Mitchell Brown
Yuhang Chen

Super Senior Music Database

Project URL

http://flip2.engr.oregonstate.edu:8082/

Executive Summary

Throughout the implementation of our database, we went through many iterations based on peer and TA feedback. One of the first updates we made after our initial proposal was changing how our M:M relationships were represented in the database. We originally had them embedded within tables, but after feedback, learned we needed to set up relationship tables separate from our four primary tables. In the next week, we updated a lot of details on our schema and ERD, improving how foreign keys and relationships were represented.

The next set of updates came in the implementation of our website and database backend. Our initial idea of having users search for songs was changed to showing the entirety of the tables on each page. We also added buttons to UPDATE and DELETE on each row to make this as easy as possible for the user, as well as drop-down menus when doing INSERTs on relationship tables. We also updated which entities could be entered as NULL to make INSERT, UPDATE, and DELETE possible.

During the final implementation, we made a few more additions and changes. Those included adding a SEARCH option to the songs page, as well as the ability to update the relationship between Albums and Artists.

Mitchell Brown
Yuhang Chen

Overview

Within the Super Senior Music website, it is important for the user to have a lot of control over the music they listen to, and for the website to be able to recommend songs the user will enjoy. With over 5 million songs to stream, it would be difficult to find music without relating the songs to each other. Introducing a database would allow key metrics to be tracked to maximize user experience.

Tracking artist names, albums, genres, in addition to songs would allow this control. Artists write many songs, sometimes even over 100 songs they release to be streamed. Albums often have 10 to 20 songs within them, and there are an endless number of genres with an endless number of songs within each. Each of these metrics relates songs to one another; albums generally contain songs of similar sound and style, artists often write much of their music with a style in mind, generally under a single genre or related genres. Genres allow for similar songs to be found together even if they are under separate albums and written by different artists.

Tracking these metrics will allow the user to find music of similar style to something they like, such as finding all modern rock songs by an array of artists. As well as this, by understanding what makes certain songs similar, the website can find new music for the user which it will then recommend to them. An example of this would be if the user enjoys listening to music by Jack Johnson, the website may recommend John Mayer for them.

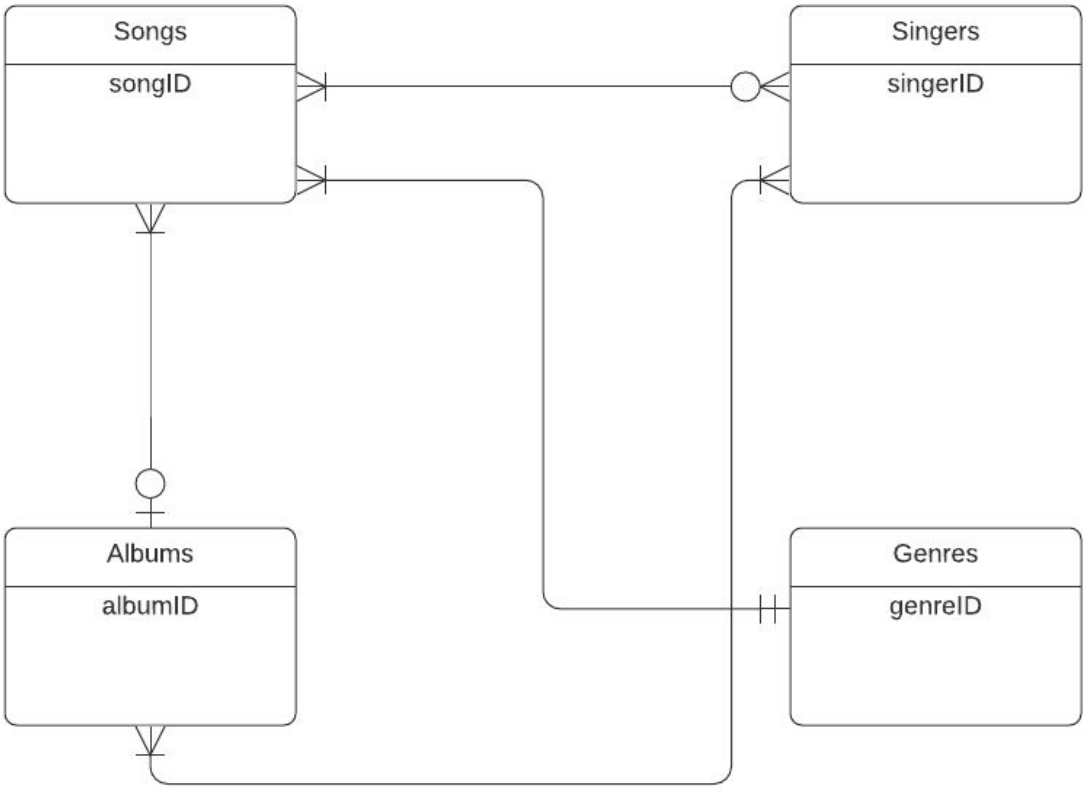
Database Outline

1. Entity:
    a. Songs: records the details of songs.
        i. songID: INT, auto_increment,unique,not null,PK
        ii. songName: VARCHAR, not null
        iii. album: one albumID, FK
        iv. genre: one genreID, FK
    b. Singers: records the details of singers who write and sing songs
        i. singerID: INT, auto_increment,unique,not null,PK
        ii. singerName: VARCHAR, not null
        iii. homeCity: VARCHAR
        iv. homeState: VARCHAR
    c. Albums: records the details of albums that contain information about songs and singers
        i. albumID: INT, auto_increment,unique,not null,PK
        ii. albumName: VARCHAR, not null
        iii. albumReleaseDate: Date

