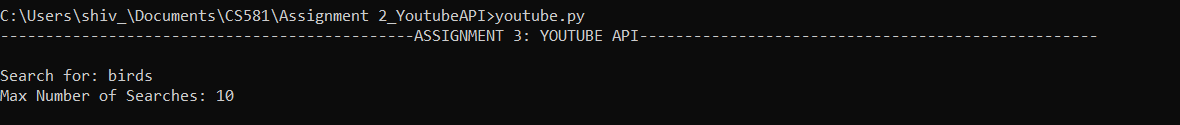
Assignment 2: Using the Google APIs to access YouTube Data

**PURPOSE**

The purpose of this program, youtube\_api.py, is to retrieve data from YouTube API and use that processed data to fulfill three types of analyses. With the YouTube data, you can add YouTube features, and manipulate data fields related to the videos or playlists. Using the YouTube API, you can gather the results of the videos by inputting the search term and the number of results.

**INPUT**

When running the program, the user is asked to input a search term. The user inputs an appropriate search term. After inputting the search term, the user comes across another command, which is to input the maximum number of results the user wants. After the user inputs the number of results, then the user gets the output, which is essentially 3 tables. Here’s an example of how it looks:

**OUTPUT**

The output of the program contains a csv file that consists of the specified raw data that was retrieved for each of the videos, along with a header that includes the data fields. The second part of the output is printed in the terminal of the 3 analyses which started with the list of videos, the highest like percentage, and the highest number of comments. All of these are printed in the A computer screen capture

Description automatically generated with medium confidenceform of a table. Here’s an example of the output:

**WHAT THE PROGRAM DOES**

The program starts off by asking the user to give two inputs. First, the user has to write what term they want to search for. Next, the program asks the user to write how many results the user wants to see. From the first analysis, I created a function called search(), each data field from the YouTube API are retrieved for each video result. After it retrieves all of the necessary data, it gets written to the .csv file, and the results get arranged into a table with headers indicating each field of the video result.

For the second analysis, highest\_likes\_percentage(), it uses the raw data retrieved from the first function and iterates through the table. A new table was made so that a new column can be added, without modifying the table from the first function. The new column would be specifically for like percentage, which takes the number of likes (for each video) divided by the number of views. Each like percentage is sorted by the highest percentage, gets rounded to 3 decimal places, and the numbers get formatted with commas. Then, a copy was made of that table was made so that data fields like the number of comments, duration, and video Id can be removed. Lastly the data gets formatted into a new table, with the data fields being only Like Percentage, Views, Likes, Title.

For the third analysis, highest\_number\_comments(), the steps are similar to the second analysis. A new table is create so that I don’t have to worry about modifying the raw data from the first function. After appending each video results to the table, then the table was sorted using the comments’ results. Then, a copy was made of that table was so that data fields like the number of likes, duration, and video id can be removed. Lastly, the data gets formatted into a new table with the data fields being only Views, Comments, and Title.

Here's a screenshot of the program: Text

Description automatically generated

**ADDITIONAL INFORMATION**

Text

Description automatically generated I was able to do the first analysis, but as I got to the second analysis, I started having trouble getting the video results to print out, as well as getting the Like Percentage to round to 3 decimal places. Another thing I noticed is when I try to get the program to work on the terminal window, I only write ‘youtube.py’ instead of ‘python3 youtube.py’.