**We Are Dorks Project Report**

**By: Kashaf Usman , Wilson Tanner, Lissette Ramos**

**Link to Github repository:** [**https://github.com/LissetteMariePortfolio/WE\_ARE\_DORKS\_SI\_206\_Final\_Project**](https://github.com/LissetteMariePortfolio/WE_ARE_DORKS_SI_206_Final_Project)

**Goals:**

We wanted to understand music, movie and social media entertainment trends by using data to determine

* + The breakdown of movie genres in the top 100 movies of IMDb
  + The breakdown of music genres in the top 100 songs / music albums via iTunes
  + The breakdown of video game genres in the top 100 video games via Steam

We reached the first two of our original goals, but we needed to change the third goal, as we needed to change the API. The third goal achieved was

* + The amount and proportion of interaction posts in 2 movie and 2 music subreddits receive (upvotes and comments)

**The Struggles We Faced**

Carrying on from the last point, when Lissette looked at the Steam’s API documentation, she found the how-to of using the API was rather unclear. It wasn’t even clear if there was a Python wrapper for this API. So she proceeded to spend that day looking at different APIs, looking for ones with clear documentation that would allow her to do the needed data analysis. Finally she landed on Reddit’s PRAW API, and with some helpful video tutorials, she switched to Reddit ~~(and fell deeper into her belief that reddit is the only tolerable form of social media)~~

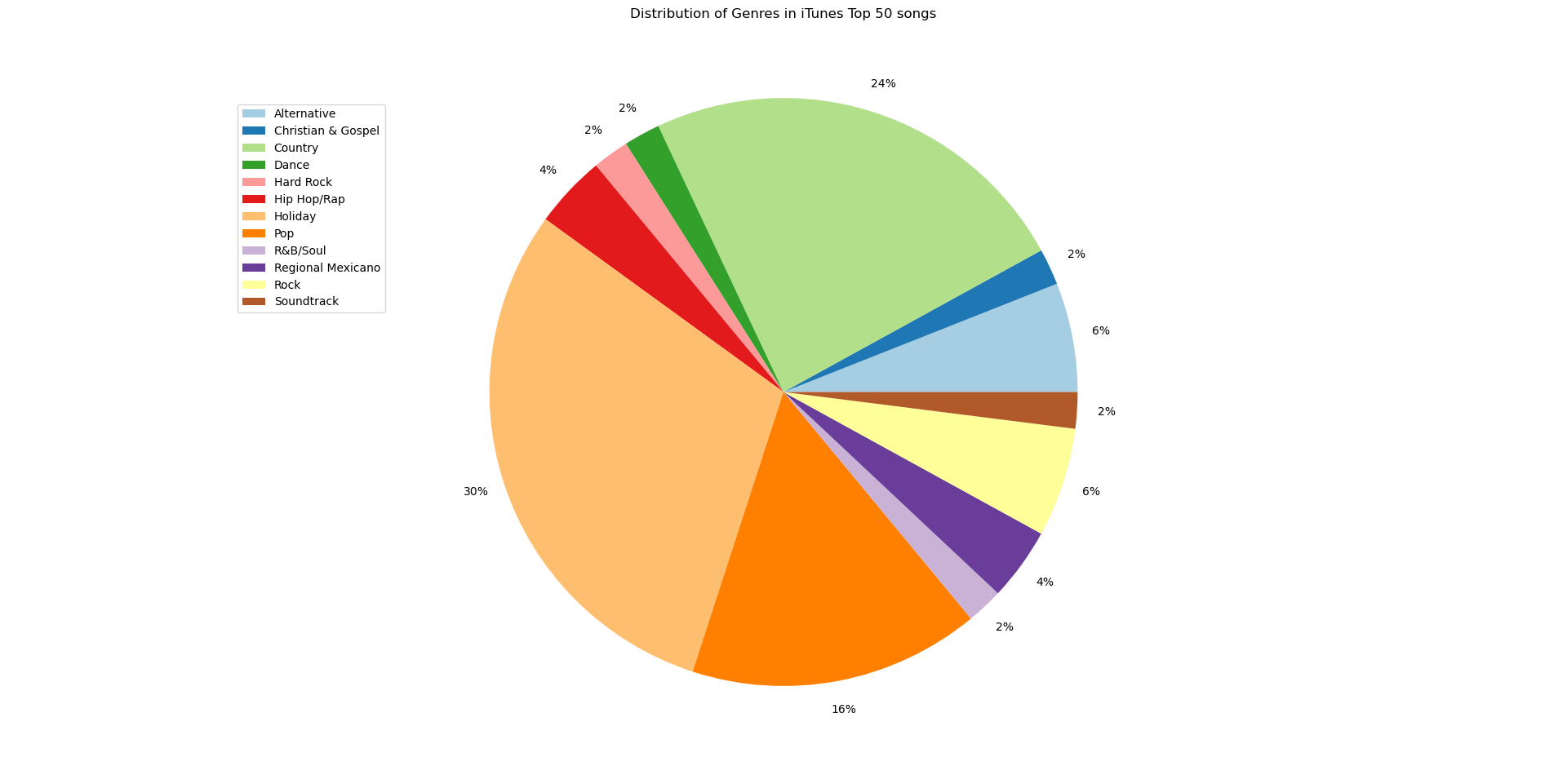
Wilson dealt with a last minute issue where his chosen iTunes API gave him 504 Gateway errors. What we didn’t know was that Amazon was down that particular day, but not knowing when or if the API would come back online at that time, Will went ahead and wrote his code to use a different API the night before the presentation.

