1. **Project Title:** Trackbase
2. **Project Summary:**
   a. A song-wiki that draws data from a CSV to list song details including release year, genre, popularity, artist, and album name. This data will also include values like danceability, energy, and beat per minute. A key feature will be a quiz with questions that will guide a query towards songs the user would be interested in listening to. After seeing the search results the user will be able to click on any artists or songs recommended to them and see further information as stated above.
3. **Description of an application of your choice. State as clearly as possible what you want to do. What problem do you want to solve, etc.?**
   a. We want to provide people with information about the music, and help find music that they will like. Our project will be a one stop shop for information and statistics about music and songs, and will provide details about the song, such as numbers of listens on certain platforms, year released, genre, album and/or compilations released in, artist of the song, producers of the song, and movies and tv shows the song was featured in. Also details about the artist, such as years active, a list of songs, a list of albums/compilations, which both can be sorted by release or by popularity, and more. Also information about which songs and artists are trending the most, which songs and artists have the most total listens. We also would like to implement a system that would allow someone to take a quiz to learn about new recommendations for music that is similar to what the quiz thinks they would like. We also would like to implement a favorite system where users can go in and favorite a song, which would also give more and better suited recommendations.
   b. When users visit the home page, they will see a curated list of trending or popular songs, and have access to a search bar for finding specific songs, artists, or playlists. Users will be able to create a list of favorite songs from the recommendations or search results and can also create custom playlists by selecting songs from these recommendations or their favorites. They can update their content by adding or removing songs from their favorites list and modifying their custom playlists. Additionally, users can delete songs from their favorites or remove entire custom playlists.

4. **What would be a good creative component (technically challenging function) that can improve the functionality of your application? (What is something cool that you want to include? How are you planning to achieve it?)**
   a. A creative component could be a quiz where we would ask basic questions of the user’s music preferences and then suggest other music albums or songs that they would like to listen to. Some questions could be “Who is your favorite artist?” and then we would take a query of the different attributes and find another artist

that is most alike. We can also ask the user to rate certain aspects of music that they enjoy from a scale 1 to 10 and try to find the closest artists or songs that match. We can do this by grouping the songs by artist and then finding the average of their attributes such as danceability, speechiness, beat per minute, etc and recommend those artists. An additional feature we could include is playing a 30 second introduction of the recommended song and allowing the user to like or dislike before recommending another song. We could take this one step further, and have the user create a playlist that we could then use to make recommendations from. Another possibility to include would be a visual graph of how unique a person's taste could be. We could gather a dataset of different user's favorite playlist and count the number of viable songs we could recommend based on their preferences and then calculate how unique they are with a pie chart. Moreover, we could include a randomization feature where the users can randomly pick a song. We were also considering having a light and dark mode for users to switch between as some of us have weak eyes.

5. **Usefulness. Explain as clearly as possible why your chosen application is useful. What are the basic functions of your web application? (What can users of this website do? Which simple and complex features are there?). Make sure to answer the following questions: Are there any similar websites/applications out there? If so, what are they, and how is yours different?**

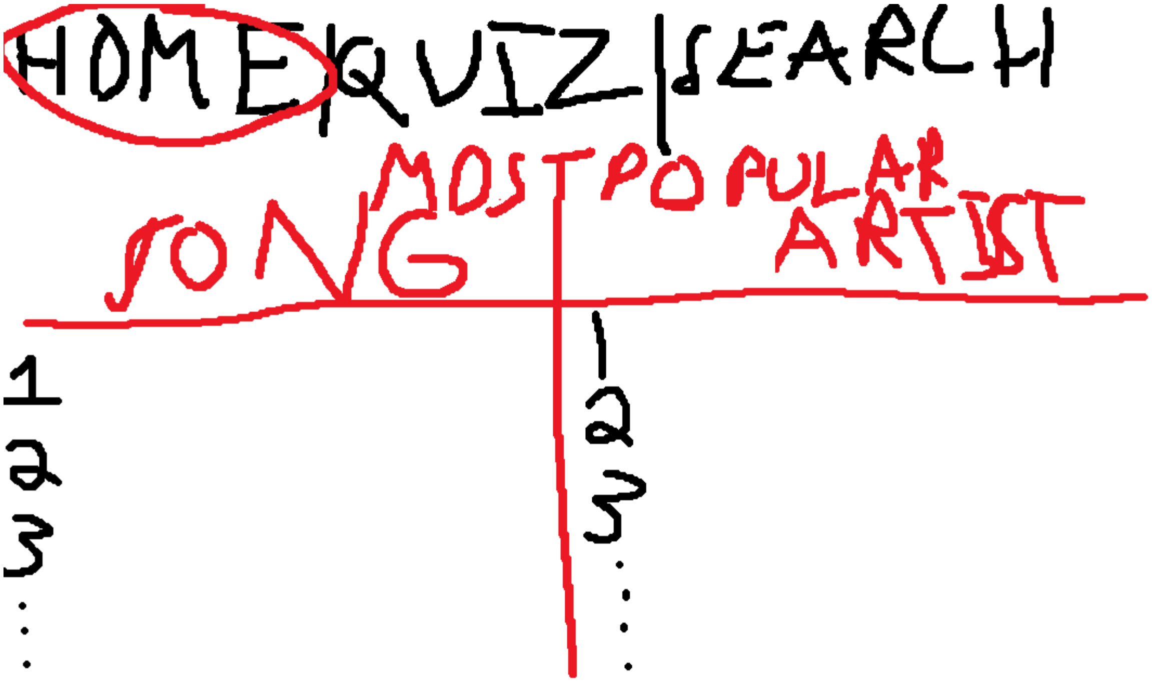
    a. This application is useful to the user because not only will it be able to recommend songs, but you will be able to search and parse through a bunch of different songs with different statistics. They will range from danceability, energy, and popularity to year-released, artist name, and album name. There will be a search filter where you can select different statistics to find a certain type of song. All of these features help the user find a bunch of different data on songs. A similar website could be something like Spotify, which has a bunch of details about a lot of different songs. They also let you search up and listen to these songs. Our website will be different however because the focus isn't to listen to the songs it is to recommend and give you information about them. After answering a short quiz it will feed you a query of songs that we think you would enjoy. Then, you will be able to click and see all the information about them. A user would come to this website to find more information about songs as well as discovering new music that they would be interested in.