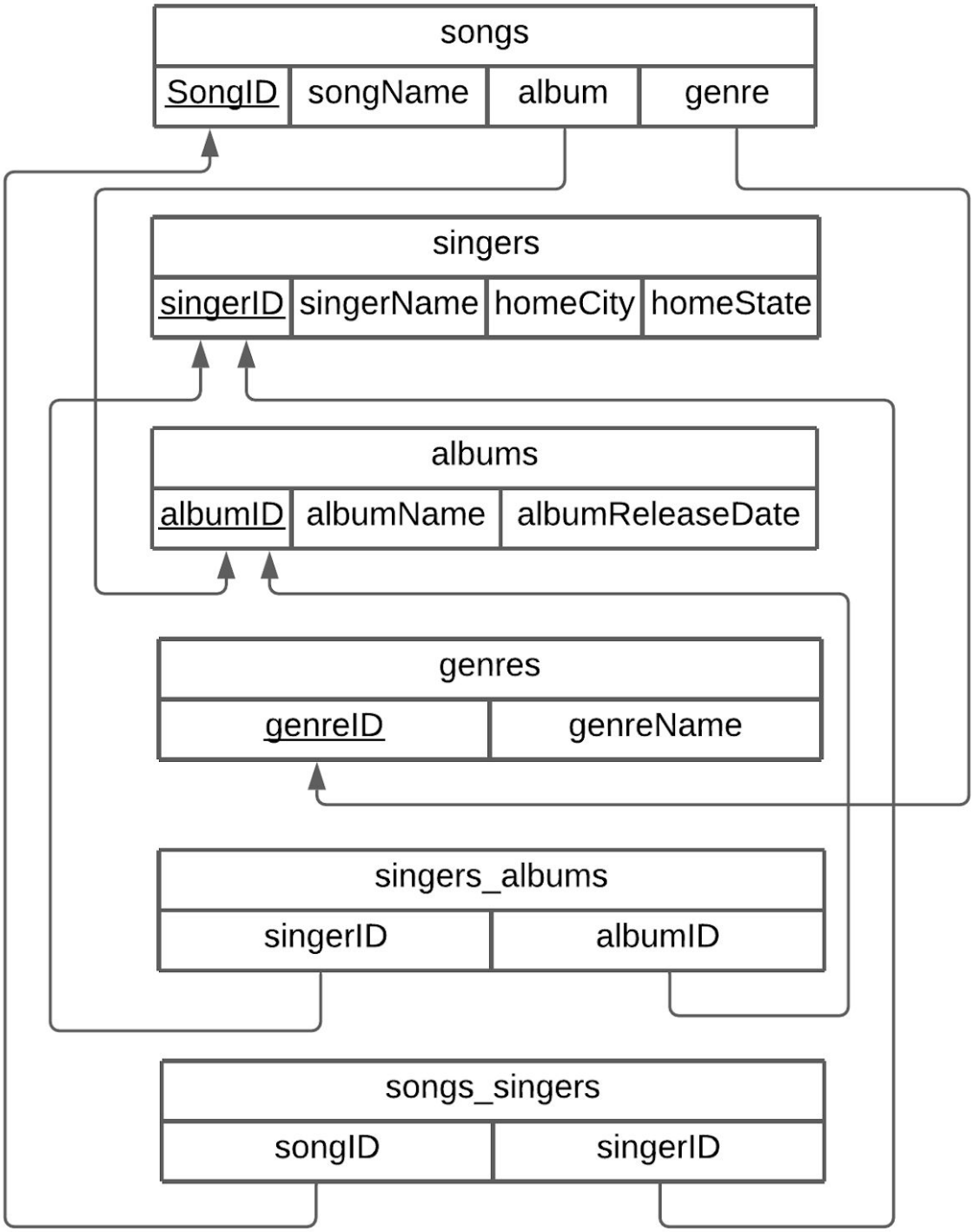
   d. Genres: collections of songs that are in a similar style
      i. genreID: INT, auto_increment,unique,not null,PK
      ii. genreName: VARCHAR, not null
2. Intersection table:
   a. Songs_Singers(This is the intersection table for M: M relationship between Songs and Singers)
      i. songID: FK
      ii. singerID: FK
   b. Singers_Albums: (This is the intersection table for M: M relationship between Singers and Albums)
      i. singerID: FK
      ii. albumID: FK
3. Description for relationships:
   a. Songs to singers: many to many
      i. One singer will have many songs under their name, and multiple artists may collaborate on the same song. (A singer need at least one song, a song need at least one singer)
   b. Songs to albums: one to many
      i. A song fits into one and only one album, but an album will have many songs on it (At least one song).
   c. Songs to genres: one to many
      i. In most cases, songs will fit into one and only one genre, but a genre will contain many different songs.
   d. Singers to albums: many to many
      i. A single singer could have multiple albums under their name, and multiple artists could collaborate on an album. (A singer needs at least one album and an album need at least one singer.)
4. Special Rules:
   a. We will assume that one song could be made by multiple, collaborative artists.
   b. We assume that a song will be unique to an album, meaning the song cannot appear on multiple albums
   c. We will make the simplification that a song will fit into a single genre, rather than fitting into multiple.
   d. We assume that an album may have multiple, collaborative artists.
   e. We assume that a song must have at least one singer, and it must be in exactly one album and one genre

Mitchell Brown
Yuhang Chen

Entity-Relationship Diagram



Schema


songs
SongID
songName
album
genre
singers
singerID
singerName
homeCity
homeState
albums
albumID
albumName
albumReleaseDate
genres
genreID
genreName
singers_albums
singerID
albumID
songs_singers
songID
singerID

Mitchell Brown
Yuhang Chen

CS 340 TEAM EVALUATION form

March 8, 2021

RATE YOUR TEAMS PERFORMANCE USING THE SCALE BELOW.

1 = Strongly Disagree 2 = Disagree 3 = Agree 4 = Strongly Agree

GROUP NUMBER	30	
NAME OF GROUP TEAM MEMBERS:	Yuhang Chen, Mitchell Brown	
SCALE AND COMMENTS	RATING	ADDITIONAL COMMENTS
HOW PREPARED WAS YOUR TEAM? Research, reading, and assignment complete	4	Well prepared, we started everything early, and communicate how we would complete tasks

HOW RESPONSIVE & COMMUNICATIVE WERE YOU BOTH AS A TEAM? Responded to requests and assignment modifications needed. Initiated and responded appropriately via email, Slack etc.	4	We had a meeting once a week to split work and we both did a good job
DID BOTH GROUP MEMBERS PARTICIPATE EQUALLY Contributed best academic ability	4	We split the job evenly and both did enough work for homework
DID YOU BOTH FOLLOW THE INITIAL TEAM CONTRACT? Were both team members both positive and productive?	4	Yes, we did

Are there any suggestions for improvement for your team and what are your goals moving forward?

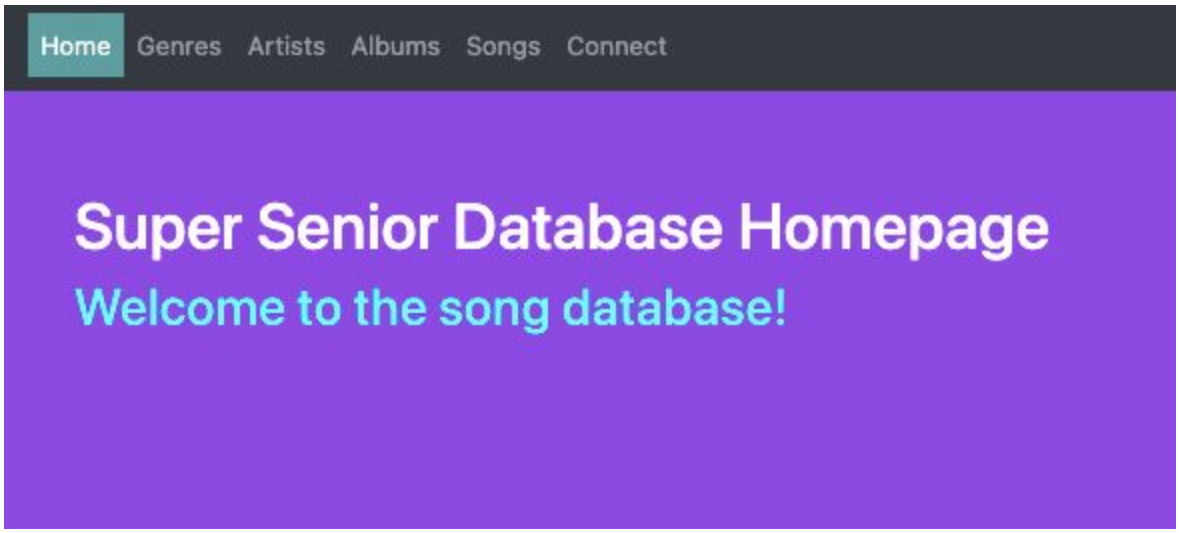
(Better communication, follow the contract better, modify the initial team contract, more contribution, etc?)?

Yuhang Chen: N/A

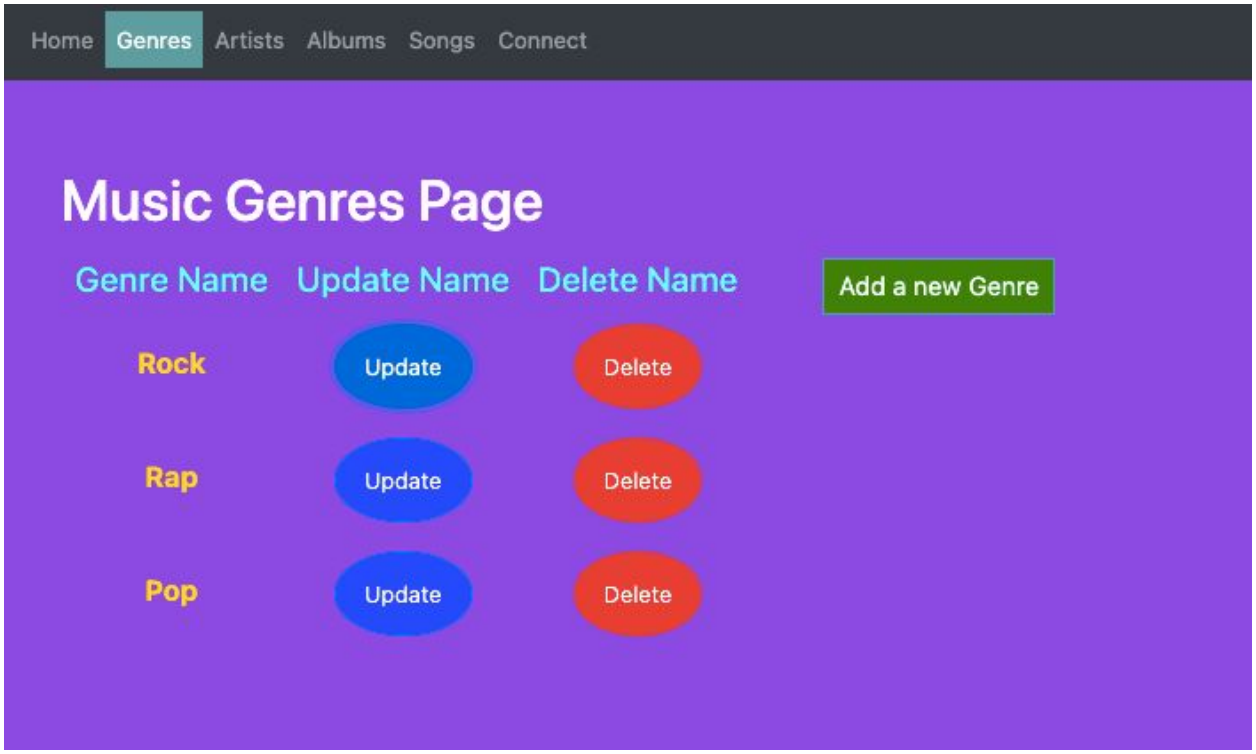
Mitchell Brown: I think we did quite well together.

