

# FINAL PROJECT: COMMENT FINDER

## EXECUTIVE SUMMARY

### **Background:**

This semester I have been working on a website idea that focuses on user comments. There are websites that focus on the news, pictures, videos, jokes and micro blogs, but none really on comments. Many people spend a ton of time reading and writing comments, but all the commenting systems used now could be improved. We believe people want to hear and be heard, and that there is a ton of good comments that could be crowd-sourced in the right system.

This website would seek to be that improvement by allowing to users to comment on anything (videos, articles, pictures, etc). Currently we are in the validation stage working on finding out what would add the most value to users of this theoretical website. While experimenting with different comment filters, voting systems, rankings and incentive systems, we need to have a database of comments to use for tests.

### **System Overview:**

The purpose of this system is to have a convenient way to scrape highly scored comments off of websites and store them for use in testing and providing content for the future website. The two websites to be scraped are YouTube and Imgur. Both are extremely popular websites that host videos and pictures respectively. Each of them boasts a large and active user comment system. Some of these comments are entertaining or enlightening, while most are not. This system does the following:

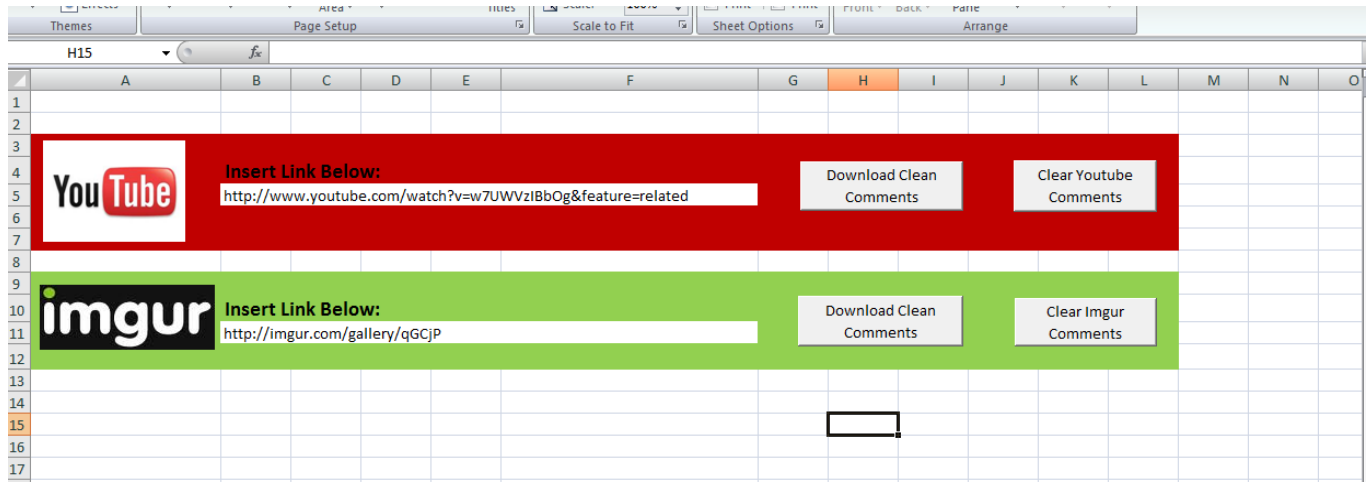
- Accept URL's from either website as inputs.
- Let the user know if the webpage has no comments.
- Parse through the data on the webpage to find the text, author and score of each comment.
- Filter the comment text to eliminate profanity and unnecessary links.
- Return on the data sheet the title, link, and filtered info for the top filtered comments of each webpage.

- Clear and automatically reformat the database at a click of button.