6. **Realness. We want you to build a real application. So, make sure to locate real datasets. Describe your data sources (Where is the data from? In what format [csv, xls, txt,...], data size [cardinality and degree], what information does the data source capture?). It would be hard to satisfy stage 2 requirements with one dataset. Thus, we strongly recommend identifying at least two different data sources for your project.**

a. We have one data source from Kaggle (https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset/data). The CSV file contains 21 columns with 114,000 rows of data. The data was collected on October 20th, 2022, and published on October 22nd, 2022. The columns on the dataset provide us the following information: track ID from Spotify, artist(s) of the song, album name the song is a part of, the popularity of the song, the duration/length of the song, if the song is explicit or not, the danceability of the song, the energy(how intense/fast) the song is, the key of the song, how loud the song is in decibels, the modality of the song, the speechiness(how much of the song includes words), how acoustic the song is, whether or not the song is instrumental(has singing or not), if the track was performed live, how happy or sad the song is, the tempo/BPM of the song, how many beats per measure, and what genre the song is a part of.

   Another data source we have is also from Kaggle (https://www.kaggle.com/datasets/amitanshjoshi/spotify-1million-tracks). The CSV file contains 20 columns with almost 1.5 million rows of data. The data collected songs in the time period of 2000-2023. It also has similar information to the previous dataset mentioned, with the difference being that this dataset with over a million data doesn’t include whether or not a song is explicit.

7. **A detailed description of the functionality that your website offers. This is where you talk about what the website delivers. Talk about how a user would interact with the application (i.e., things that one could create, delete, update, or search for). Read the requirements for stage 4 to see what other functionalities you want to provide to the users. You should include:**
   a. A low-fidelity UI mockup: What do you imagine your final application’s interface might look like? A PowerPoint slide or a pencil sketch on a piece of paper works!


HOME
QUIZ
SEARCH
MOST POPULAR
SONG
ARTIST
1
2
3
...
1
2
3
...

HOME | [✓ QUIZ] | SEARCH

1. HOW FAST DO YOU LIKE YOUR MUSIC?

a. b. c. e.

