Spotify Draft Battle

*Anything written in italics means it was written in hindsight.*

Contents

[Initial Ideas 3](#_Toc180048854)

[The Problem 4](#_Toc180048855)

[My Solution 5](#_Toc180048856)

[Third-Party Influences 5](#_Toc180048857)

[Key Terminology (explained within the context of my project): 5](#_Toc180048858)

[Research 5](#_Toc180048859)

[Researching how to implement my idea 5](#_Toc180048860)

[Choosing what music platform 5](#_Toc180048861)

[Structure of my Web-App 6](#_Toc180048862)

[Step-by-step process 7](#_Toc180048863)

[Questionnaire 8](#_Toc180048864)

[Project Objectives: 12](#_Toc180048865)

[Researching how to implement my code 14](#_Toc180048866)

[Learning the Spotify API in JS 15](#_Toc180048867)

[Creating a UI 16](#_Toc180048868)

[Implementation into Python 21](#_Toc180048869)

# Initial Ideas

I want to have a choice between a variety of ideas so that I can choose the best out of a good bunch. I will use the following criteria to do this

1. Something that links into something from Computer Science theory.
2. Something that links into an academic interest.
3. Something that optimizes processes.
4. Something to do with my hobbies outside of school.
5. Anything else I want to do.

|  |  |  |  |
| --- | --- | --- | --- |
| Project Title | Brief Idea | Technologies involved | Disadvantages |
| Fact check Indexer | A program that indexes through databases that are used in investigations and fact checks to see if the investigation is more or less truthful. | AI (Artificial Intelligence)  Databases | Will require obscene amounts of data and memory. |
| The ABC conjectures | Write a program which solves the ABC conjecture. | Python  Maths | Essentially impossible. |
| Symptom-to-diagnosis | You can input the symptoms into the program, and it outputs a list of what you could have and what to do about it. |  | Won’t be as effective as talking to an actual doctor and might diagnose people incorrectly. |
| Personalised news site | You can sign up and post full articles on something you’re interested in. All articles will have to be fact-checked and proven, but it could result in a positive spread of information and education. | Verification  User Management  UI | Has potential to go wrong as people could be only seeing articles that favour their radical, political beliefs.  Also this is every news distributor that already exists with the only difference being that the articles are fact checked. |
| Sheet Music Converter | You play a note, and it converts it into sheet music. | Microphone  Sound recognition | Websites like this already exist. |
| SportsMania – Become the ultimate athlete | A game where you and 4 other people get put into a team and play against another team of 5 a 90-second game plays out where each player must make a decision until the game can progress. – Football.  Could adapt it to other sports. |  | Making a game online let alone with more than 2 players would be something very hard and probably outside of my programming capabilities. |
| Minits/Slots app | It’s a social media app/website. The idea is that many people record/upload a photo or a video to a pool. Every 60 seconds, a random person’s submission is chosen to be displayed to every single person on the website/app. | Database  HTML/CSS/JS | Once again, online features are very hard to implement and outside of my programming capabilities. |

|  |  |  |  |
| --- | --- | --- | --- |
| Project Title | Brief Idea | Technologies involved | Disadvantages |
| Playlist Battle | You and your friend can choose one of your own playlists, or an album you like, or play from scratch. If you've chosen a playlist or an album to draft songs from, you take it in turns in drafting a song from that playlist. You organise your picks in order from worst to best, and then the rating game begins. You get your opponent’s songs in a random order, and you rate them on a scale from 1-10. Every time a new song comes up, a small part of the song will play in the background so that the player gets an idea. Once all of the songs have been rated, the final stage begins. Both drafts are reverted back into their organised order, and position by position (from worst to best) the other person’s ratings are revealed. The scoring can either be:   * Total Ratings vs Total Ratings * The winner of each battle adds one to the total of the person’s score * Average Rating vs Average Rating | Spotify API  Algorithm | I can’t think of a system of how the rosters can go head to head. |
| Musician Finder | Find musicians for your band or duet and filter by skill level and you can upload your own performances / training to verify.  LinkedIn / Fiverr for musicians | Messaging Algorithm  Recommended musicians | Has massive potential to go wrong – people might not be trustworthy. |
| Workout Structurer/Logger | You can create programs for each of your workouts and see your one-rep maxes, and your personal records. You can then create a leaderboard with your friends / global. |  | Too many of these already exist but require premium access to have the features I want to include. |