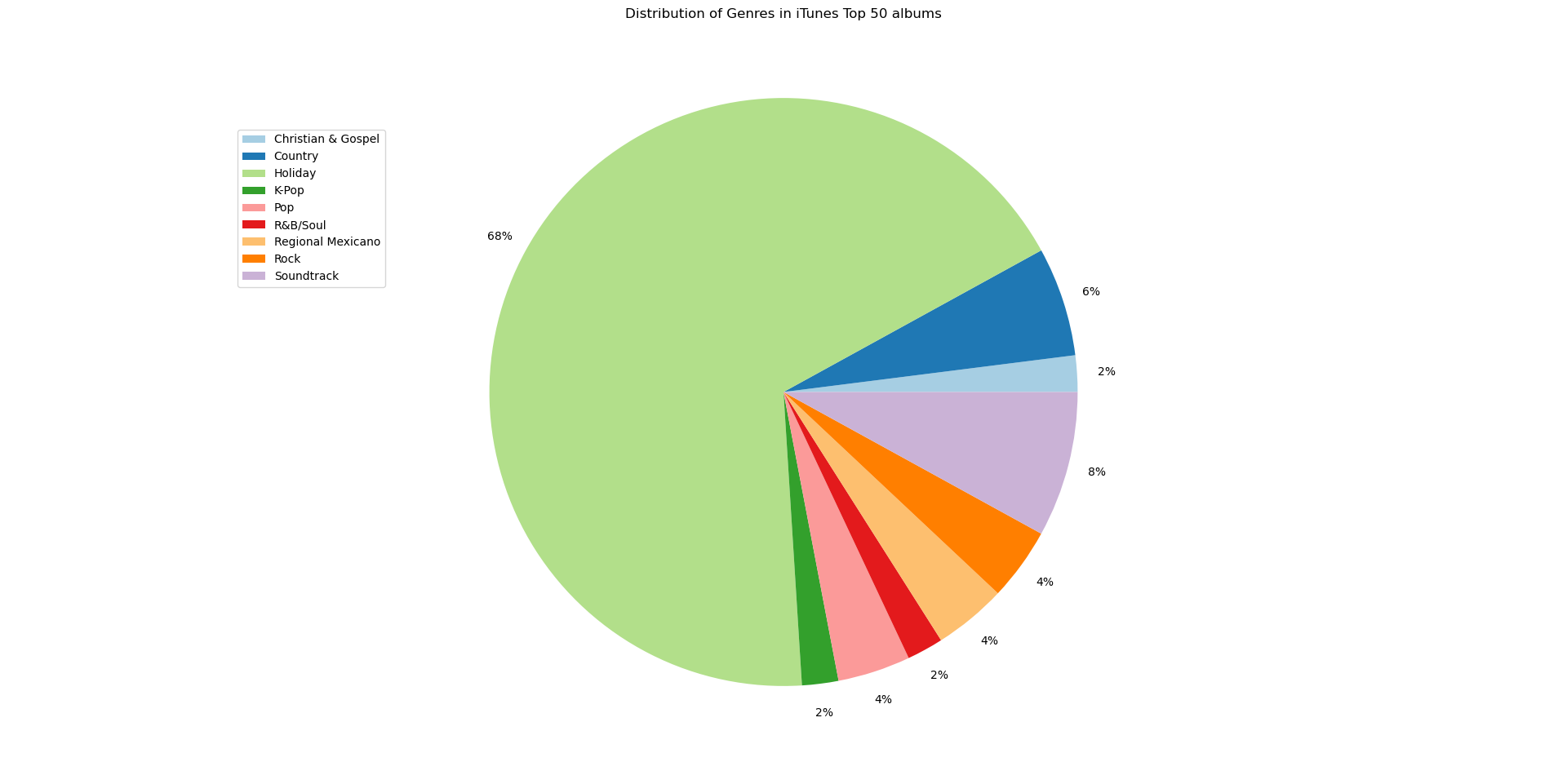
Kashaf dealt with a personal emergency where she needed to leave the country for India which has much worse internet. She was still able to successfully complete her code for her chosen API, running meetings was just more complicated, and due to technical issues, she was unable to present with us in class. She also was unable to help us put the code together. We lost contact with her after the presentation on December 8 so she could not make the proper edits to her table to better integrate it into our database. She also has not done her section of the report (function / resource documentation).