User Interface Screenshots

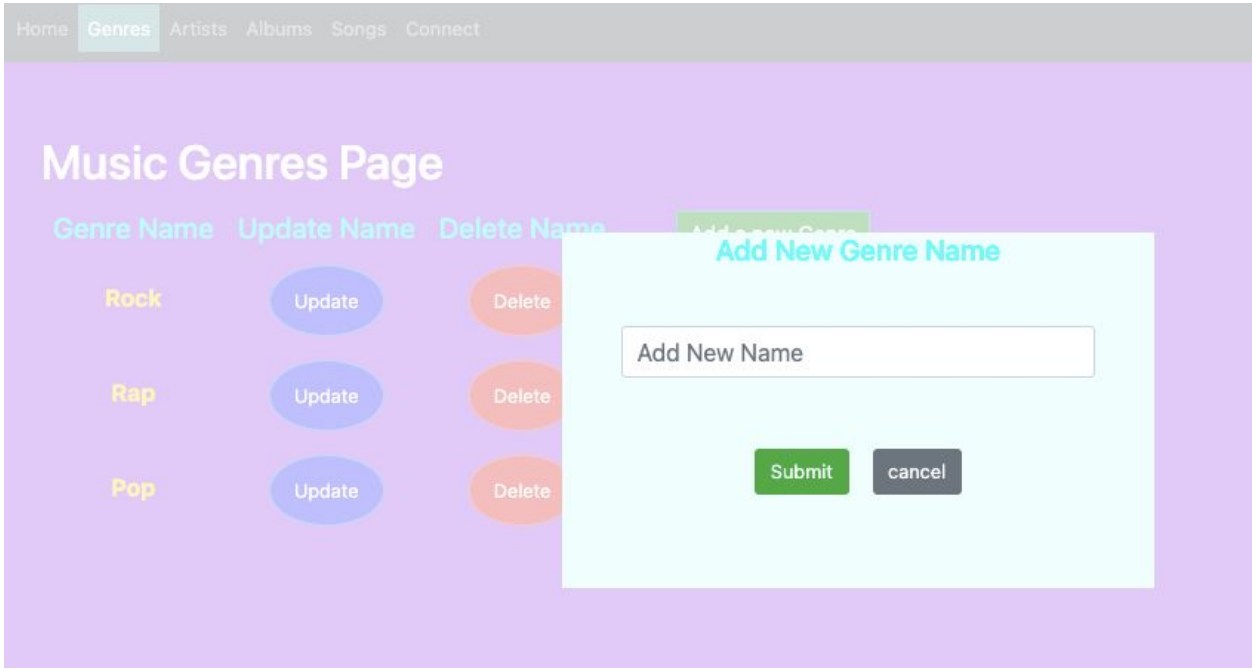
1. Landing Page



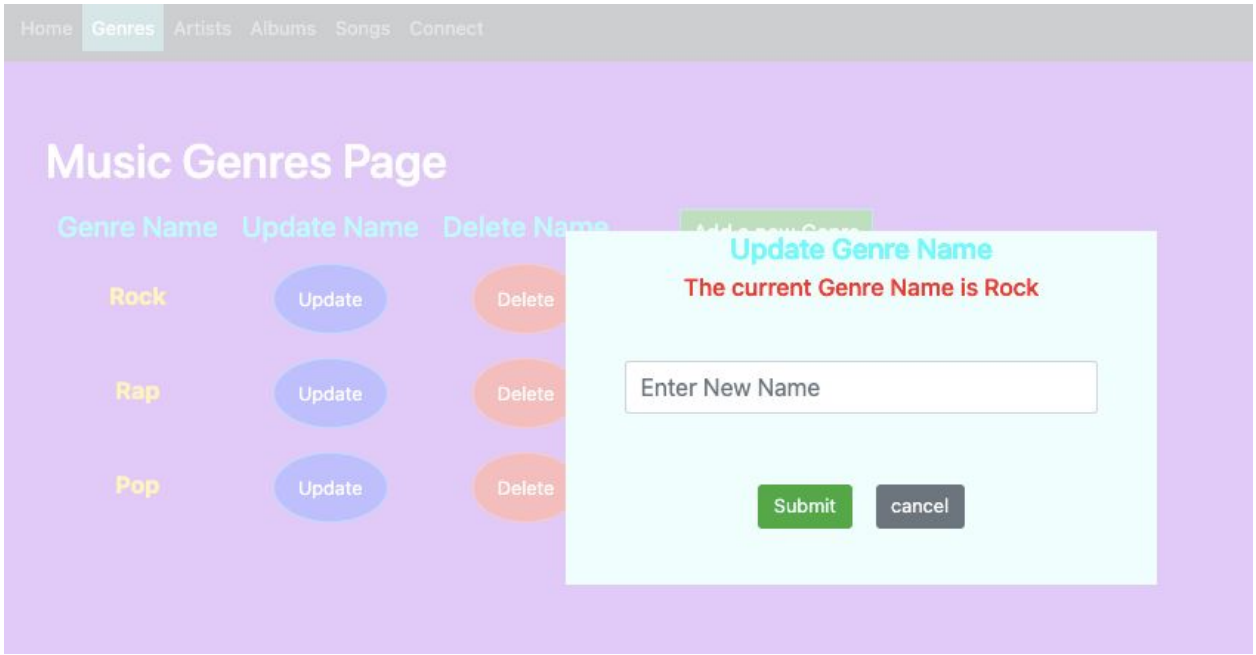
2. READ genres page



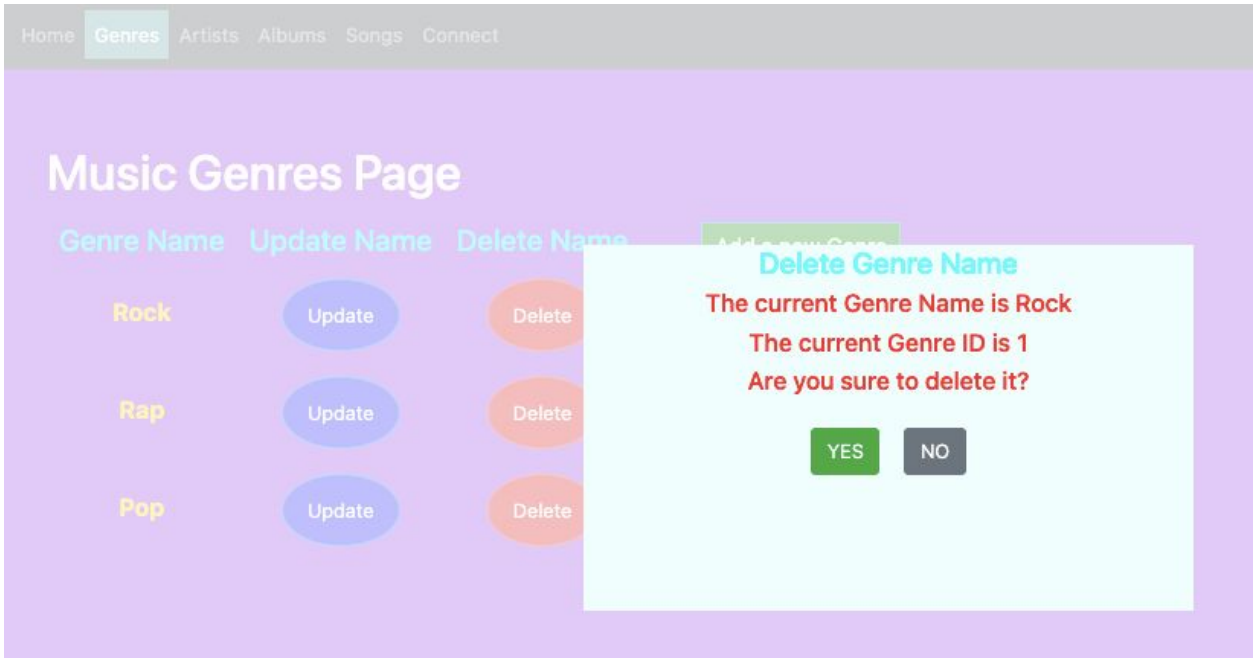
3. ADD genres page