# 

# The Problem

My problem about listening to music is that it’s very hard to compare music or show your music to your friends. At the moment a popular way of expressing your musical taste is to just switch on a speaker and take it in turns putting on a song. However from my own experience, when it’s your turn to choose that song you end up trying to find a song that’s a perfect blend of every single genre you’ve ever enjoyed listening to in your life – which no one is going to pay attention to properly because they’re trying to think of what song they’re going to play next.

My finished project should be something that people can use to show off their music taste efficiently and with enthusiasm. I intend on gamifying this project as a way to make people more enthusiastic about this.

# My Solution

As I’ve just mentioned I plan on gamifying this project, and I’ve come to the conclusion I want to make a web-app where multiple users can choose a playlist from a Spotify Account[[1]](#footnote-2) and take it in turns drafting a song from the playlist. After each song has been drafted from the playlist each player will have a roster of songs. Then each player’s roster will compete with the other’s in some format[[2]](#footnote-3).

# Third-Party Influences

For my research I decided to get multiple users in the form of a series of questionnaires. I plan on receiving feedback from this same group of people throughout the whole project. I will also try to find a single user who will be more involved in my project than just a series of questionnaires, and I will try to get personalised responses from this person.

[This](https://forms.gle/cZzRvXfGKtcbhZ47A) is the questionnaire I sent out. I will refer to this questionnaire later in my research as I use it to write my objectives.

# Key Terminology (explained within the context of my project):

* **Drafting**: When each player takes it in turns to take a song from a playlist and add it to their ‘roster’.
* **Roster**: A collection of songs that a player has drafted within the game.

# Research

## Researching how to implement my idea

### Choosing what music platform

When I started my research I was planning on researching extracting information from different music platforms such as Apple Music, Amazon Music, Spotify etc using their respective APIs, however I felt this would significantly overcomplicate my project. Therefore I decided to only learn how to use the Spotify API for my project for the following reasons:

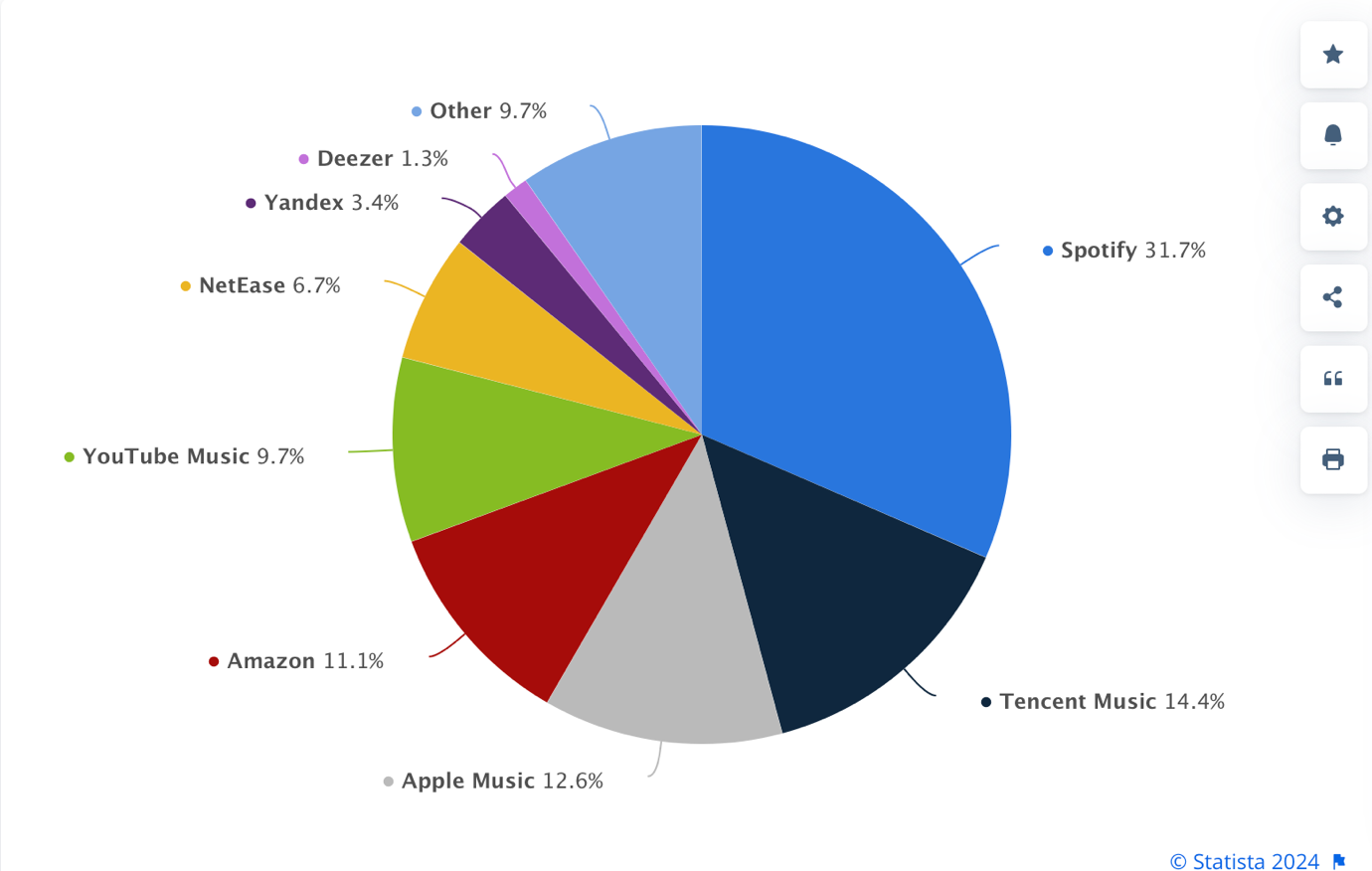
* I already have my own Spotify Premium account meaning I have access to the API and all of the relevant information surrounding it.
* I’ve only ever used Spotify so having to try and understand the features of a new musical software would make my project much harder.
* [](https://www.statista.com/statistics/653926/music-streaming-service-subscriber-share/#:~:text=That%20said%2C%20Spotify%20still%20held,suggest%20that%20this%20will%20continue.)According to Statista, in 2024 Spotify is the most popular music streaming service by far meaning that this project will be able to be used by the majority of music listeners if this was ever to be published.

Figure 1: Click on the image and it sends you to the website

### Structure of my Web-App

Midway through my research into the Spotify API I realised that I didn’t fully know what I wanted to achieve from that research. I realised that I needed to understand what the structure of my web-app would be before continuing my research. I created the following Objectives which I would need to complete before continuing my research on the Spotify API:

1. [Create a Questionnaire to decide certain things for me.](#_Questionnaire)
2. [Know the step-by-step process that would happen from the point anyone loads into the website to the ending of the game.](#_Step-by-step_process)
3. [Create a Questionnaire to decide certain things for me.](#_Questionnaire_1)
   * Decide what kind of drafting mode I would use to draft songs from playlists to build a player’s roster.
   * Decide how player’s rosters would go head-to-head against each other.
4. [Know the step-by-step process that would happen from the point anyone loads into the website to the ending of the game.](#_Step-by-step_process)
5. [Implement a simple version of this into Python.](#_Implementation_into_Python)
6. Write out a series of detailed objectives which I would want my project to achieve.

