**Project Report**

Jacob Dumford

Tyler Sammons

**SUMMARY**

Our project is a Music sharing website similar to soundcloud. The site allows users to browse audio files that any other logged in user has submitted to the site. The audience is anyone who likes to make, listen to, or share music.

**COMPONENTS**

The components of our project include:

· A main page where mp3 files can be played by any visitor to the site

· A selector that allows users to sort by genre

· A selector that allows user­s to sort by either the time uploaded or by popularity

· The ability to create an account or log-in via Facebook

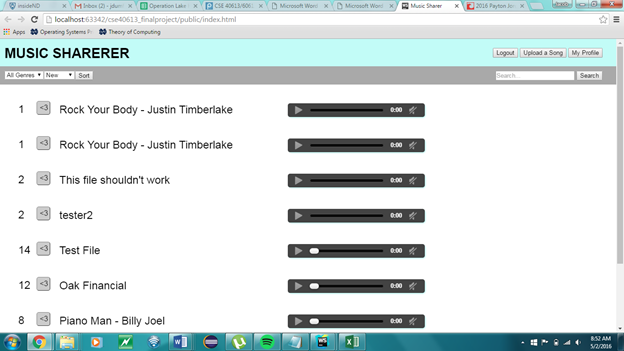
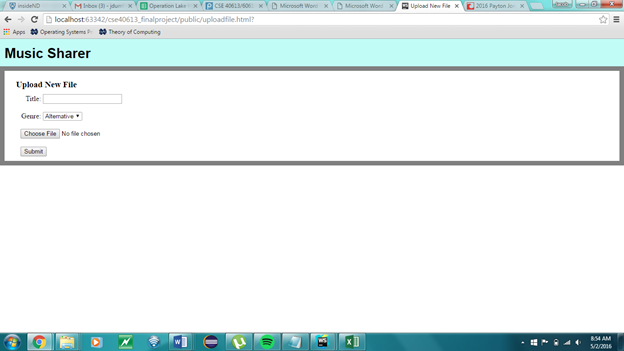
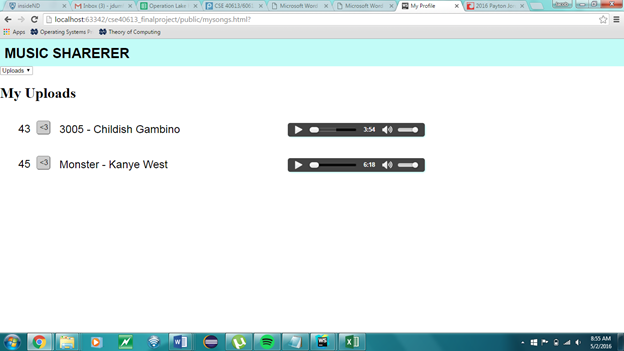
· The ability for logged-in users to up vote an mp3 file once

· The ability for logged-in users to submit an mp3 file with a description and a genre

· The ability to search for a file

· A “My-Profile” section for logged-in users to view the files they have submitted and the files they have “liked”

**VIEWS**

There are 3 pages that make up the basic structure.  
  
Home page:  
  
   
Upload page:   
  
My profile: 

**FRONTEND LIBRARIES AND FRAMEWORK**

The libraries we used included: JQuery, the Firebase API, and the Amazon AWS API.

We used the JQuery library to help build the upvote feature and display the components of our web page properly, much like the way JQuery was used in the FeedND webpage.

The Firebase API was used to upload our data to our firebase app and to deploy the website, and the Amazon AWS was used to upload the mp3 files to our S3 bucket.

**BACKEND SETUP**

For the backend setup, we used Firebase to store the majority of data with AWS to store the audio files themselves. In Firebase we store a list of songs which each have a unique id as well as the filename, title, genre, likes, and a timestamp depicting when it was uploaded. We also store user information such as name, email, phone number, address, etc. as well as a list of all the songs that user has uploaded and all the songs that user has liked.

**DATA DICTIONARY**

Our data is stored on Firebase in two separate instances. The first is called “songs,” and the second is called “users.”

Each song is stored using a “song id”, which then contains five separate pieces of data: filename, genre, likes, timestamp, and title. The filename is encoded using UTF-8 character encoding so there are no difficulties retrieving the file from AWS.

Each user is stored using a “user id”, which then contains “delivery address,” “likes,” “uploads,” “email,” “name,” and “phone.”The “likes” and “uploads” then contain a dictionary of file id’s which the user has liked or uploaded.

Here is an example of a song element:

songs

-KGi4ocFcSsYFHbSbVY0 (Song id)

- **filename**: ”MDYgTW9uc3Rlci5tNGE=”

- **genre**: “hiphop”

- **likes**: 45

-**timestamp**: 1462135899857

-**title**: “Monster – Kanye West”

Here is an example of a user element:

Users

-5aa1cb86-3832-4295-bd73-f7c519ff4ad

-deliveryaddress

-**city**: “prospect heights”

-**state**: “Illinois”

-**street**: “522”

-**zip**: “50060”

-email: tsam@nd.edu

-likes

-**KGi4ocFcSsYFHbSbVY0**: “Monster – Kanye West”

-**etc…**

**-**name: “Ty”

-phone: “8476024243”

-uploads

-**KGifY7hT6dhVMq\_QEm0:** “test file”

-**etc…**

**DESCRIPTION OF CONTRIBUTIONS**

Tyler:

Handled uploading files to AWS, wrote the like and search functions, retrieval from AWS/ firebase, did most of the styling

Jacob:

Wrote the upload function/stored info in Firebase, wrote sorting functions, created myProfile, limited users to liking a song once.

**USER GUIDE**

First type in the url:<https://finalproj.firebaseapp.com>. The first thing a user will see is the main page with every file listed, organized by most recently uploaded. Without logging in, a user can play these files by hitting the play button, sort the files by choosing their preference from the dropdown menus. Once they choose there genre and sorting preference, they must click the “sort” button which will display the proper files. Any user can also search by typing a string into the search bar and clicking the search button or hitting enter. If the user wants more functionality than this, they must create an account by either clicking “signup” and filling in the info, clicking “login” if they already have an account, or by logging in with Facebook. Once logged in, the user can now up vote files, upload files by clicking on the upload button, and go to the “My profile” section which allows users to toggle between files they have uploaded and files they have up voted.

**COMPARISON TO RELATED WEBSITES / APPS**

The most similar website to ours is soundcloud. The basic functionality of the two sites is the same. One thing we added that soundcloud doesn’t allow is the ability to sort by genre. As we continue to add functionality and styling, we would differentiate our site more from soundcloud.

**FUTURE WORK**

-Obviously, we will need to come up with a name for the site.

Otherwise we would definitely work on the design and user-interface of the site. We spent time trying to fix the header in place, but had some difficulties with CSS.

The next component we would want to add is the ability to create and store playlists.

We could work on building a recommender into the site.