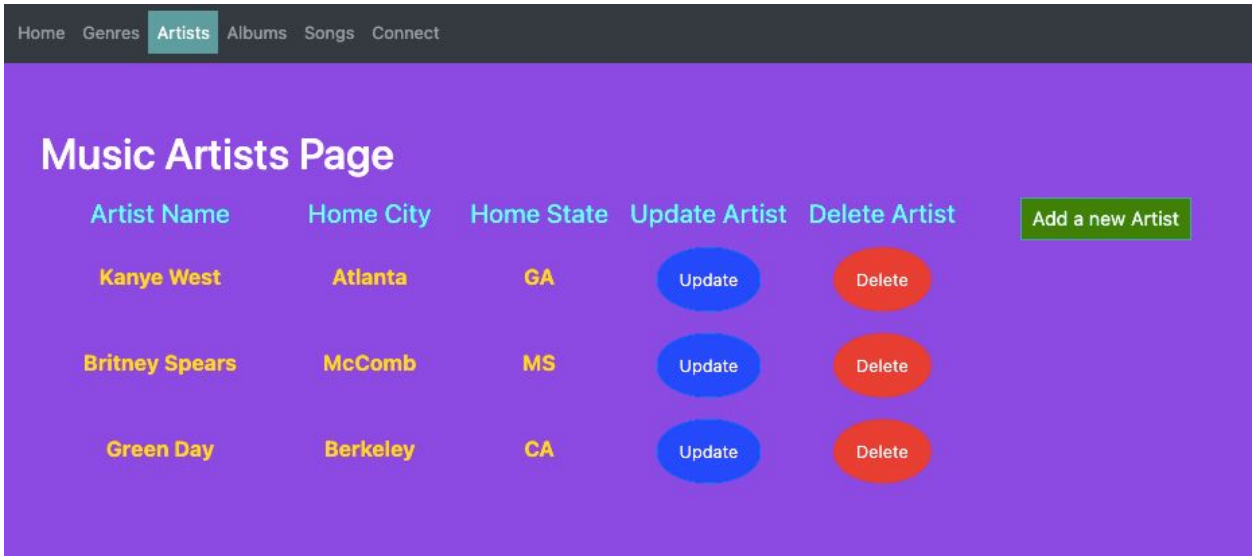
4. UPDATE genres page



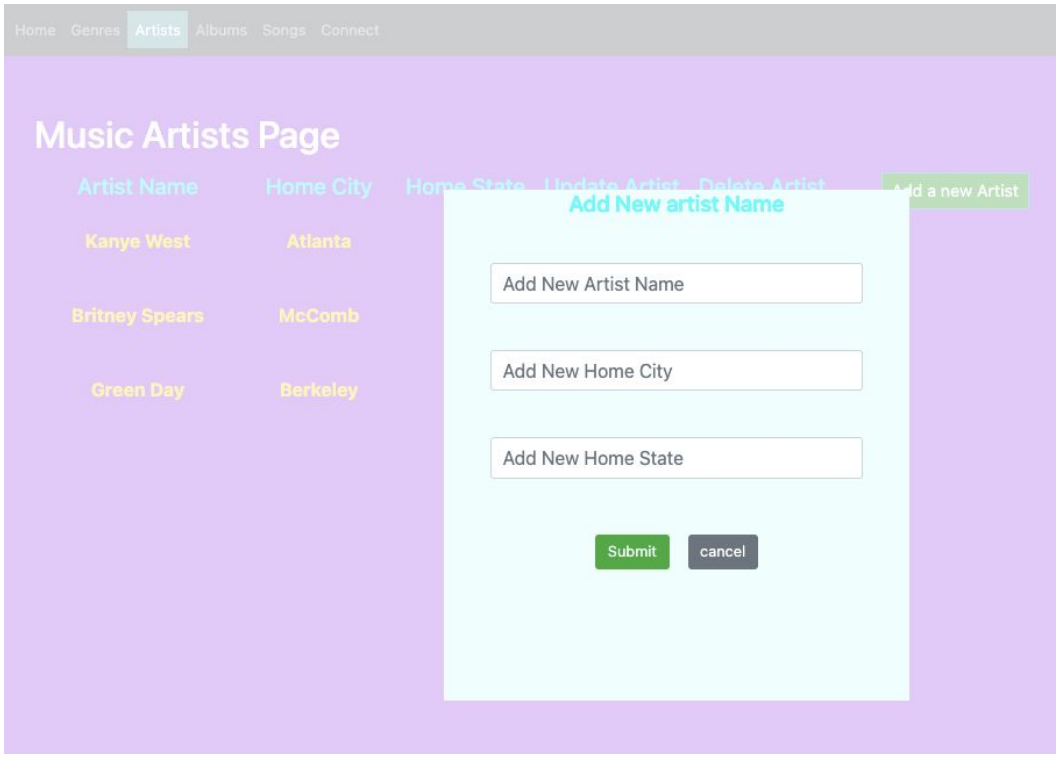
5. DELETE genres page



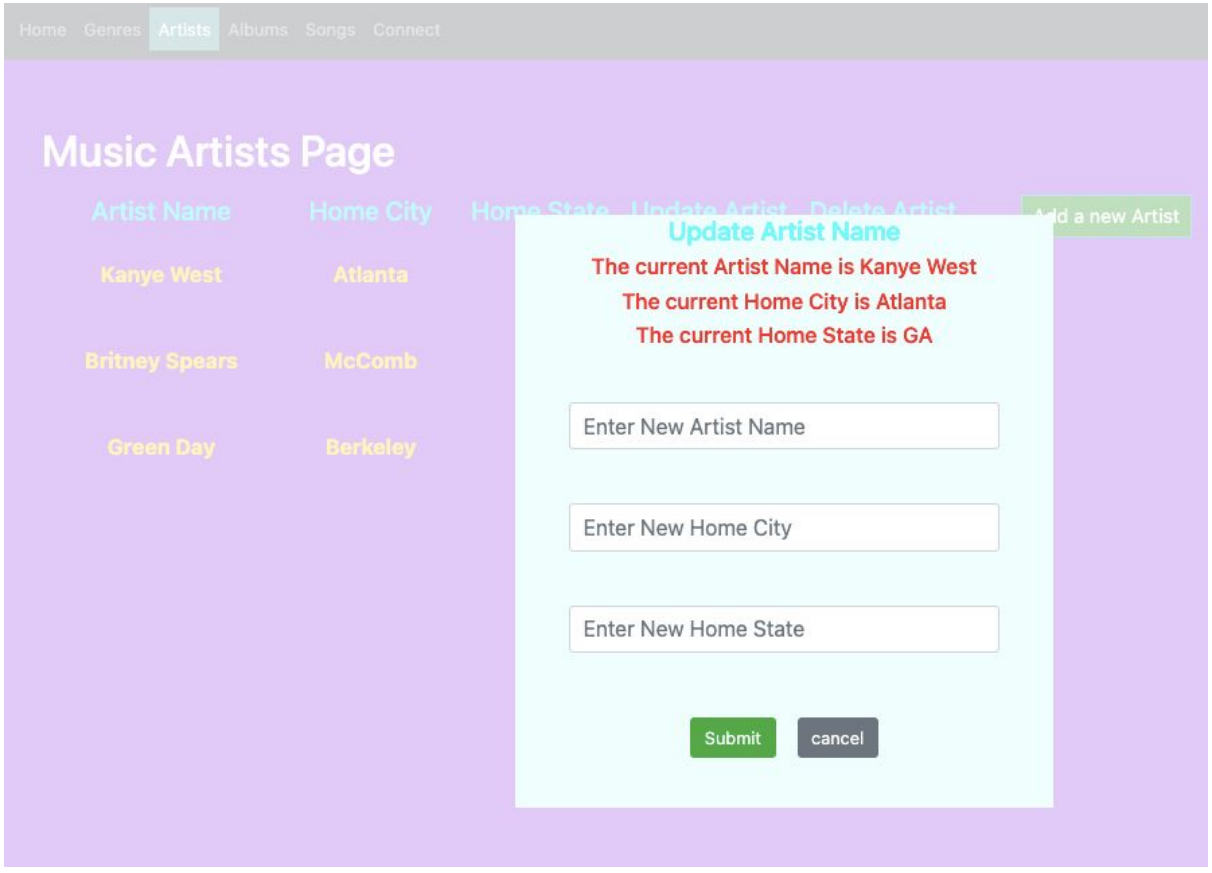
6. READ artists page