### Step-by-step process

When having to create something from scratch, I always use paper to draw out ideas as I can visualise it much easier – as I can link and draw arrows from one step to another. Here is a scan of the original step-by-step process I drew.

A close-up of a paper

Description automatically generated

### Questionnaire

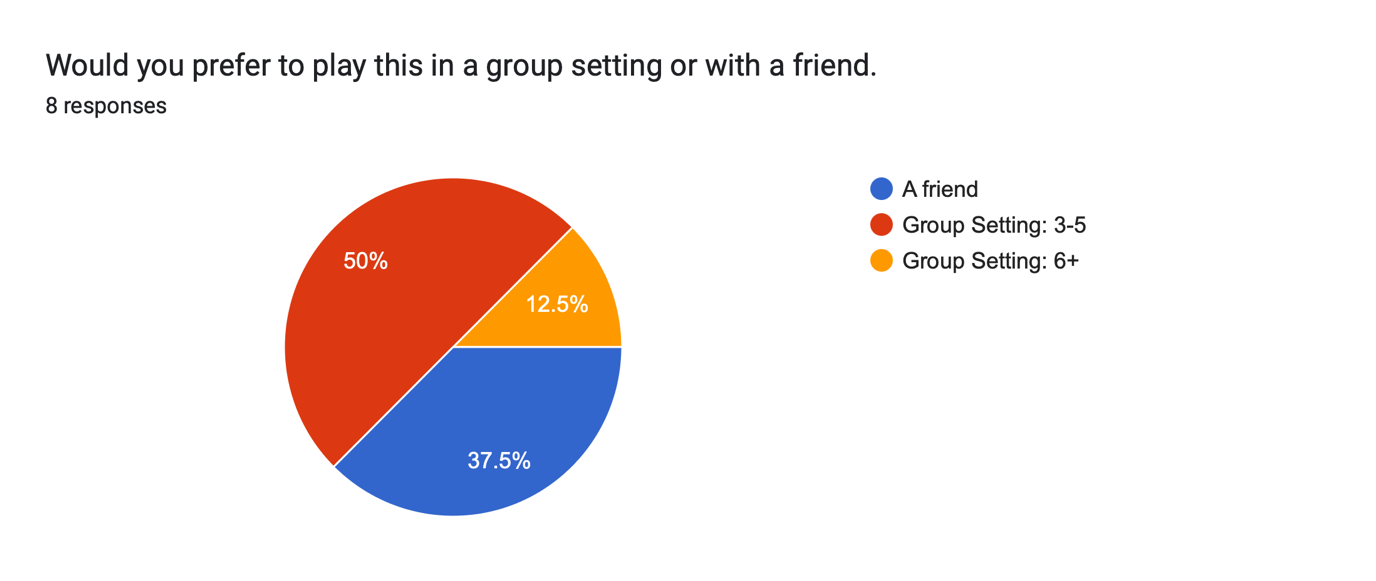
For each question I’ll provide a justification for why I asked that question.

To make sure that everyone that received my form knew the full context of my final product, I included the following description – which is essentially just a writeup of the image in the [step-by-step](#_Step-by-step_process) section.

A white and purple box with black text

Description automatically generated

**Question 1. Would you prefer to play this in a group setting or with a friend.**



I wanted to know the range of group numbers of people. In terms of complexity, it would be easier to implement a 2-player game, however I personally believe the game would be more enjoyable with more people – as 50% of responders also decided.

**Question 2: Do you think this app should have features that improve the accessibility for the visually impaired?**

**A blue and red pie chart

Description automatically generated**

I was thinking of implementing a system which helped people with colour blindness – and the results showed me that it would be significantly useful and necessary.