**Files**

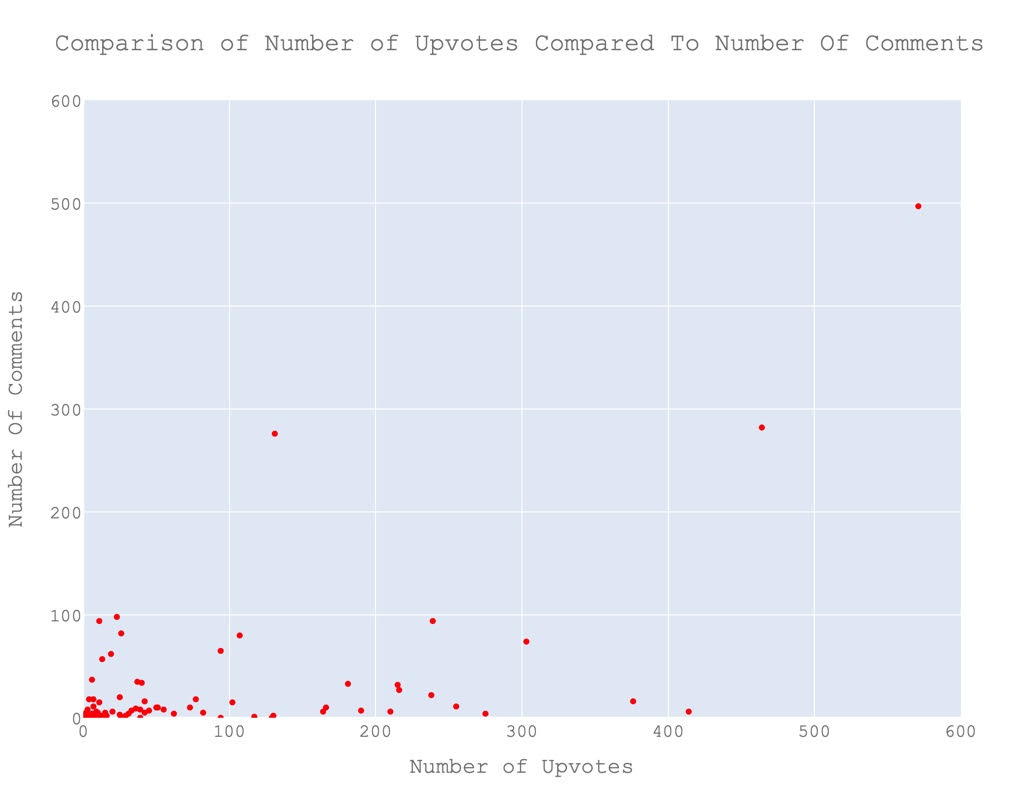
**Graph images**

**Music Genre Visuals:**

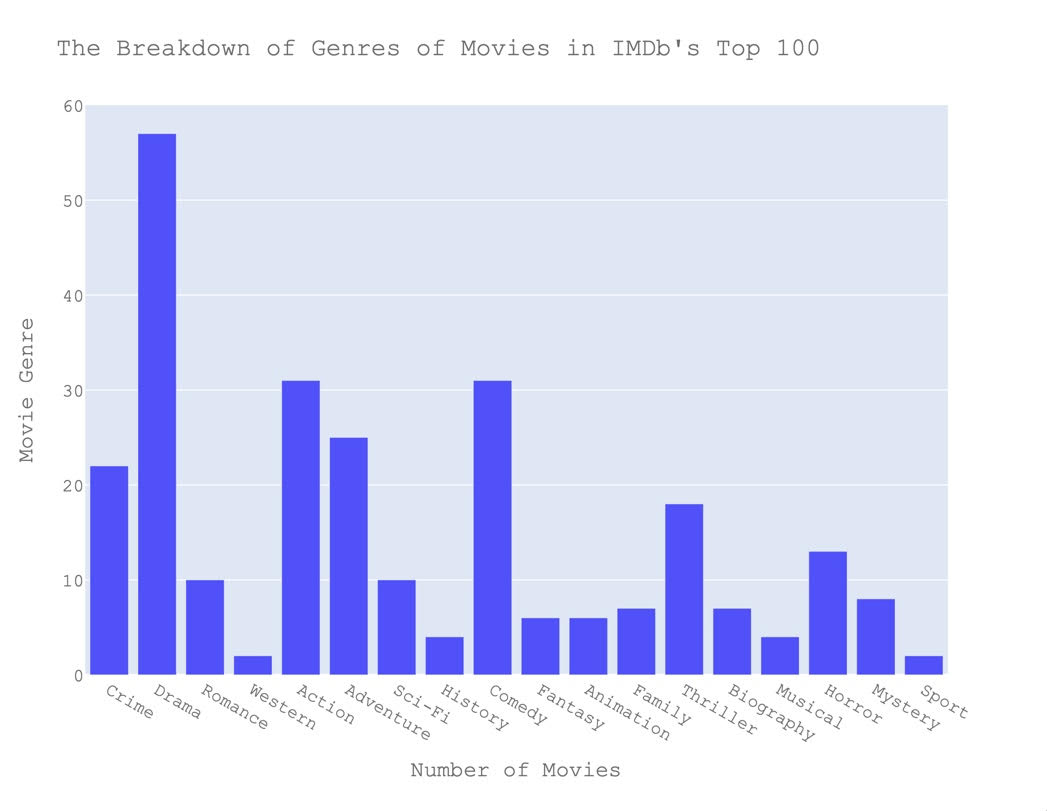
****

****

**Reddit Interaction Correlation Graph:**

****

**Genres of Top 100 Movies:**

****

**File of Calculations**

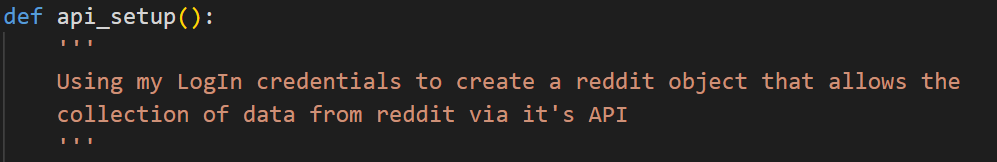
See text file

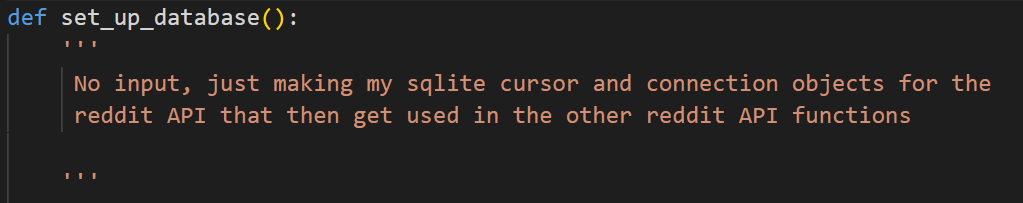