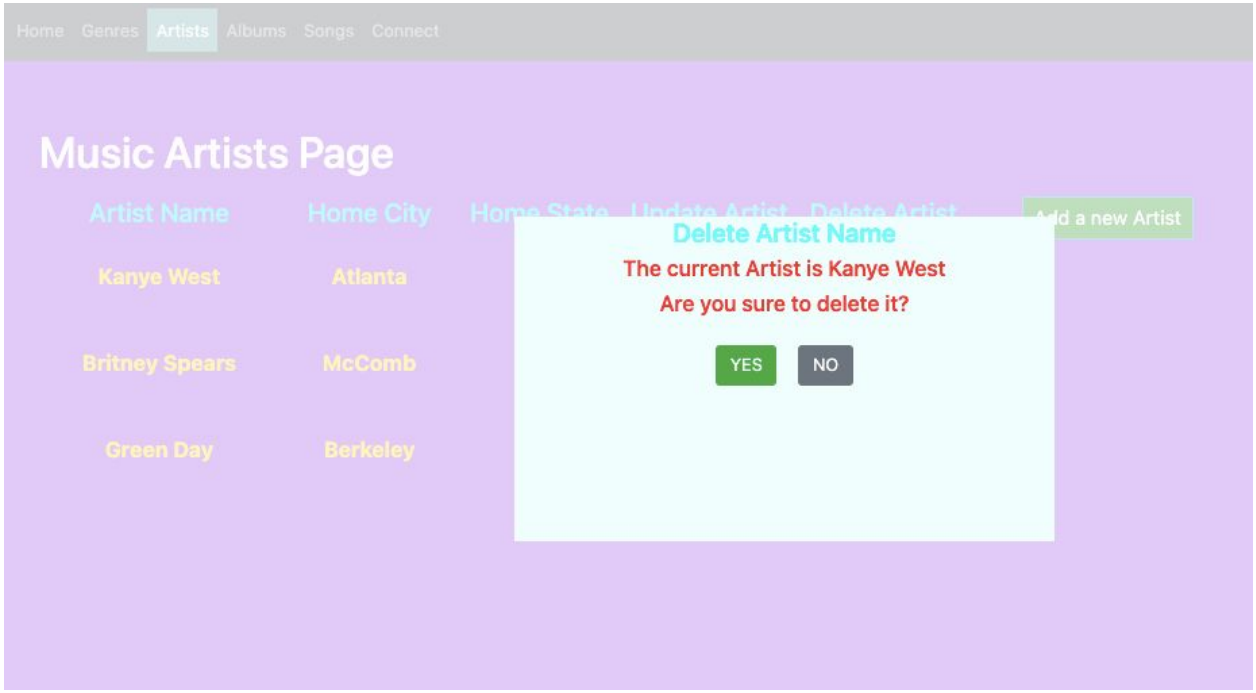
7. ADD artists page



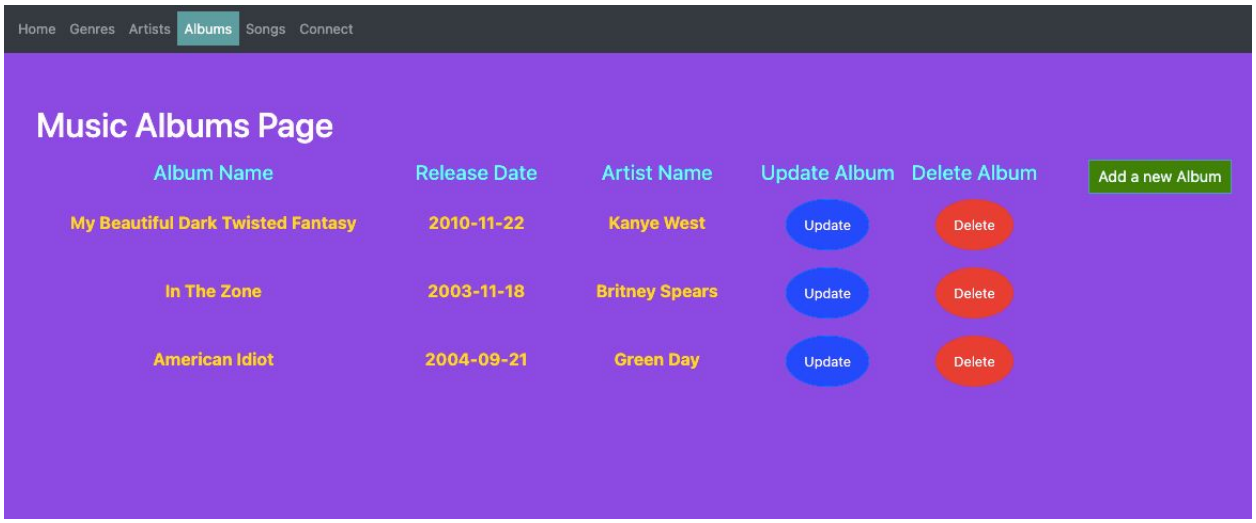
8. UPDATE artists page



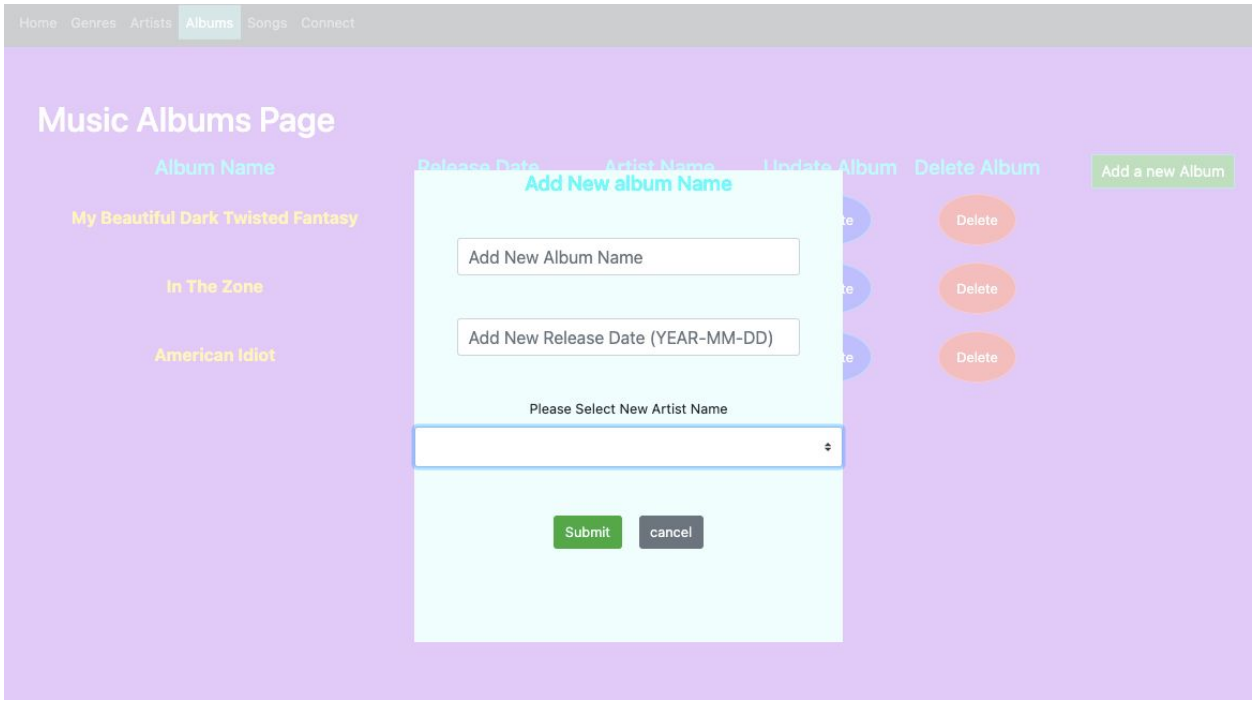
9. DELETE artists page



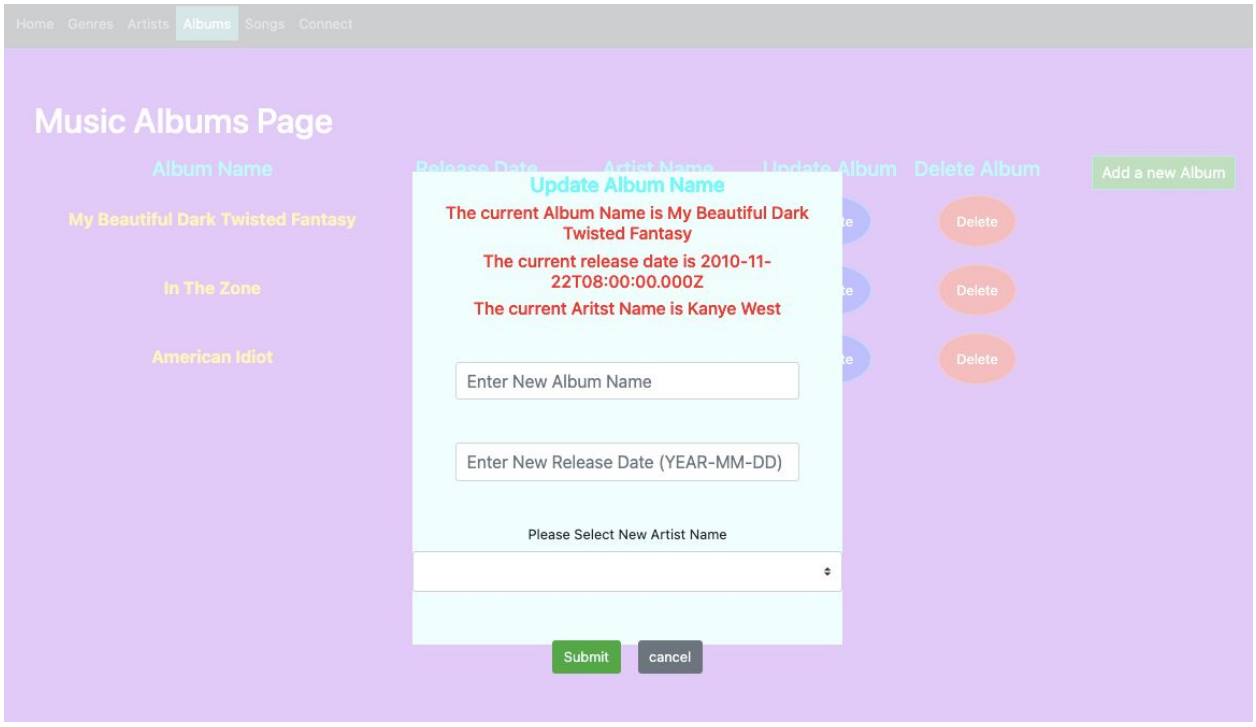