**Question 3: Would you rather draft from a personalised playlist or an album?**

A blue and orange pie chart

Description automatically generated

Funnily enough, nobody selected the ‘Album’ choice and to make matters more complicated: I had a 50-50. For the sake of simplicity – and because everyone is fine with choosing from a playlist – I’m going to go with just a playlist.



**Question 4: How should the order for the drafting process be decided?**



A pie chart with numbers and a few dots

Description automatically generated with medium confidence

I was stuck on what order people should be drafted in. I was stuck between the 3 options so I decided to get my Form-Responders to decide for me. To clarify this is the process that happens before the actual draft – as the drafting needs an order of people and this process will decide that order.

**Question 5: Once the original order is decided, what should the drafting process be like?**

A graph with purple bars

Description automatically generated

This is now deciding the actual drafting process. My 3 choices were:

1. Let’s say there’s 4 players drafting and the order of the draft is Player 1, Player 2, Player 3, Player 4.

Players 1-3 each take a turn choosing a song, but then Player 4 gets to choose 2 as they have less to choose from. Then the list goes back in reverse order until Player 1 – who then takes 2 again if there are still songs to choose.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Turn | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Player | 1 | 2 | 3 | 4 | 4 | 3 | 2 | 1 |

This system is based off of Catan.

1. Each person takes it in turns to choose one player.

Player 1

Player 2

Player 8

Player 7

Player 3

Player 4

Player 6

Player 5

This would loop round until the

### Project Objectives:

1. Throughout the whole game the user must be able to open settings.
   1. This would probably be done in the form of an irremovable button in the top right of the screen.
      1. When clicked on, this should open a drop-down menu of at least 3 different options: Profile, Game Log, Other.
2. When the user/s are not in a game (i.e set-up, navigating through settings etc…) there should be an option to go “Back”.
   1. This should be done in the form of an irremovable button in the top left of the screen.
      1. When pressed, it should take the user back to the previous page.
   2. The button should not exist in the middle of the game.
3. The opening page should have a “Start New Game” Button.
   1. This button must be the focal point of the opening page.
   2. Pressing this button should then go to a new page with 2 different options.
      1. Choosing a playlist.
         1. If they choose this option they have to connect their Spotify account and choose a playlist from their library.
         2. There needs to be a validation process making sure that there are at least 10 songs in the playlist.
      2. Playing as a guest.
         1. The user doesn’t need to connect their Spotify account and will be able to choose from a Preset Playlist.
4. There must be a maximum of \_\_\_\_ players.
   1. There must be an option to choose how many players are playing.
      1. This would be done either in the form of:
         1. A text box
            1. If this option is chosen, there should be a validation process checking that the user has actually entered an integer that isn’t outside the range of how many players allowed.
         2. A drop-down menu.
   2. Each player must be able to be told apart in-game.
      1. For each player, a page should appear which allows them to do the following:
         1. Enter a player name.
            1. If a name isn’t entered / is empty, there should be a default name that is given to that user (Such as “Player 1” or “Player” etc.)
            2. 2 players can’t have the same name.

If this happens one of the following should be implemented:

A validation process that tells the player that somebody else has chosen that name.

The second player to choose that name will have it saved as the following: ‘{Name}1’. The number should increase by 1 for every usage.

* + - 1. Associate their player with a colour and a shape (for people dealing with colour-blindness).
    1. At the bottom of this page, there should be a Confirm Button.
       1. When this button is pressed, the user can’t go back.
          1. When Player 1 confirms, the “Back” button mentioned in Objective 2. Will disappear.
  1. After all of the players have ‘confirmed’ it should go to a page with a detailed list of all the users.
     1. This should be implemented in the format: ‘Chosen Shape’ with a colour fill of the ‘Colour Selected’, with their ‘Chosen Name’ to the right of that, and an ‘Edit’ Button on the outside.
        1. When the ‘Edit’ button is pressed, a dropdown menu should appear with the following options:
           1. Change the player’s name

The objective 4.2.1.1.2 should still be implemented.