**Instructions For Running The Code:**

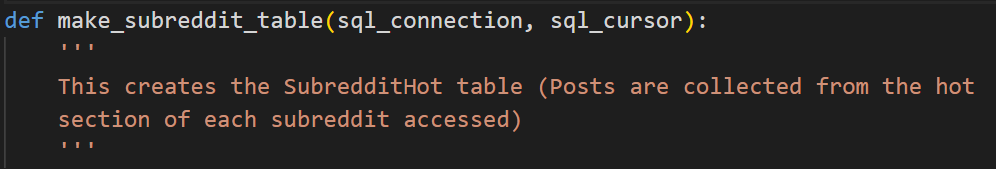
Put entertainment.py in your favorite IDE and just hit play. That’s it. You might need to pip install praw or other resources. Warning, praw slightly increases the time it takes for the code to run immediately, it just takes some time for praw to finish. Sorry for that. When the code runs the 4th time, you’ll first see Kashaf’s bar chart, then after a short pause, you will see Lissette’s Reddit chart, then after a shorter pause you will see Will’s pie chart. If you wish to run the code again after that, you must close out of the Python pie chart window as it stops further visuals from being shown.

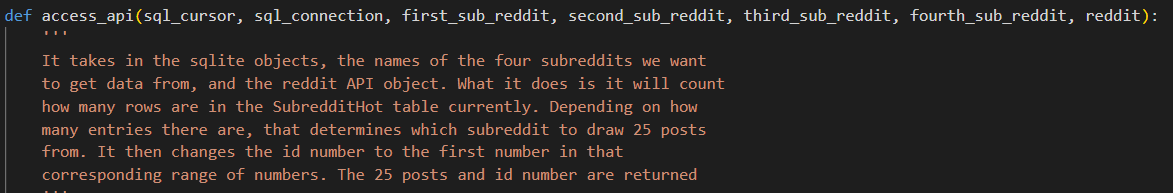
**Code Documentation:**

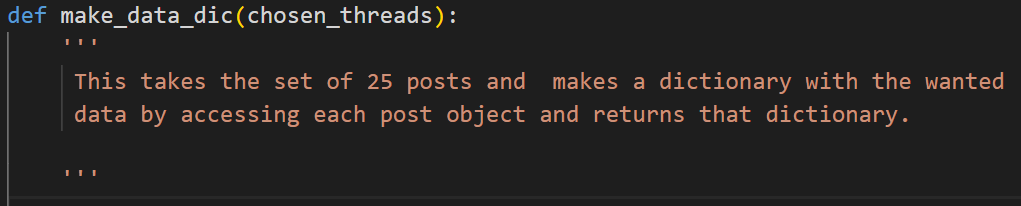
Lissette’s Code:

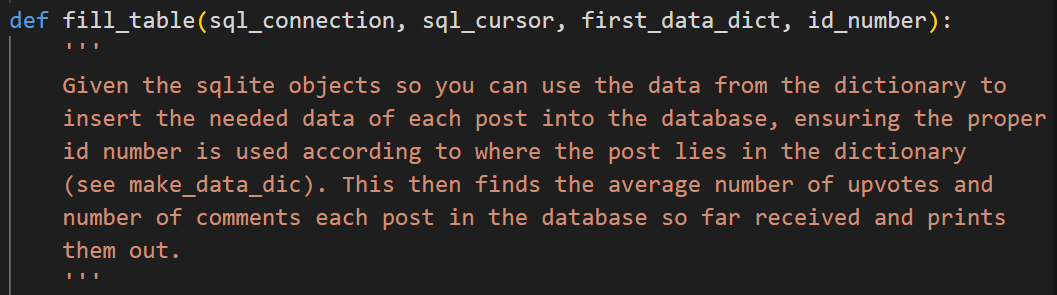


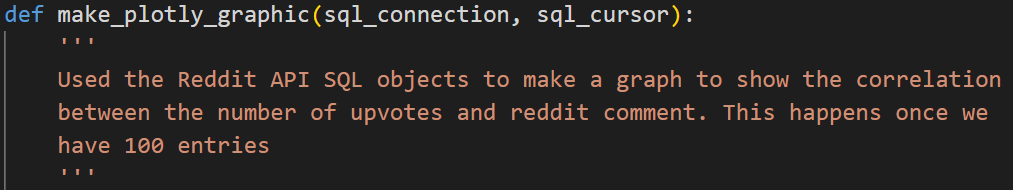




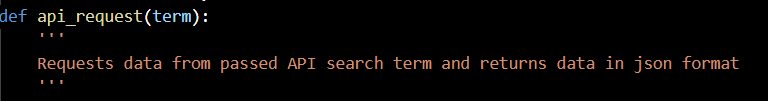


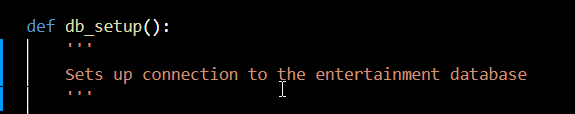


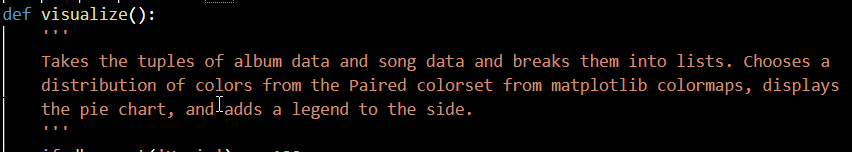
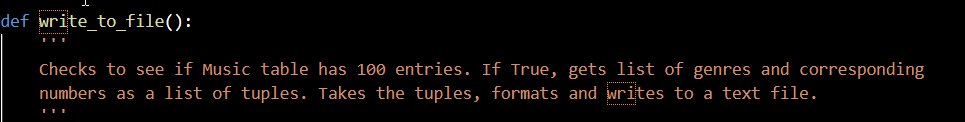
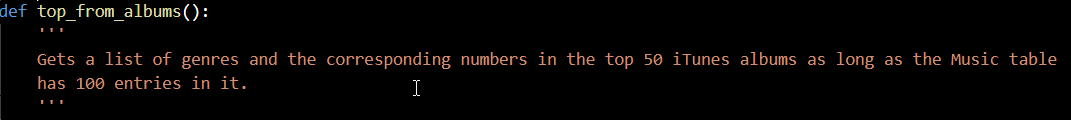
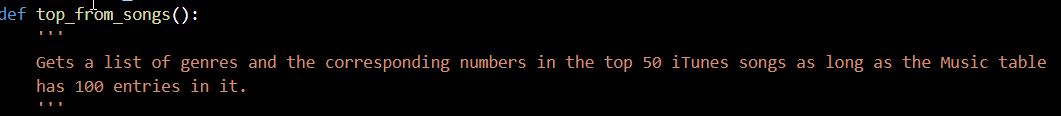
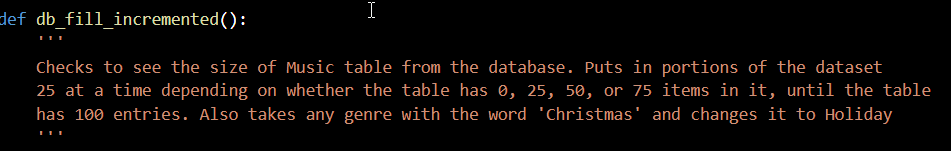
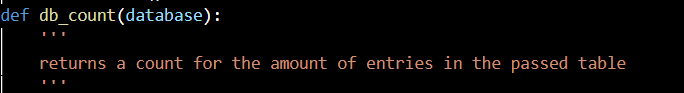
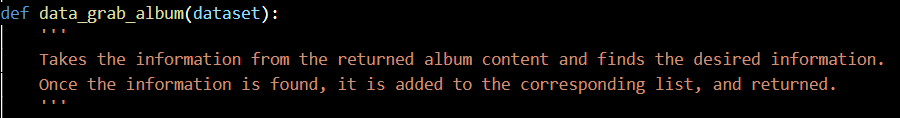
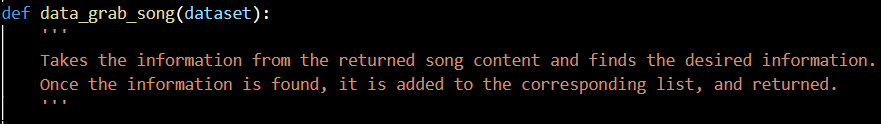
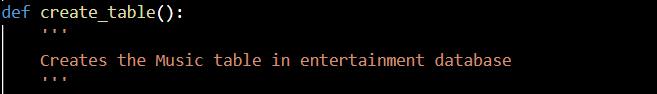




Itunes functions (Will):







**Resources**

| Date | Issue Description | Location of Resource | Result |
| --- | --- | --- | --- |
| 11/24 | Needing instruction for using Reddit’s API via Python | [Introduction and Basics - Python Reddit API Wrapper (PRAW) tutorial p.1](https://www.youtube.com/watch?v=NRgfgtzIhBQ) | Was able to successfully use the Python wrapped Reddit API to gather, query and analyze data from Reddit |
| 12/5 | Plotly Express wasn’t working, needed a different Plotly solution to make a scatterplot | https://plotly.com/python/ | Was able to use a GraphicObject to create a scatter plot I then further customized |
| 12/11 | Needed a way to add id numbers to table entries when the data was added to those entries with a function that used a \*i argument (coded by | https://www.w3schools.com/sql/sql\_update.asp | Came up with the idea of updating the table after the fact, and used this link to learn how to do this and used W3 to refresh myself on how to do other things. |
| 12/12 | Had more genres for music than there are colors in the matplotlib default colormap. Had to find a way to get more colors onto the chart without reusing them. | https://matplotlib.org/stable/tutorials/colors/colormaps.html | Found a larger colormap and use the cm module of matplotlib to implement it. |
| 12/1 | Apple music data trees were pretty deep and could be confusing. | https://codebeautify.org/jsonviewer | I used a json viewer so that I could see all the information and find the correct paths to what I needed. |
| 12/11 | Needed a bit of a refresher on database sql queries and syntax. | https://www.w3schools.com/python/python\_mysql\_where.asp | Went to the w3 schools website and read the tutorials. |