

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 any 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*. We also successfully made use of branches on GitHub to avoid merge conflicts.

Problems: Minor issues with Github merges and conflicts. We may have overestimated the amount of time we have left to work on the project.

Future Improvements: Working diligently throughout the given time to complete the project.

Team Choice

We plan to have the music and lyrics be synced together. We will also try to speed up the mp3s which seem to make the index page somewhat sluggish. We will allow users to upload their own profile picture instead of just getting the default one.

Sahil Chaturvedi

I helped with working on the model, view, form, and template of the “comments”. I also helped with the documentation.

15%

Nischal Tamang

I worked on the upvotes on the website. I wrote a JavaScript file to send a ajax request to our database on every click of the upvote. Then I wrote corresponding functions in Views.py to update the upvote count in the database. I also made the home page dynamic by feeding in an array of tracks & using a for loop in the template. I then added functionality to the search feature to allow users to search our database. 15%

Ri Neng Huang

30%

Helped in implementing user authentication so that upon creating a User, an artist is automatically created with fields filled in with the supplied user information. Implemented url mappings so that each track and artist gets their own unique url and renders the designated html with the information of that artist/track. Created view functions for retrieving track informations, artist information, and profile. Created upload view function so that it creates a track in the database as well as storing the uploaded track mp3 into the database. Created trackform and helped in creating forms.py. Provided audio functionality in homepage. Each audio corresponds to the proper track.

James Michaels

25%

Created initial setup by porting over locallibrary files and functionality. Worked on rough user login form and database interaction. Tried to catch other group members up on how different parts of the framework interact with each other (such as how views.py plays middle-man between forms & html and the database) so that their efforts would be more effective. Brought the profile page up to date and made nav bar display user’s first name and profile picture, or login link if not logged in.

Josh Canuel

15%

I spent time searching for bugs/fixing them. I made the page for the lyrics dynamic, and made the form to be able to post lyrics.