10. READ albums page



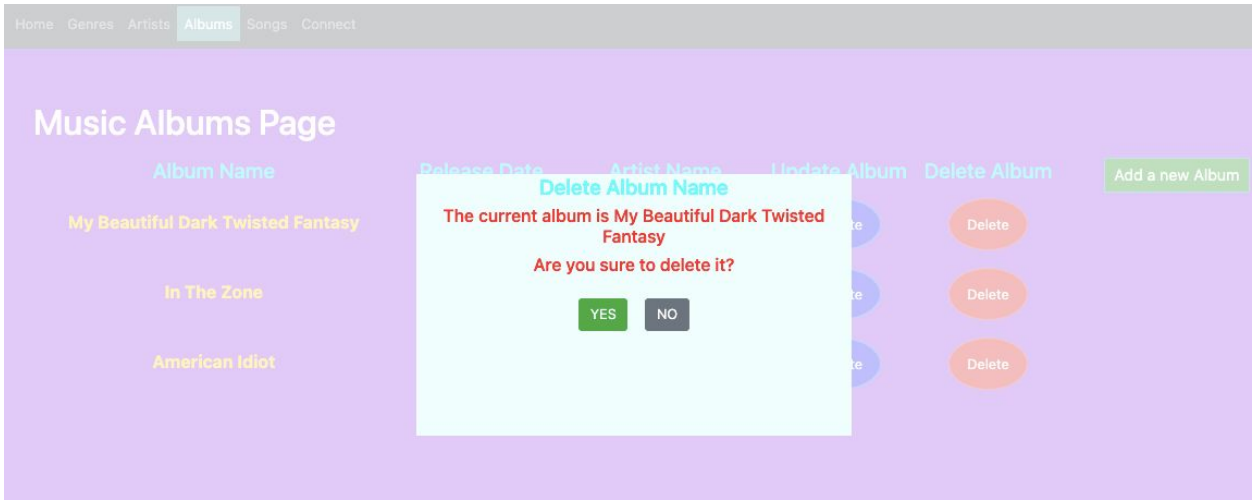
11. ADD albums page



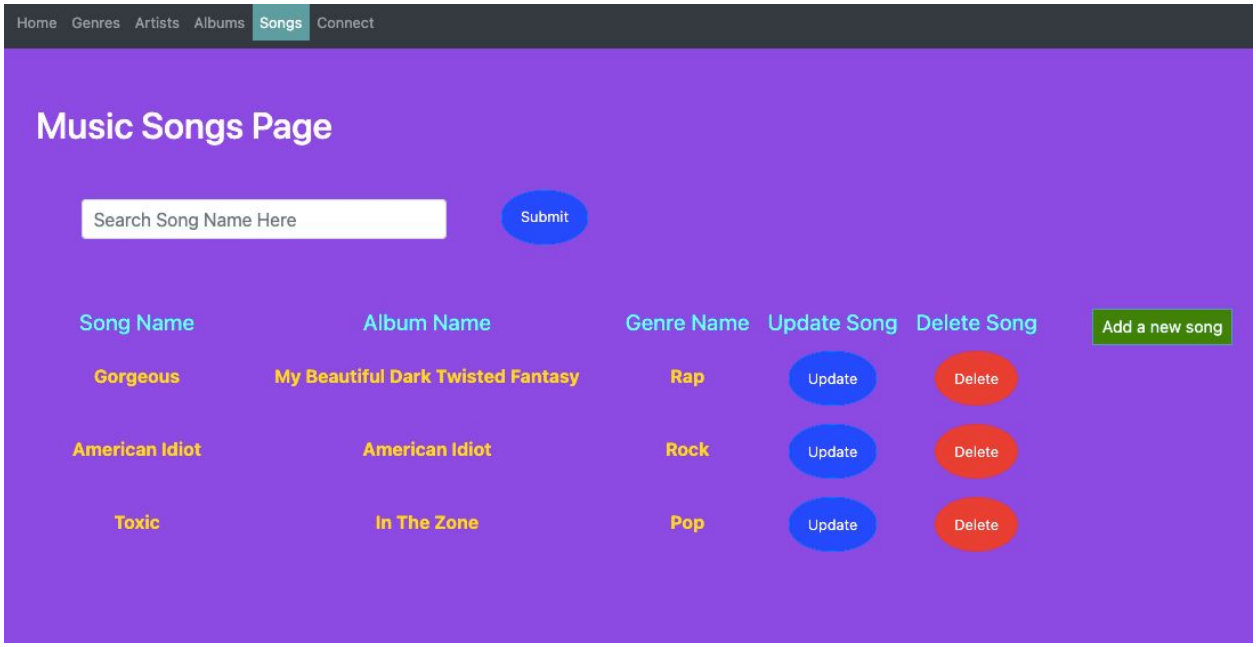
12. UPDATE artists page



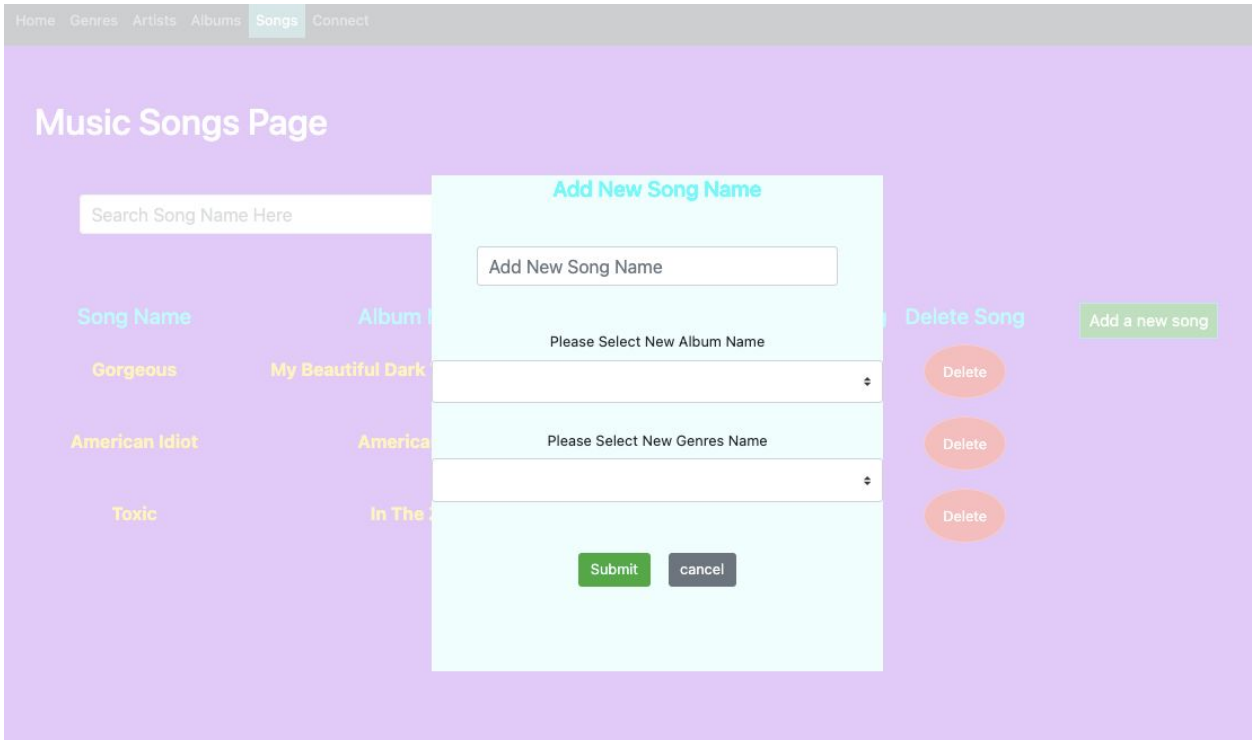