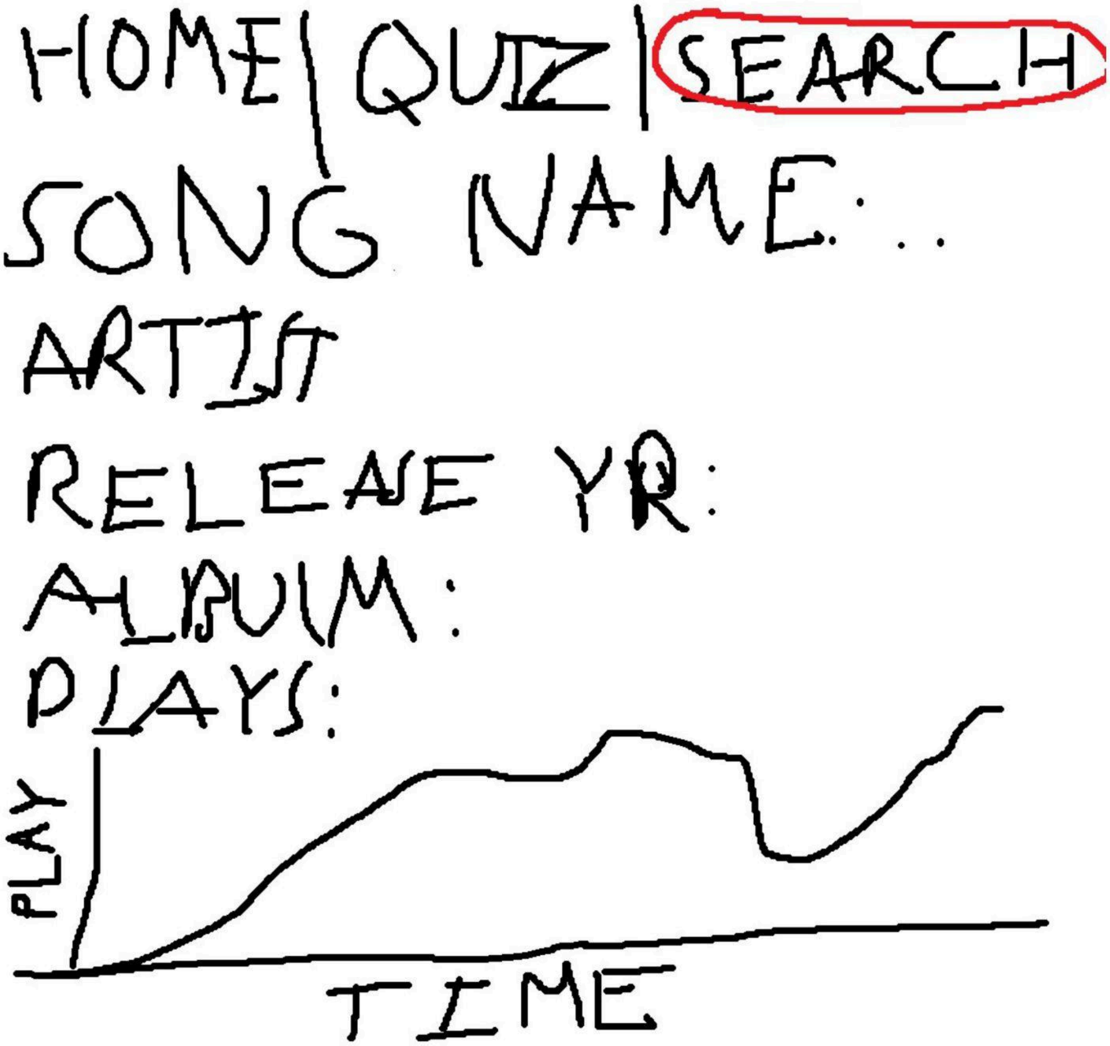
2 . . . .

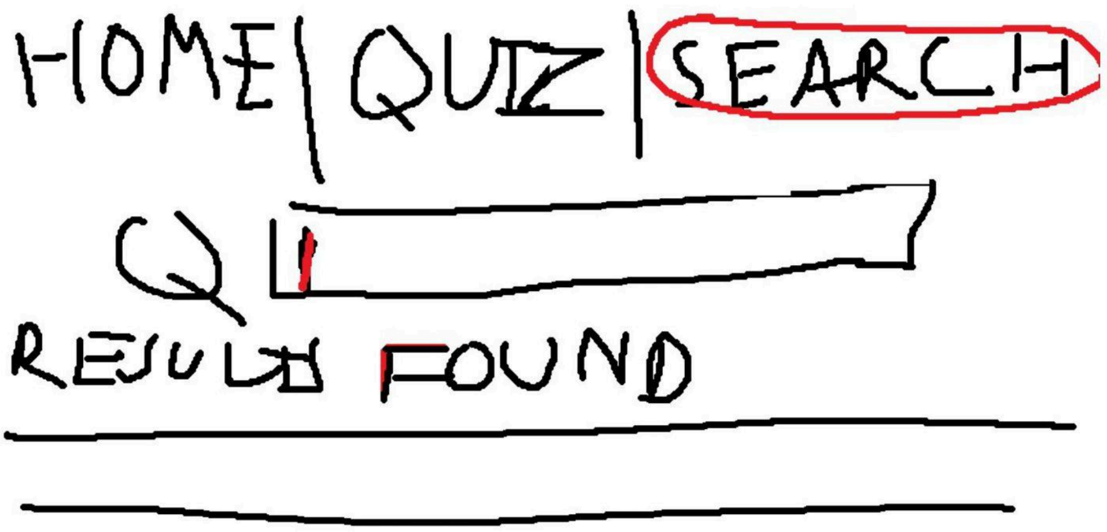
3 . . . - .

YOU MIGHT LIKE:

DRAKE

\


HOME | QUIZ | SEARCH
SONG NAME: ..
ARTIST
RELEASE YR:
ALBUM:
PLAYS:
PLAY
TIME



b. Project work distribution: Who will be responsible for each of the tasks or subtasks? Explain how backend systems will be distributed across members. Be as specific as possible as this could be part of the final peer evaluation metrics.

   i. Ben

      1. Will work on the search engine, setting up how a query can be made from what the user types to retrieve proper song data.

   ii. Jack

      1. Will work on developing the quiz feature on the website, making questions and using the answers to form a query that can be used with the database to output results on the website

   iii. Phillip

      1. Will work on setting up the website and connecting it to the database, and displaying statistics for a certain song/album

   iv. Sabir

      1. Will work on converting the data from the CSV files into a SQL Database and help set up for query use for website