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
- Create and modify tables in a SQLite Database
- Utilize APIs (including researching possible methods)
- Utilize visualization software (including researching options)
- Document your code
- Work in teams of 2 to 3 people

Deliverables and Submission Process:

1. **You must submit a project plan by November 20th**

2. **You must have all of your data collection done by December 11th.** Students who demo their fully completed project by December 11th (in lecture) will receive 15 points of extra credit. The presentation should be 4-5 minutes long. We may not have time for everyone who volunteers to present.

3. **You must submit a report on your project and a zipped copy of all of your code by Dec 19th by 11:59pm.** Absolutely no late assignments will be accepted.

4. **You must demo your project** on either December 11th during the last lecture, December 18th from 2pm to 4pm or by appointment, or Dec 20th from 1:30pm to 3:30pm (during our final exam time). Pick your first and second choice for presenting at **https://tinyurl.com/206-Final**

Background:
In this assignment you will be using the skills learned from the course to gather data from APIs, store that data in a database in several tables, calculate new data from the data in the database, create visualizations from what you calculated, and print out a report with the calculated data and the visualizations. If you have 2 people in your group you will need to work with at least 2 APIs and create at least 2 visualizations. If you have 3 people in your group you will need to work with at least 3 APIs and create at least 3 visualizations.

PART 1 – Submit your plan (10 points)
Submit your plan for your final project on Canvas by 11:59 pm on November 20th. You will earn 2 point each for c-g below.

a. What is your group's name?
b. Who are you working with?
c. What APIs will you be gathering data from?
d. What data will you collect from each API and store in a database? Be specific.
e. What data will you be calculating from the data in the database?

f. What visualization package will you be using (Matplotlib, Plotly, Seaborn, etc)?
g. What graphs/charts will you be creating?

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

- For a two-person group access 2 APIs of your choice (e.g. Facebook, GitHub, Gmail, Yelp, etc). For a three-person group you must access 3 APIs. This is worth 10 points.
- Access and store at least 100 items in your database from each API (10 points). The database must have at least two tables in it per API (20 points) where at least two tables share a key.
- You must limit how much data you store from the API to 20 or less items each time your code runs (60 points). If the API doesn't allow you to limit the number of responses, you can limit how many items you store in the database each time you run your code. This means you will need to gather the data over time and have a way to figure out what new data you need to gather each time you run your code. Some APIs allow you to set the amount of data you get each time, which you can use. The data must be stored in a SQLite database.

PART 3 – Process the data (50 points)

- 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. It could be an average of the number of items per day.
- You must do at least one database join (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. You could 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 (10 points)
2. The goals that were achieved (10 points)
3. The problems that you faced (10 points)

4. Your file that contains the calculations from the data in the database (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 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?**

You will be graded on clarity, completeness, and presentation (no typos, neatly formatted, etc.)

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

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.