13. DELETE artists page



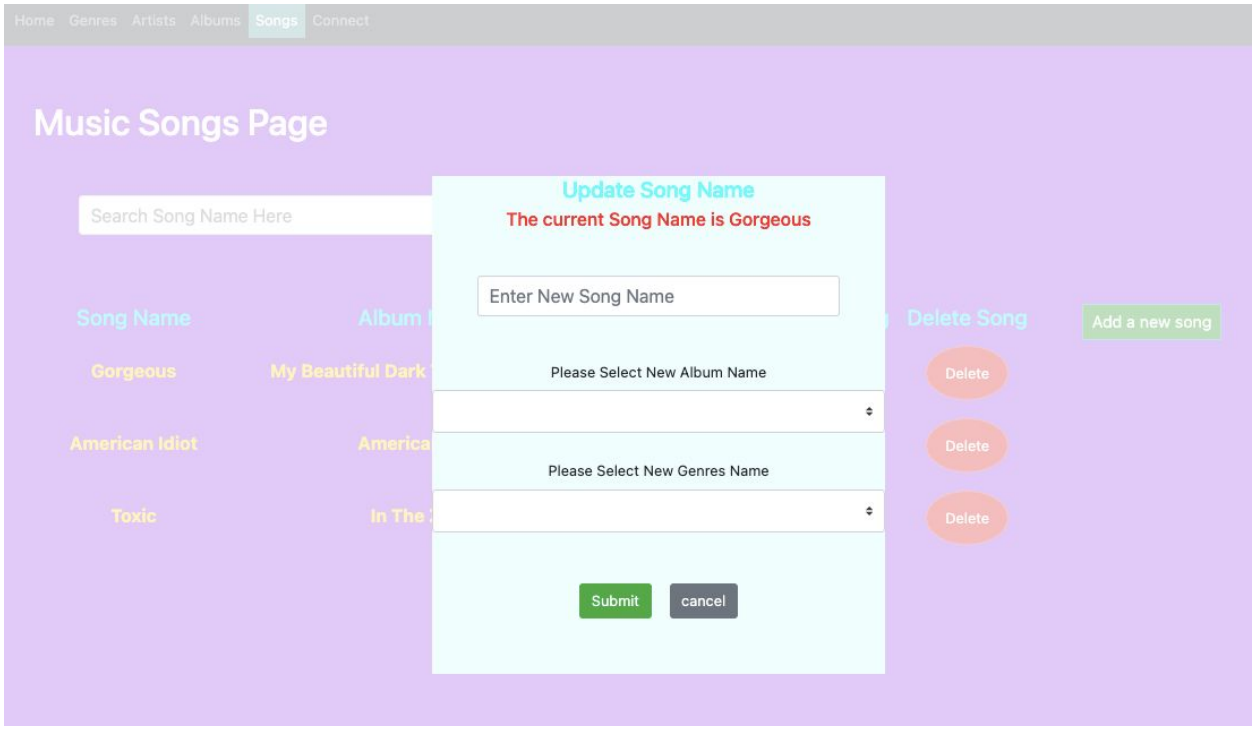
14. READ songs page



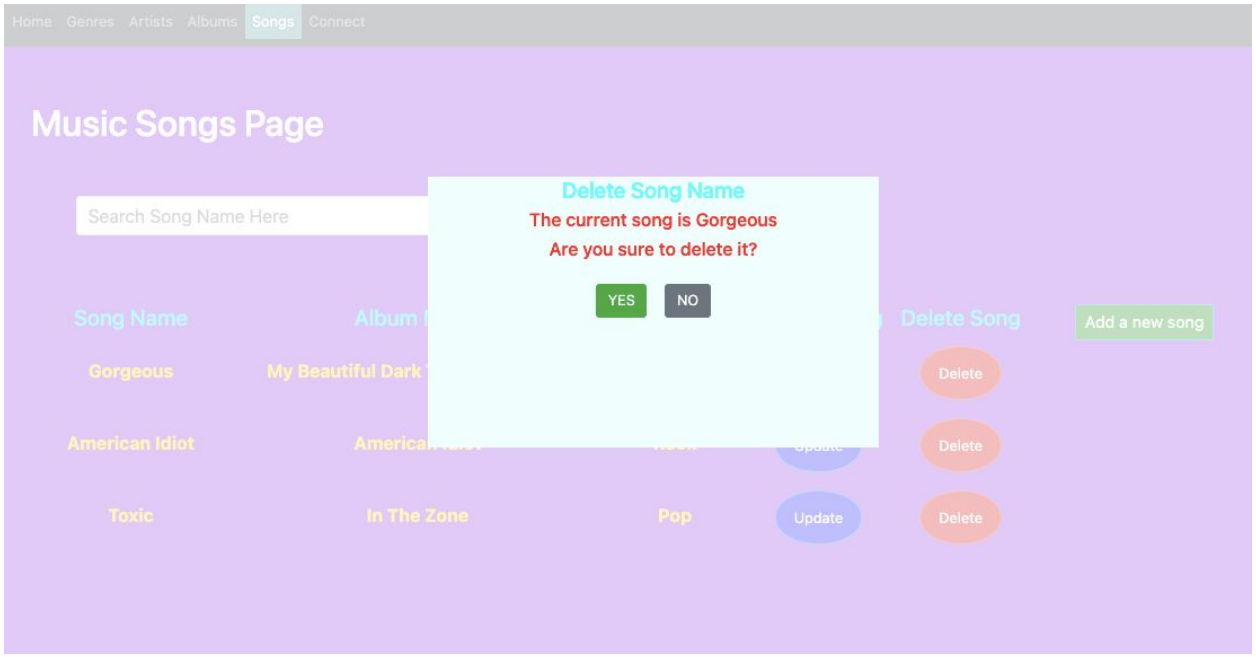
15. ADD songs page



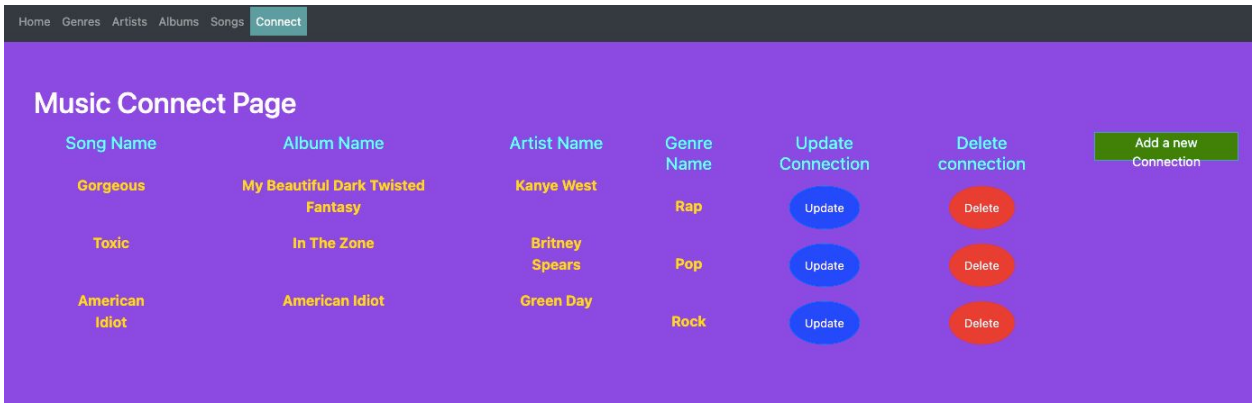
16. UPDATE songs page



