

Project title: Fantasy Spotify

Project summary: We develop a game similar to Fantasy Football but for Spotify artists. Each artist will have a price that you can sign them. Users will be given a “budget” of 100 million to sign 5 artists. Every month, we assign points for each artist based on Spotify performance and users get however many points their chosen artists earned.

Description of application:

- What we want to do: Using Spotify data, we can rate the performance of artists using different metrics (eg. monthly listeners) and we will assign them performance points. The game will contain up to 100 of the most popular artists as part of the selection pool. Users who have the artist in their roster will earn that amount of points. There will be a leaderboard for users with the most points. This will be updated monthly after we rate the performance points of the artists for that month.
- What problem do we want to solve: Fantasy games have traditionally only been used in association with sports. However, we design a fantasy game for music enjoyers. This provides an entertaining way for people to support their favorite artists and make predictions on what songs/albums will be most successful. Many “super” fans can also use this as a tool to compare their favorite artist with others. It can also be used as a way to see how much your music taste aligns with others.

Creative component:

- Update prices of artists based on ownership level (need apscheduler, numpy, pandas etc.)
- Using time series graphs to visualize the historical price trend and update it daily. (need recharts, chart.js, flask, Spotify API, apscheduler etc.)
- Use Google trends to recommend artists to users (pytrends, google-trends API etc.)
- Use time-series weighted historical song sales to initialize price for each artists

Usefulness:

- What are the basic functions of your web application? Users can view their points, budget, leaderboard, trending artists, stats of artists they’ve chosen. They can also search for artists to make their roster selection.
- What can users of this website do? Users can pick their roster of 5 artists out of 100 using a “budget” of 100 million. They can modify their roster any time
- Which simple and complex features are there? Prices for signing artists will be determined by their demand and are updated frequently using a formula. We are hoping to converge to an equilibrium for how much “value” an artist provides.

- Are there any similar websites/applications out there: Yes there are many Fantasy games such as <https://fantasy.premierleague.com/>. However, nearly all of them involve sports and not music artists.

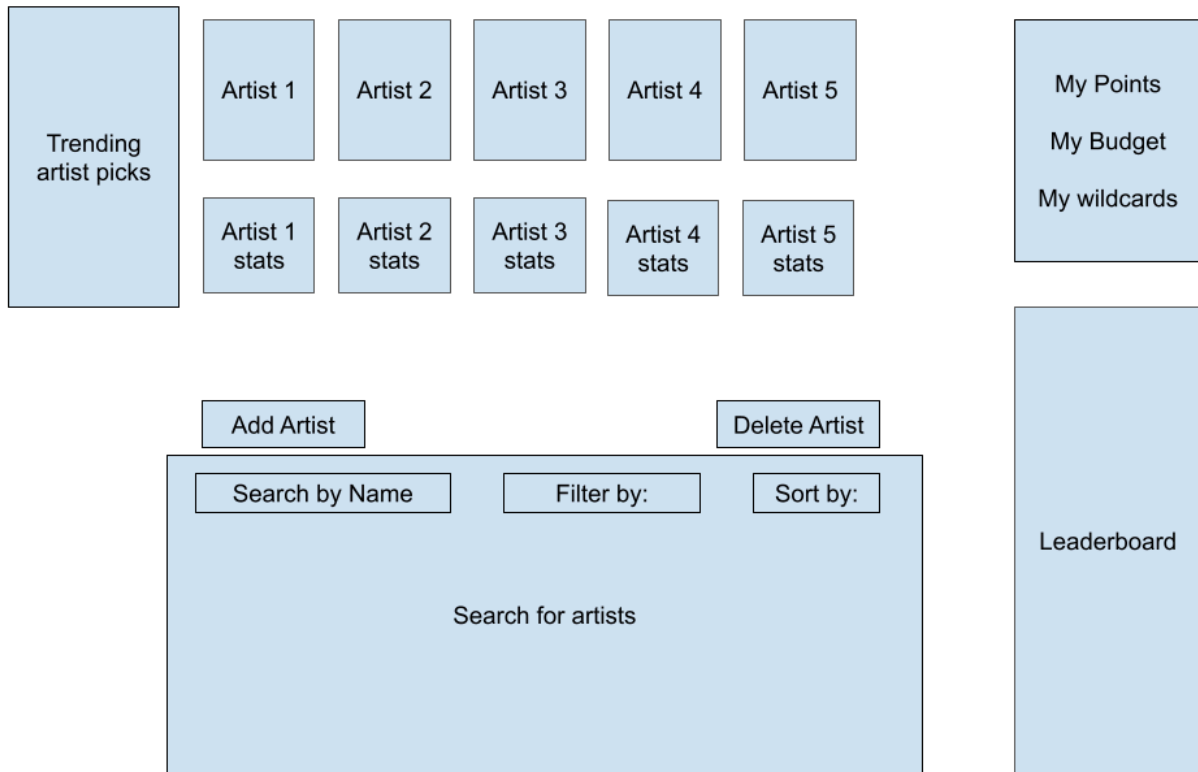
#### Realness:

- Spotify API - We retrieve data from Spotify using <https://developer.spotify.com/documentation/web-api> and will put it in a csv file. This will be done once a month to update artists' monthly performance. We will get various performance stats such as monthly listeners, follower growth, likes for the top 100 artists. Since it is only for the top 100 artists, the data size need not be very large.
- Twitter API or Google trends API to find trending artists, which will return the data in json format. Again, we retrieve this data once a month for the same top 100 artists as Spotify. This time we mainly look for how much they are mentioned on Twitter or Google.
- Datasets and API from chartmetric (<https://chartmetric.com/>), which have datasets of various formats. It is a similar collection of musical artists data but from various sources. Again, we all retrieve data for top 100 artists, but this dataset makes it much easier to get monthly information such as listeners, new followers, and other monthly engagement metrics.

#### Functionality:

- Users can create a roster and will be given a 100 million "budget"
- Users can sign artists within the budget
  - Users can filter by various criteria to find artists they like
- Users can "sell" artists in order to sign someone else
- Users can search and filter artists
- Users can see price updates for artists
- Wildcards (implement if we have time)- every year, each user gets to use one of 3 special wildcards to boost their points
- Users can see Spotify data of their chosen artists how many points they got
- Users can see a leaderboard
- Users can cancel an artist signing within a specific time if they change their mind

#### A low-fidelity UI mockup



Project work distribution:

Frontend:

- Website design and UI - Chris

Backend:

- Development of price change and performance point formulas - David
- Send/Retrieve data using API - Cyrus
- Update user's roster of artists, user's points - Shreyes
- Leaderboard - David

Software deployment - Cyrus