## IMPLEMENTATION DOCUMENTATION

### Inputs

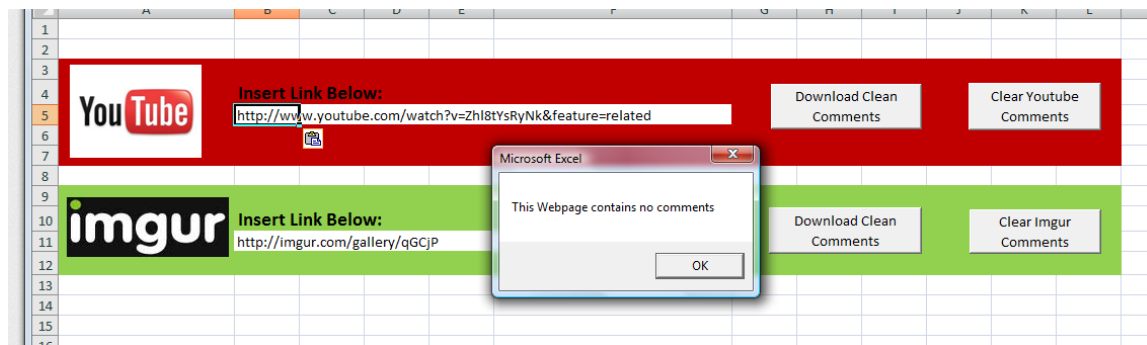
The input system is very simple.



A separate space is provided to insert links to webpages from each website. In order to make the system more clean and organized, the sub procedures and databases for each website have been separated. Once a valid link is entered, the "Download Clean Comments" button launches the web agent and starts the program. The web agent saves the source code of the webpage

### No Comments

Sometimes on YouTube, the comment section of webpage is closed and there are no comments to harvest. When this happens the program will return the following message:



Before anything is entered in the database in the next tab, the sub does a search using the get.Text and instr functions to determine if the comment section is closed. A closed comment section sends the message and shuts down the sub procedure.

## **Parsing**

Once the sub procedure determines there are comments to be harvested, the parsing begins. Using a Do loop and the web agent's getText and moveTo functions, the system goes through the source code pulling the author, score, and text of the top comments as well as the title and link of the picture or video.

Luckily, YouTube and Imgur organize their comments so the highest scored ones are at the top. This makes getting the best comments and ordering them easy for the system. If so desired though, the system could easily set minimum scores for harvesting. Usually the sub procedure as it stands now pulls ten or less of the top comments from the page.

## **Filtering**

Filtering was primarily for removing unwanted links and profanity. We are after the actual content of the comments, not the embedded links. However, most of these comments have the code for one and sometimes two links in the middle of the text portion of the comment. Three different IF statements and several Len, right, left, and instr functions were employed to root out the unwanted links.

The future website plan includes several filter options to improve the users' comment reading experience. One of the biggest issues is filtering out profanity. Filtering for "common" profanity in this database was accomplished by using many Replace functions. The system is able to pick up most instances of profanity and delete them.

## **Returning Data**

Once the data has been pulled and filtered, it is placed into the two separate databases on the "Datapage" tab shown below.



## DIFFICULTIES ENCOUNTERED

While building this system I encountered several difficulties. Some of the big ones are listed below:

- One of the main difficulties I encountered was accounting for the differences among webpages. Even just among YouTube webpages there was enough differences in the source code to trip up the sub procedure. It took a good amount to find formulas that would work for each of the pages.
- In the text body of the comments, the links in the middle of the text required quite a bit of thinking and testing to make sure just the relevant text was pulled. At least for me, the formulas were complicated e.g. `comment = Right(comment, (Len(comment) - ((InStr(1, comment, "A>") + 2))))` –that is a lot of parentheses
- Another difficulty was catching all the profanity. It was awkward to make the long list of replace functions, but after testing it on rap video webpages I quickly found out that the list wasn't long enough. Also those commenters are not always the best spellers and are fond of writing in ALL CAPS. Since the replace function is case and spelling sensitive it took a little more effort.
- I also had trouble formatting cells and worksheets through VBA. Up till now I have mostly ignored that aspect of the language, so there were growing pains learning how to do that.
- Also with formatting, it took me awhile to get the database to go where I wanted it and not overwrite previous entries.

## LESSONS LEARNED

I learned more on this project than any other including –

- A ton about the web agent class. I played around with the class constantly while doing this project and I became much more familiar with it.
- More on how different webpages are set up. Looking for hours on end at dozens of html source code pages I feel like I have a better understanding of them now.
- How to parse like a pro. Before I was shaky at best. This system gave me a ton of practice and I learned all kinds of tricks by googling and testing on how to get it done faster.

- How to use long complicated formulas using the Len, right, left, instr, and other functions. At first it was overwhelming to have all these functions with functions, but I learned to break it up into little steps I could grasp.
- As mentioned above in the problem section, I learned all about how to format cells, do colors, fonts, alignments, and everything I knew was possible but had not learned yet. It was my first time really writing and using the “With” format.
- Finally, I learned out to find a problem, come up with a strategy, plan out a solution and execute.