SI 206 Data-Oriented Programming

Project Name: Data-Oriented programming with APIs and Visualization

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)
- Follow proper coding conventions (documenting the functions)

***You can optionally suggest an alternative project. Your alternative project needs to be approved by **Dec 4th** and your proposal should address how you will be hitting on each of the objectives above. (If you will not be meeting an objective make sure to say why and what you will be doing instead).

Deliverables and Submission Process:

1. **You must submit a plan by December 2nd -** APIs you plan on using, visualization tools/package you want to implement, etc. Make sure to include what data you will be collecting. **Proposals for alternate assignments are also due Dec 2nd.**

2. **You must have all of your data collection done by December 10th.** Students who demo their fully completed project by 12/10 (in lecture) will receive 20 points of extra credit. Sign up at https://goo.gl/forms/cxEORmWY5dnWMtfJ3 We may not have time for everyone who volunteers to present. We will take people in the order that they sign up.

3. **You must submit a report on your project and a zipped copy of all of your code and other files on GitHub by December 15th at 11:59 pm.** *Absolutely no late assignments will be accepted.*

Background:

In this assignment, you will be using the skills learned from the course to create your own Social Media/software package tracker. The project is worth 310 points, but also has extra credit worth up to 100 points so you could earn 410 points total. If you also demo your project in lecture on Dec 10th you can earn an additional 20 points of extra credit, so the total could be 430 points.

PART 0 - Submit Plan (10 pts)

Submit your plan by Dec 2nd 11:59 pm on Canvas. This should be a paragraph saying what APIs you plan on using, what visualization tools/packages you want to use, etc. Make sure to include what data you will be collecting and what you will be calculating.

PART 1 - Get Data (100 pts)

- Access the API for a social media site or website of your choice (e.g., Facebook, GitHub, Instagram, Gmail, YouTube or any other site you have an active account with). ***Twitter is not an option.***
- Cache the data in a SQLite database over time (must have a way to check if the data is already in the database, the ability to be restarted, and must limit how much data it collects at a time).
- Access at least 100 interactions (posts, emails, commits, likes, etc) using the API.

PART 2 - Process Data (100 points)

- Calculate something from the data such as the number of items (posts, likes, etc) per day (Sunday, Monday, etc), the day with the most items, the day with the least items, and the average number of items per day.
- Create a "report" - (screen display, file output, or other easy-to-read formats) that contains the calculated data. The report can be terminal output or a .json/.csv/.txt file that is easy to read.
- If days of the week is not a natural data point, make sure to specify your other metric in the Dec 2nd plan. Another example metric could be hours of the day.

PART 3 - Plot Data (50 pts)

- You must have at least one visualization of the data. You are free to choose any visualization tool/software of your choice. You should mention which tool you will be using for visualization in your Dec 2nd plan.
- Suggestions of visualizations include comparing social media accesses on each day of the week, a google map with the locations of your Facebook friends, a Word Cloud, etc.
- You will not get the full 50 points if your application doesn't show a deeper understanding of the code - change the default colors, including legends and chart names, etc. from any sample code you use.

Here is an example looking at how many posts a user made in each day on Facebook vs. Twitter: https://plot.ly/~colleenV/6/twitter-vs-facebook/ Here Twitter is used for comparison, not as one of the APIs.

PART 4 - Report (50 pts)

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

1. Your goals
2. Which goals you achieved
3. What problems you faced
4. Your social media "report"
5. Instructions for running your code.
6. Documentation for each function you wrote (Code must be fully formatted. Describe the input and output of all functions)

7. You must also clearly document all the resources that you used. The documentation should be of the form:

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

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

BONUS A - Use additional API sources (Max 60 points)

- Earn 30 points for each additional API for a maximum of up to 60 additional points.

BONUS B - Use additional data points (Max 10 points)

- You can use additional data points to represent your data.
- For example, you can break each day into time of day (use 12:00am - 5:59am, 6:00am - 11:59pm, 12pm - 5:59 pm, and 6:00pm - 11:59pm). This gives you 28 data points instead of 7.
- If time is not a natural data point, make sure to specify your other metric in the Dec 4th plan.

BONUS C - Create additional visualizations (Max 30 points)

- Earn 15 points for each additional visualization up to 30 additional points

Integrity Policy:
All materials submitted by students must be their own work - you may not submit material from previous semesters or examples taken from class or the Internet. ***Students may NOT discuss the homework with others***. ***Any instances of cheating will receive a 0 on the assignment and one letter grade deduction in the final course grade.*** If you are unsure about the integrity of your submission, you have 48 hours after submission to withdraw your submission.

Useful Links
Facebook Graph API (https://developers.facebook.com/docs/graph-api/)
Github API (https://developer.github.com/v3/)
Gmail API (https://developers.google.com/gmail/api/)
Remember, for some, 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 homeworks and projects, here you get to choose your own API. This means that likely the API 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! - Maybe overly cliched, but this project is broad on purpose. Choose a social media site 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.