

legendary-tribbles: Ayman Habib, Eric Sohel, Karen Shekyan, Jian Hong Li

SoftDev: Pd 8

P04: What Does Your Data Do? Design Doc

Target Ship Date: 05/19/23

Idea: Create a data visualizer that takes in a user's Spotify Account to return their listening data, provide song/playlist recommendations, and even use text analytics to write new lyrics.

Breakdown of Tasks & Member Assignments:

- ☐ API and Python implementation (Ayman, Jian Hong)
- ☐ Python Flask work (Karen, Jian Hong)
- ☐ Account signup and login (Karen, Jian Hong)
- ☐ HTML (Ayman, Eric)
- ☐ Bootstrap framework for styling (Ayman, Eric)
- ☐ JavaScript data animation/visualization (Ayman, Eric)
- ☐ Project management (Ayman)

Possible data to be visualized:

- User or artist's followers
- Audio features (e.g.: acousticness, danceability, etc)
- Popularity over time of genre/artist
 - A.k.a.: How fast an artist grew using their follower count over time
- Most listened to genres of a user

Databases/APIs:

Spotify API:

<https://developer.spotify.com/documentation/web-api/howtos/web-app-profile>

Musixmatch API for lyrics:

<https://developer.musixmatch.com/documentation/api-reference/artist-chart-get>

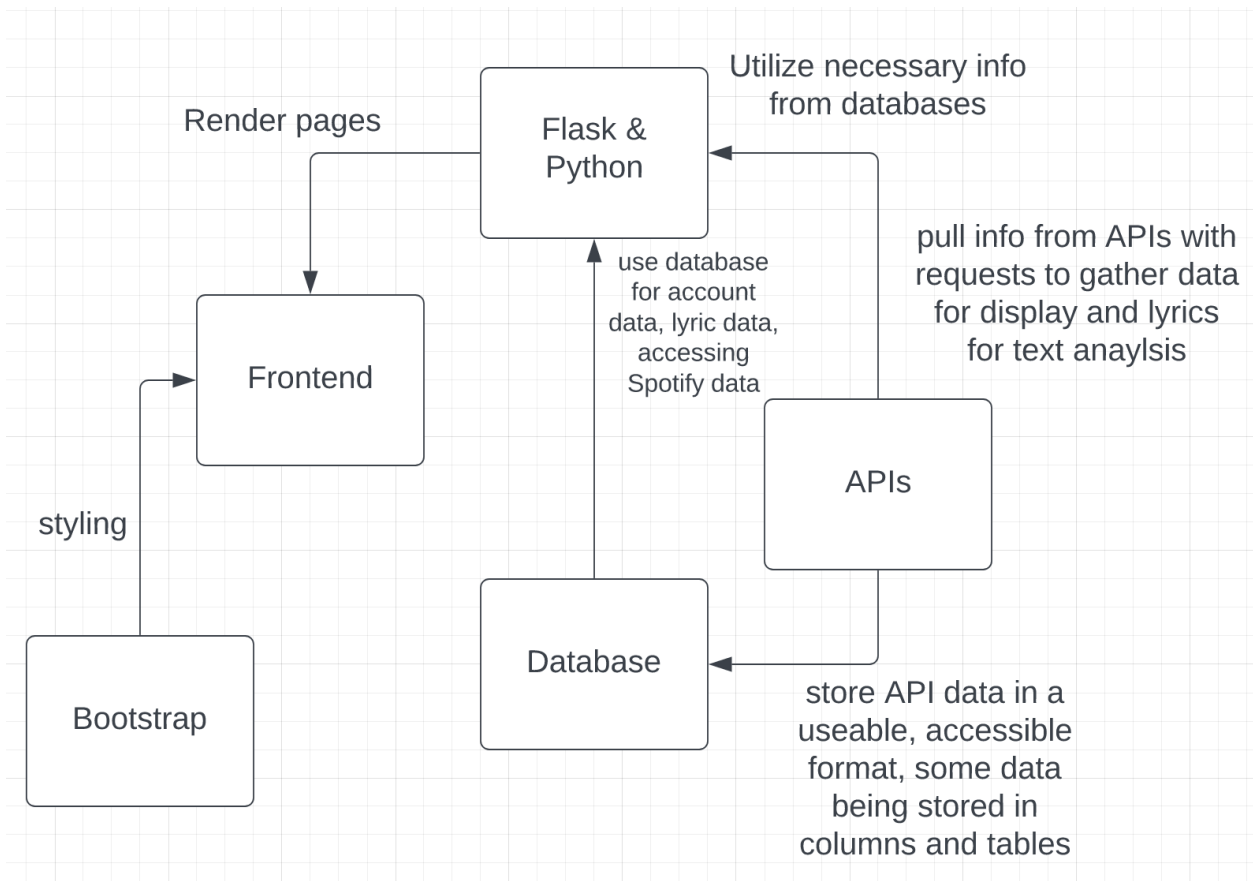
(Will store lyrics in local database due to quota)

Components:

1. Database
 - a. Stores user account information
 - Username, password, favorite artist, favorite song
 - b. Stores song lyrics and new song lyrics
2. Frontend
 - a. Display/visualize data to user
 - b. Allow users to input lyrics in order to generate new ones(?)

3. Flask & Python
 - a. Render frontend pages
 - b. Account login/setup
 - c. Database setup
 - d. Spotify API access
 - i. User data, artist data, etc.
 - e. Text generation functionality
4. Bootstrap framework
 - a. Make frontend pretty, intuitive, and user-friendly
 - b. Make data visualization cleaner
5. APIs
 - a. Provide user info (Spotify stats) and artist info

Component Map:



SQLite3 Database

- Tables:
 1. Primarily, we will use tables to store user login information

User ID (primary key)	Int
Username	String
Password	String
Spotify Username	String

We may amalgamate user info and make the user use their Spotify username and password as their user and pass for the site

2. Storing lyrics

Song ID	Int
Song Name	String
Lyric	String

3. Storing new lyrics

New Song	ID
Lyric	String

Frontend

- Bootstrap
 - Easily customizable components
 - Dropdown menu and progress bars
 - Grids
 - Form controls
- HTML
 - Will be used for templates to be connected to Flask

Roadmap:

- Access Spotify API and determine what data can be used and how
- Construct local database with song lyrics
- Build HTML templates, Javascript files, and flask web app
- Combine API and database into flask
- Work on animation/visualization in Javascript
- Bootstrap and CSS for aesthetics

- Testing and debugging

Sitemap

