**Kamikazee Karaoke**

Kamikazee Karaoke will be an app that allows users who wish to sing a karaoke song to do so either alone, or in a group setting of some sort. The app itself will consist of two databases and the use of several API’s. The databases will include two tables, possibly more depending on whether stretch goal criteria is met, or not. The first table will be Songs, possibly with a different, more suitable name, but at this point I can’t think of one. It will consist of ID as the primary key, Artist Name, song title, genre and year. Users will be able to select genre, artist or year in order to refine their song selection. Once these options are selected a user will be able to view songs, or press the Kamikazee button, randomly selecting from those selected song id’s. There will also be a table called users which will contain username as the primary key, email, password, first and last name and date of birth. There is also the possibility of exchanging date of birth for security questions and answers. The DOB and Security questions will be used for resetting a password when a user forgets.

The API’s used will for sure be youtube’s API for the video of the karaoke song, possibly for the music video as well. After that I intend to use a lyric API to pull the lyrics into the application. I’ve found two, one is lyrics.ovh and the other is MusixMatch. While MusixMatch was my first choice, after research it was discovered the free version is very limited. Lyrics.ovh has a full library of lyrics free, at least from what I’ve seen. The API is also very easy to use, it’s a simple GET request that requires an artist and title. It responds with JSON and a single output, lyrics. Tested the output and have seen it pull for the example song, even for songs I know have lyrics and exist. It takes a bit of time, so I would definitely need some form of loading screen to accommodate the load time. It also struggles to find some artists and songs. It may not be the best option, but it’s one of them.

This app will include a few options for selecting the song. First a user will be able to press the Kamikazee button and a random song will load. This will be a song based on an ID that is randomly selected between 0 and the length of the songs table. When loaded it will produce the karaoke video, the song video and lyrics. It may have an option to instead have just the song, if the song video isn’t an option. It depends on the use of Spotify OR Apple Music APIs.

The app will also allow a user to select groupings of songs, or even just select songs themselves, best on artist, genre or year range. Selecting any of these criteria will allow the user to view a list of the songs, OR once again select Kamikazee and randomly select a song from the list of groupings. At that point the process will begin again, loading songs, videos, and lyrics. At this point that is the process for the app, but I have considered, and am still considering, the option of a user selecting Sing, or Choose Again before the app loads, giving the user a bit more control over their choice of songs.

The user will be able to view a list of the last X songs they’ve sung in their user account. This will be done by adding to a new table, so the database will require one more table called sung. The table name may change between now and creation. It will consist of ID as primary key, username and song ID. The user will be able to view a link to the song, and it will be displayed as an <a href=”/{artist}/{song}>{artist} {song}</a> element. The finer details are still under construction.

Stretch goals for this include web sockets that allow users to create rooms where they can sing and share their version of these songs. This will require a lot more, but essentially we would need access to camera and microphone. This will be used to share the users performance. Possibly could save to a database, but that is less pressing, the user could essentially save it to their own phone, instead of on the app. Details will be filled in if it comes to this point.