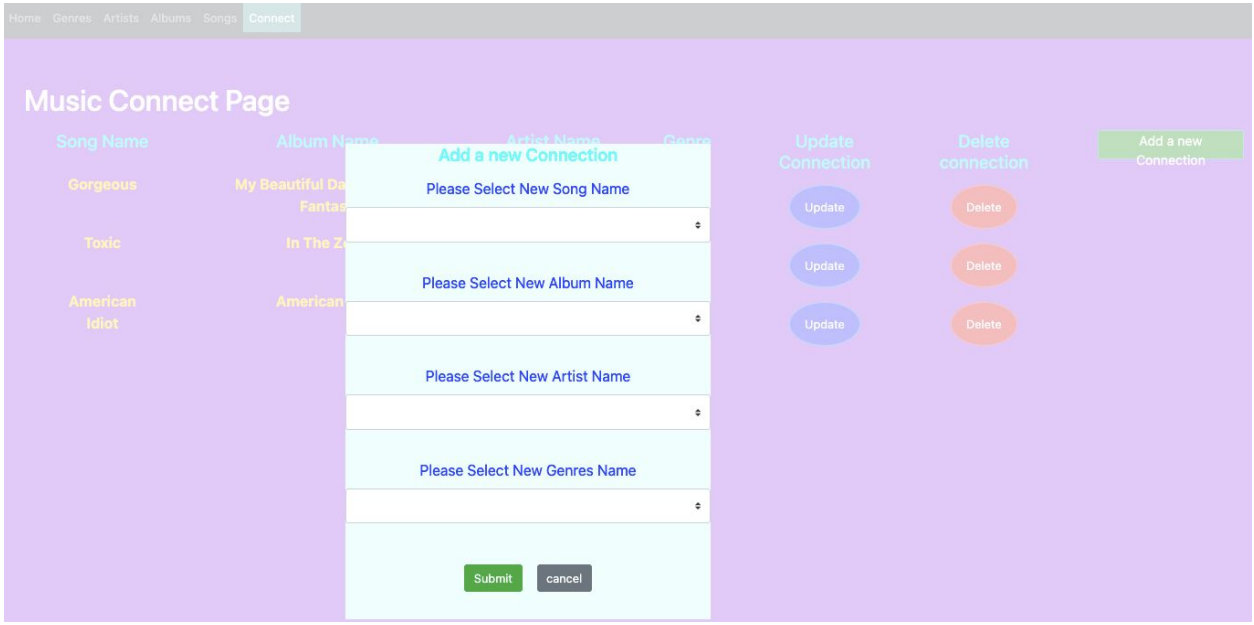
17. DELETE songs page



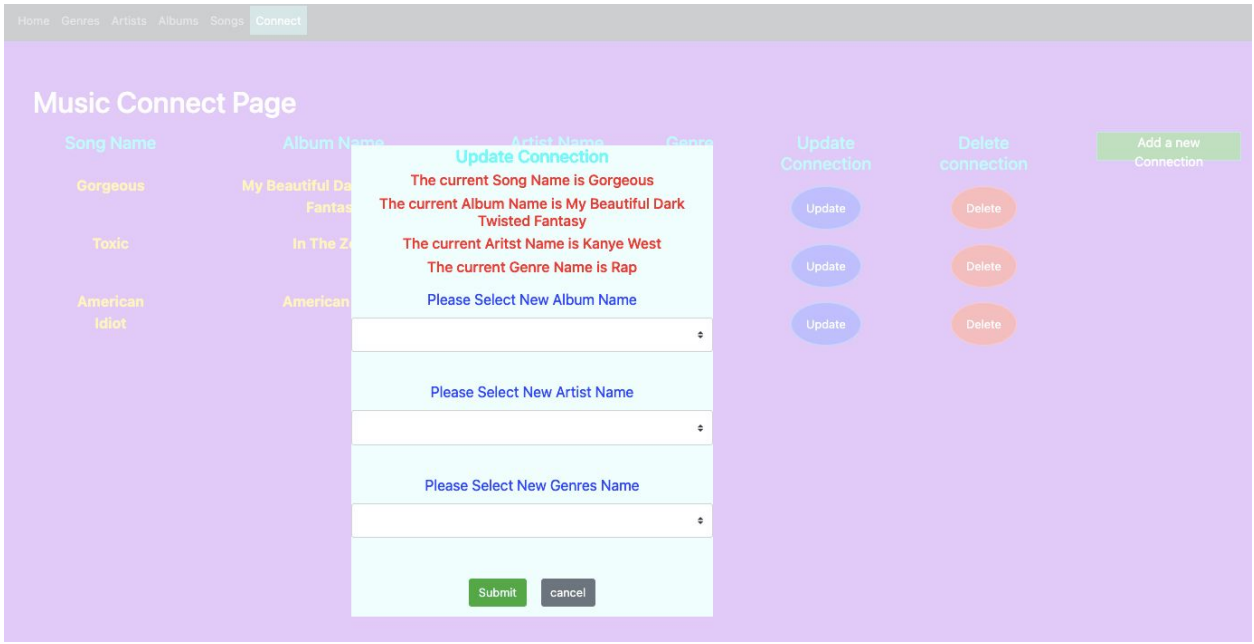
18. READ connect page



19. ADD connect page



20. UPDATE connect page



21. DELETE connect page