submit a project plan by Tuesday April 4th!** See the details below. The submitted plan will be graded, but you can change what you plan to do without letting us know.
2. Students who present their project (must have all data gathered in a database, select the stored data, do calculations, and create visualizations) on **Tuesday April 18th** (during the last lecture) will receive 15 points of extra credit. They will still have till April 21st to complete the report and submit the project. We have a limited number of slots for teams to present during the last lecture.
3. **You must submit a report on your project and a zipped copy of all of your code on canvas by April 21<sup>st</sup> at 11:59pm.** Absolutely no late assignments will be accepted.
4. **Your group must attend a grading session on Zoom on April 21<sup>st</sup> or 24th if you don't present on April 18th. Your group will be able to pick their first and second choice for when they will present/answer questions. However, you may not get your first choice.**

#### **Background:**

In this assignment you will be using the skills learned from the course to gather data from one or more APIs and optionally a website (using Beautiful Soup), store that data in a database in several tables, calculate something from the data that you select from the database, and create visualizations. **If you have 2 people in your group you will need to either work with two APIs or one API and one website with BeautifulSoup and create at least two visualizations.** If you have 3 people in your group you will need to either work with three APIs or two APIs and one website with BeautifulSoup and create at least three visualizations.

**You must write at least one function in one file to gather data from the APIs/website (using Beautiful Soup) and store it in a database and at least one more function in another file to select data from the database and perform calculations and visualize the results. We recommend having functions for each API/website in separate files so that you don't have multiple people editing the same file.**

**For more specific details about the grading and how you might lose points see the separate grading rubric.**

### **PART 1 – Submit your plan (10 points)**

Submit your plan for your final project on Canvas by 11:59 pm on April 4th. You will earn 2 points each for items d-h below.

- a. What is your group's name?
- b. Who are the people in the group (first name, last name, umich email)?
- c. What APIs/websites will you be gathering data from? The base URLs for the APIs/websites must be different for them to count as different APIs.
- d. What data will you collect from each API/website and store in a database? Be specific.
- e. What data will you be calculating from the data in the database? Be specific.
- f. What visualization package will you be using (Matplotlib, Plotly, Seaborn, etc)?
- g. What graphs/charts will you be creating?
- h. Who is responsible for what? Please note that all team members should do an equal amount of programming and total work.

### **PART 2 – Gather the data and save it to a single database (100 points)**

- For a two-person group you must access either two APIs or one API and one website with BSoup (e.g. Facebook, GitHub, Gmail, Yelp, etc). For a three-person group you must access three APIs or two APIs and one website. This is worth 10 points.
- Access and store at least 100 items in your database from each API/website (10 points) in at least one table per API/website.
- For at least one API you must have two tables that share an integer key (20 points). **You must not have duplicate string data in your database! Do not just split data from one table into two! Also, there should be only one final database!**
- You must limit how much data you store from an API into the database each time you execute the file that stores data to the database to **25 or fewer items** (60 points). The data must be stored in a SQLite database. This means that you must run the file that stores the data multiple times to gather at least 100 items total without duplicating any data or changing the source code.

### **PART 3 – Process the data (50 points) – after you have gathered all your data.**

- You must select some data from all of the tables in your database and calculate something from that data (20 points). You could calculate the count of how many items occur on a particular day of the week or the average of the number of items per day.

- You must do at least one database join to select your data for your calculations or visualizations (20 points).
- Write out the calculated data to a file as text (10 points)

#### **PART 4 – Visualize the data (50 points)**

- If you have 2 people in your group you must create at least 2 visualizations of the calculated data. If you have 3 people you must create at least 3 visualizations. You are free to choose any visualization tool/software that you can create with Python code.
- You will not earn the full 50 points if your visualizations don't go beyond the examples you were given in lecture. If you use an example from lecture, you should change something from the example you were given in lecture, such as change the colors of the bars in a bar chart for example.

#### **PART 5 – Report (100 points)**

In addition to your API activity results, you will be creating a report for your overall project.

The report must include:

1. The goals for your project including what APIs/websites you planned to work with and what data you planned to gather (10 points)
2. The goals that were achieved including what APIs/websites you actually worked with and what data you did gather (10 points)
3. The problems that you faced (10 points)
4. The calculations from the data in the database (i.e. a screen shot) (10 points)
5. The visualization that you created (i.e. screen shot or image file) (10 points)
6. Instructions for running your code (10 points)
7. Documentation for each function that you wrote. This includes describing the input and output for each function (20 points)
8. You must also clearly document all resources you used. The documentation should be of the following form (20 points)

Date	Issue Description	Location of Resource	Result (did it solve the issue?)
------	-------------------	----------------------	-------------------------------------

#### **BONUS A - Add additional API sources (Max 30 points)**

- Earn up to 30 points for an additional API. You have to gather 100 items from the API and store it in the database. You must calculate something from the data in the database. You must write out the calculation in a file.

#### **BONUS B - Add additional visualizations (Max 30 points)**

- Earn up to 15 points for each additional visualization.

## Useful Links

List of free APIs <https://github.com/public-apis/public-apis>

Github API <https://developer.github.com/v3/>

Gmail API <https://developers.google.com/gmail/api/>

You have to use python-specific packages. For example, you might have to google “Gmail API for Python”.

### **Further Examples of Visualizations**

In Gmail, what percentage of emails are sent from github on Monday, on Tuesday, etc.

In Facebook, a scatter plot with length of post vs. number of likes.

In Spotify, for your five favorite bands, compare how many songs of theirs are in your playlists.

## Tips

**Start early** - This project involves learning and using a new API. Planning ahead is important, and make sure to give yourself enough time to ask questions if stuck.

**Learn online** - There are many tutorials and helpful information online. Since this is the first time you are encountering a given API, you will probably make use of them (and we encourage you to make use of them!). Remember, though, that you must document all the resources you use.

**Debugging and looking for help** - Unlike past homework and projects, here you get to choose your own APIs. This means that likely the APIs you choose will not have been seen by the instructors of the course. They will try to help in any way they can, but more often than not, you will have to debug your own code. Once again, online resources and tutorials are useful!

**Have fun!** - This project is broad on purpose. Choose sites that you are genuinely interested in and extract the information you want to see! Working on a project that is interesting is 100x better than working on a dull, boring project.