Team Stack Overflow

Spit.fire

github-repository-url

Team Overview

Team Member Name	Github Usernames
Sahil Chaturvedi	sahilc0
Ri Neng Huang	rineng
James Michaels	J-Mike
Joshua Canuel	josh356
Nischal Tamang	nischaltamang

Team Writeup

Overview:

There hasn't been too many drastic changes to the proposal. Spit.fire crowdsources hip hop instrumentals and lyrics to find the next top artist. Users submit instrumentals, others submit lyrics to go with the track. Then the community decides which tracks and lyrics are the best, through an upvote system. Top voted instrumentals and lyrics are prominently featured in the main page. Eventually, the contest page will enable brands to set up sponsored contests with a curated instrumental, and provide a description of what lyrics to rap about. Winners will be able earn a prize from the sponsor/host.

Design Overview:

Spit.fire revolves around music. Producers will be able to upload an audio file (1) of their best instrumentals and lyricists can upload their raps/lyrics (2) to an instrumental that interests them. Other users can upvote (3) these instrumental tracks and lyrics. Spit.fire is also partially a social networking site as users can interact with each other using comments (4) on lyrics and by recommending others to "spit" on a certain track. The site will also track users' personal data (name, images, location, social links) (5).

Our data models are Track, Artist, Lyrics, LyricComment, TrackComment, Genre, & Sponsor. Each Track has a 1 to many relationship with Lyrics, TrackComment Genre. Our important URL Routes are '' (index.html), 'soundtrack' (soundtrack.html), 'profile' (profile.html), 'lyrics' (lyrics-sync.html), 'upload' (upload.html), and 'contest' (contest.html).

Problems/Successes:

Successes: We worked very well together as a team, and coordinated in person meetings regularly as well as consistent communication over the chat app *Slack*.

Problems: Minor issues with Github merges and conflicts.

Future Improvements: We will learn how to merge better and create branches to avoid conflicts when pushing and pulling from Github.

Sahil Chaturvedi
Worked on the Sponsors model: updated the name and description.
Worked on the contests.html template page, replacing the static data with variables for the views and model.
Updated the views.py with information for the contests.html page. Sponsor name, and sponsor description added.
Modified the upvote button on pages to make it dynamic.
Helped build this write up documentation.
16%

Nischal Tamang
I mainly worked on anything related to the lyrics-sync section. So I updated the lyrics-sync html file to support django's contexts and wrote the corresponding function in the views.py file. I also added a default/sample audio to all the audio players on the page. Then with guidance from James & Ronny, I reduced redundancy of the context provided in views.py for the profile page. I did this by feeding in a track object in the context and calling required fields in the template itself. I also implemented a django for loop in the profile.html file to remove html clutter.
16%

Ri Neng Huang
I created view functions for index, and profile pages as well as the base layout for the other views. I worked a bit on defining attributes in the models. I created the Base template and then have the other pages extend it. Made navbar buttons dynamic so that they take you to the proper pages. In addition, I also created mock data for the database. I implemented URL mappings. I populated the index page, some of the profile page, and some of the soundtrack page by making queries to the database.
26%

James Michaels

I completed Part 0 in full and wrote a 90% complete version of the models.py file. On Nischal's recommendation and using a helpful link provided by Ri Neng, I spent some time trying to improve my usage of git, exploring branches and how to create and merge them, and then spent some time trying to relay my experience back to the group. I wrote some very minor revisions for the group section of this file.

I explored how to reduce redundancy and hard-coding between the context provided in views.py and the associated .html page as I worked on the view for track and the dynamic variables in soundtrack.html. I also did some revisions of the base template html file and the file structure so that images would be correctly dynamically loaded and would actually be in the place where the functions were trying to get them from.

26%

Josh Canuel
I created the data model diagram using draw.io. I spent time trying to fix part of the nav bar not functioning properly, and generally searching / testing for bugs. I cleaned up the upload page.
16%