* + - * 1. Change the player’s colour and shape.

The colours and shapes that other people have chosen shouldn’t be an option to choose from.

1. The drafting process should take no longer than 2 times the number of players in minutes (3 players should take 6 minutes, 4 players 8 minutes etc…).
2. Each player should have the same number of songs in their roster.
   1. There must be a validation process to make sure that this happens.

## Researching how to implement my code

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

### Learning the Spotify API in JS

When first beginning to research the Spotify API I used [this website](https://developer.spotify.com/documentation/web-api) to help me begin learning how to use it. It was very confusing for me as it was all done in the terminal and using Javascript – a language I had very little experience with. It was also the first time I had ever used an API so I wasn’t familiar with how they worked. My plan was to follow the tutorial on Javascript until I had a vague understanding of what to expect from this API and then to find a similar tutorial but using Python.

The first thing I managed to do using this tutorial was get an access token and use that to extract information about any artist on Spotify I wanted. This process begun with me creating an app on the dashboard *(which is something I will have to do when I switch to Python anyway)* which gave me a personal Client Secret and Client ID – which I would have to use to get this access token. I noticed that the token only lasted for **3600 seconds (1 hour)** which I’ll have to take into consideration later. The image below shows an implementation of me receiving information about my chosen artist (Kendrick Lamar).

A black screen with white text

Description automatically generated

### Creating a UI

Once I completed this I wanted to improve my understanding of what this API could do – and more importantly whether I could access Personal Profiles and the playlists on those profiles. I also wanted to start building an extremely simple and basic UI to display this data. I started to follow [this tutorial.](https://developer.spotify.com/documentation/web-api/howtos/web-app-profile)

It required npm, vite and node.js – none of which I’d ever heard of let alone knew how to install. When attempting to do so I realised there were issues with how my laptop was setup (I hadn’t updated my terminal and not had the right programs installed to allow for the installation of the programs I needed to install.) I also decided to work across 2 different terminals – my MacBook terminal and the VSCode terminal. Theoretically they were supposed to be the same and have the same programs installed but every time I wanted to try anything in the VSCode terminal it wouldn’t work the same as in the MacBook terminal – (and I was extremely set on making the VSCode terminal work because that would most likely be the software I would use for when I switched to Python)[[3]](#footnote-4). I realised that my VSCode terminal wasn’t zsh but my MacBook one was – so that was a much easier fix than I originally thought. Once I finally installed all 3 programs I followed the tutorial, and as I wasn’t familiar with Javascript that well I attempted to analyse everything I was taking from the website and understanding the purpose of each line of code. After everything, my Javascript program looked like this:

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

Which, alongside the default HTML provided on the website looked like this on a localhost server:

A person in a military uniform

Description automatically generated

### Implementation into Python

It was recommended to me to do the majority of my code in Python, as it’s the programming language I’m most familiar with. When I was using the Javascript tutorial I tried to understand what each piece of code did before I used it in my programs, however I wouldn’t be able to implement anything from memory because I’ve not used much Javascript before. Learning an entire programming language and then completing a complex program with it within the time frame I have isn’t ideal at all, especially when I could do it in a language I understand.

Spotify themselves don’t provide a Python Tutorial on how to use their API, however there’s a publicly available Tutorial called ‘Spotipy’ (linked [here](https://spotipy.readthedocs.io/en/2.24.0/))

1. *I will explain why I’ve chosen Spotify specifically in my RESEARCH.* [↑](#footnote-ref-2)
2. *This format will be influenced by the questionnaire I sent out.* [↑](#footnote-ref-3)
3. *In hindsight I could’ve just done everything on the MacBook terminal and then sorted out the VSCode later but I didn’t lose too much time on this.* [↑](#footnote-ref-4)