

CPSC 473-01: Web Design

Assignment #4

1. Our team's project is to create a community-based Twitter gateway which will allow user groups, or organizations, to submit Tweets with others that are part of the group. Our website will allow single users to submit a Tweet for others to view prior to being submitted to Twitter. In order to publish the Tweet, it must receive a certain number of votes from other members of the group.

A post which meets a certain criteria will be published to Twitter — this criteria to publish a Tweet is still under evaluation. We're looking into requiring a certain percentage of the group's total population to vote on the unpublished Tweet to qualify, and a certain percentage of up-votes vs. down-votes to publish. Another way to organize our project will be to have a Tweet Topic posted and have users submit potential Tweets related to said topic. The Tweet with the highest number of votes for the Topic will be published to Twitter.

2. & 3. List of potentially useful APIs and explanation as to why it could be useful:
 - **Tweet Sentiment:** Tweet Sentiment would allow us to categorize comments based on their sentiment (*negative/neutral/positive*). This could give us the benefit of allowing users to filter out negative Tweets and focus on positive ones — this could help allow groups to shape a more positive appearance on Twitter: <https://www.tweetsentimentapi.com>
 - **Feathr.it:** The Feathr.it API will allow users to automatically reduce the length of potential Tweets with the click of a button from our website — this will help users stay within Twitter's 140 character limit: <http://www.feathr.it/apidoc>
 - **Reddit:** This is the api for reddit's voting. On reddit people will vote on a particular post or comment. People can either vote up or down. When they vote up the post's number will go up and if they vote down then the number will go down, so a post can have negative votes. I thought of this as useful for our assignment since people will post and other users will either like or dislike the post. The owner will be able to see the amount of likes and dislikes a particular post got. <https://github.com/reddit/reddit/blob/master/r2/r2/controllers/api.py#L1946>

Wednesday, March 11, 2015

- **YouTube:** This api is for youtube. This url in particular shows us how they handle the ratings on a video. On YouTube people will post a video and people will either like it or dislike it. Unlike Reddit, YouTube will show how many likes and dislikes and not take away likes depending on the likes. This could also be helpful for our assignment since we could see how many dislike a particular post got in compared to the likes.
https://developers.google.com/youtube/2.0/developers_guide_protocol_ratings
 - **Strawpoll:** Strawpoll.me is a website that allows users to build a poll by listing a topic and listing poll options. This allows other users to vote on them. In short, people can create a list and have people vote on the item they like most. The idea of this website applies to our project since we want group members to post Tweets and have other members vote on the Tweet to determine if it gets published. The Tweet with the most votes will be published.
<https://gist.github.com/xxOrpheus/7ecee52b3292aacb649c>
4. I feel like the most useful API for our project is actually one of the more simplistic ones — Feathr.it would be a very useful tool for our users and integrating it directly into our website will allow for an excellent quality of life feature. The demo can be found here:
https://github.com/jmovius/CPSC_473/tree/master/04_031115/FeathrItDemo
 5. We decided to work with Feathr.it API since our assignment deals with posting to Twitter and Feathr.it allows users to more easily stay within Twitter's 140 character restriction. Feathr.it will let the user tweak the post before they attempt to publish it to Twitter, this concept can be beneficial to our project since we could allow the our users to potentially express more information in the same amount of characters.
 6. Potential issues of working with the Feathr.it API:
 - One potential issue of using the Feathr.it API is the user may receive a shortened Tweet that includes too many symbols and makes reading the Tweet a chore. The result could be something that is more difficult to read and may frustrate the reader. To combat this issue, we can implement an "Undo" button (*this is actually implemented in the demo*).
 - Another issue with using this API is the API is not a local library and information is served by Feathr.it; therefore, if the Feathr.it server goes down, we would have to account for Feathr.it server errors, or a lack of response. To avoid potential issues, we can set a timeout or create a handler to gracefully take care of external server errors.

Wednesday, March 11, 2015

7. My experience using the using the Feathr.it API was excellent — the API is very easy to use and the results are very easy to handle. The benefits this API can provide are great by allowing users to quickly and easily edit their Tweets to shorten them down to 140 characters. If the API receives a string less than 140 characters, then it will return the original, unedited string. It is important to note the API will only edit the string until the total size of the string is under 140 characters; therefore, a string starting with “To be this...” and is 141 characters long, the API response will contain the string “2 be this...” and it will be a total of 140 characters. Based on my experience, the benefits and ease of use make this API an excellent addition to our Twitter gateway project.
8. Feathr.it only has one function call, so the API is very simplistic; however, this function does have multiple parameters. Since Feather.it can replace words with symbols, the results can frustrate some users. Including check-boxes to turn on/off each parameter, such as whether or not to include symbols in the response, would be ideal.