



## Introduction

SongSurf is an interactive website that is able to recommend artists, albums, and tracks to users based on what the music they input. It allows users to like and save these recommendations with an easy-to-use user interface (UI). Each user can make their own account to save the recommendations so the website can support multiple users. Using a wide variety of technologies such as APIs, databases, and web frameworks we are able to present this simplistic and practical tool that music fans or anyone trying to find some new music can use.

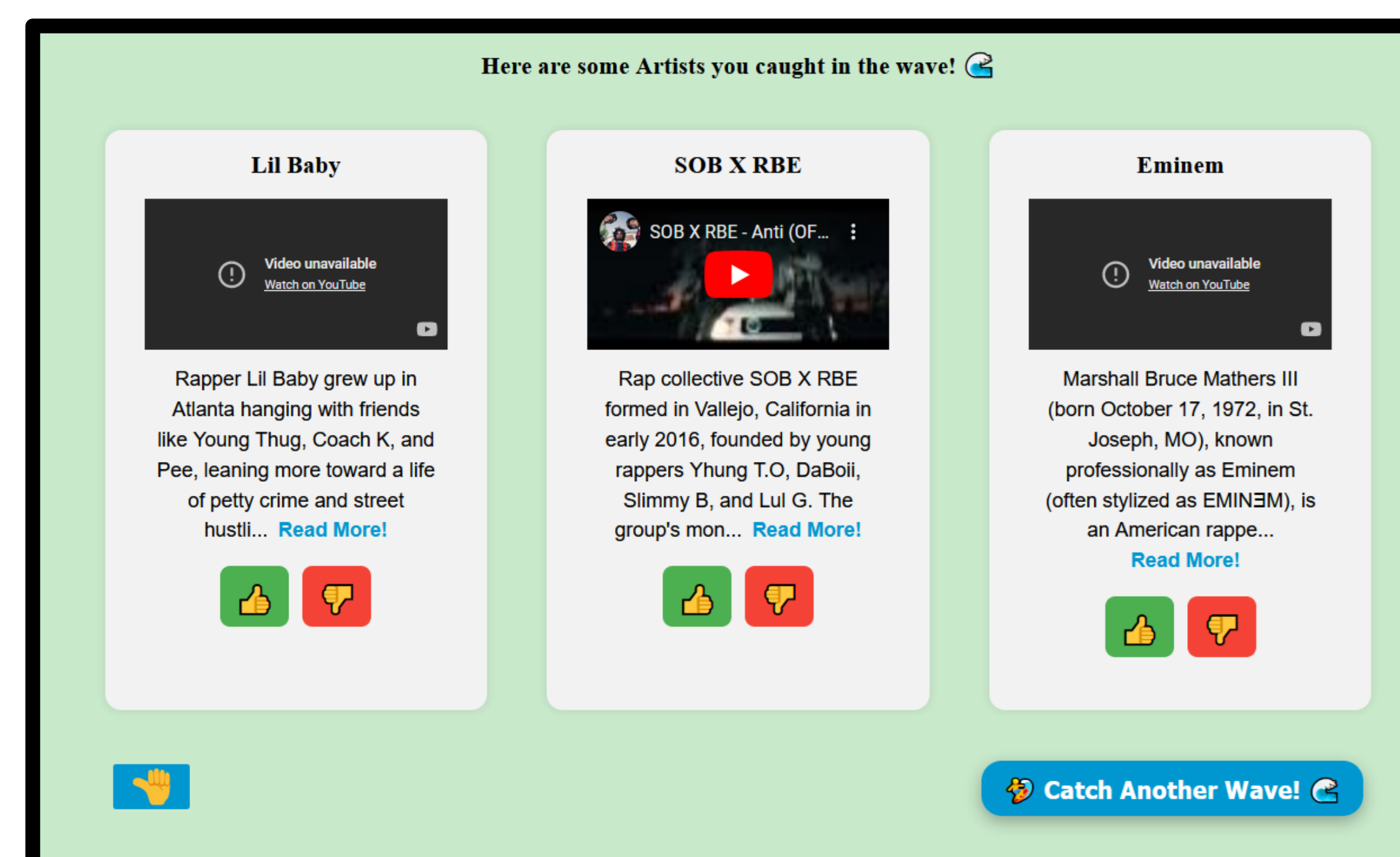
## Architecture

- **Languages:** Python, HTML, CSS, Javascript
- **Framework:** Flask
- **Tools:** SQLAlchemy
- **API:** last.fm, Spotify

A screenshot of the SongSurf website's "Find Me" section. It features a search bar with three radio buttons: "Artists" (selected), "Albums", and "Tracks". Below the search bar, there is a section titled "Enter up to 3 Artists you love:" with a note: "Note: if you chose albums or tracks enter the artist's name on the right". There are three input fields labeled "Artist name:" with numbers 1, 2, and 3. At the bottom, there is a green button labeled "SURF" with a surfboard icon.

## User Interface

SongSurf's clean, minimalistic, and pleasing design makes music discovery easy for users. The UI provides easily navigable buttons, and responsive system feedback should the user make mistakes. Initially, the user will choose what they want recommended and input their favorite artists/albums/tracks. Then, the users are given a selection of personally picked recommendations. The selections are displayed in detail with a description and preview of the music. The user can easily like or dislike an artist/album/track and see where their favorites are.



## Database

SongSurf boasts a relational database to quickly load information to a user. From information about albums, artists, to tracks. Including but not limited to: name, image, url, description, how a track is connected to an album and to an artist and vice versa. The database stores all of the likes, dislikes, and recommendations for different users. This database utilizes API Caching, to reduce the number of API calls, improve performance, and improve reliability.

## Algorithm

SongSurf utilizes the last.fm API to leverage a vast database of user-aggregated data from millions of music enthusiasts.

Dozens of tags are associated with artists, albums, and songs, forming an extensive metadata repository sourced from the last.fm user community.

SongSurf utilizes these tags along with what the user liked or disliked to deliver personalized music recommendations that users will love.

## Lessons Learned

- I. A well thought out plan with design specifications before implementation is essential to deliver everything on time.
- II. Prepare for unexpected changes, so be open-minded and flexible when developing.
- III. It is good to continually test the project through development to ensure a high quality product.
- IV. Constant clear communication amongst group members is required for the project to stay on track to a successful deployment



UNIVERSITY OF THE PACIFIC  
Shahbaj Sohal | Patrick Nisperos  
William Balbuena