Andrea Mock
CS304
Prof. Scott Anderson
April 2, 2021

# CS304 Project - MusicShare

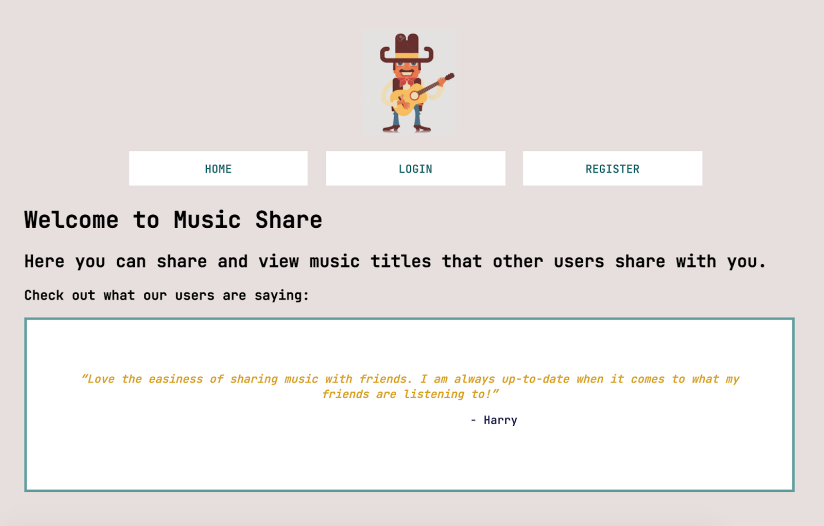
## 1. Introduction

As part of my database course, CS304: Databases with Web Interfaces, at Wellesley College, I implemented a music-sharing platform titled MusicShare. The project's goal was to apply skills learned throughout the course, including displaying, searching for, and updating data and using sessions and authentication when building a web application. I incorporated several of these aspects into my project. MusicShare allows users to log in with their username and password or create an account if they do not have one. Once a user has successfully logged in, they can view all the music they and others have added and search for friends who are part of the platform. A user can follow other users, add new music and update their profile and the music information they have added. The idea behind MusicShare is to create a more user-centered music sharing platform that is built through user collaboration rather than an outside company. Thus, the hope is that the platform will be more tailored to the users' interests and allow users to play a more active role in shaping what music is shared on the music portal.

This paper is structured as follows: I begin with a brief overview of how users interact with the MusicShare website and the different functionalities. Afterward, I discuss the technical details of the implementation and how files and information users upload are being stored in the backend. Finally, I conclude with a summary of why MusicShare should be considered an alternative to the more widely used music platforms.

## 2. Website Guide

MusicShare offers users multiple functionalities, including logging in, adding music to the database, updating a user's profile, and following other users. The following provides a brief walk-through of how to use the site:



Welcome to Music Share

Here you can share and view music titles that other users share with you.

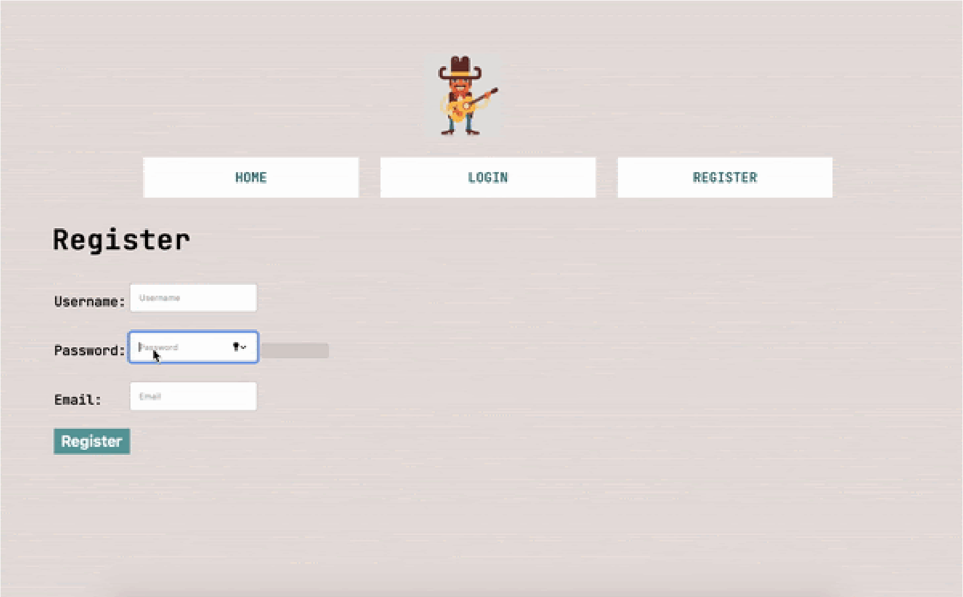
Check out what our users are saying:

*"Love the easiness of sharing music with friends. I am always up-to-date when it comes to what my friends are listening to!"*

- Harry

**Figure 1**: Landing page of the MusicShare site

The landing page introduces the site to the user and allows a user to log in or register for a MusicShare account (see Figure 1). To create an account, a user has to register with a username, password, and email. A password strength meter allows the user to evaluate their password's strength while they are typing it in (see Figure 2).



Register

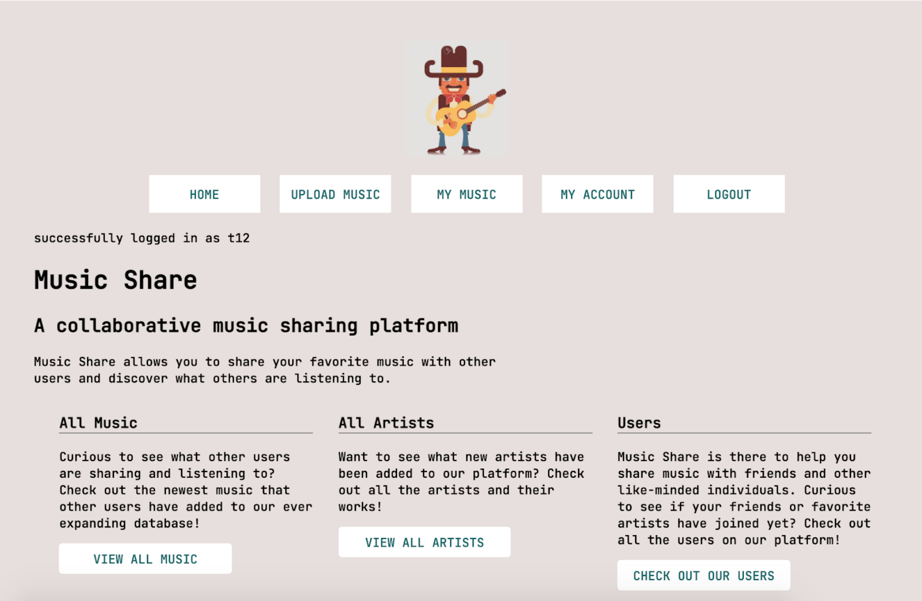
Username: Username

Password: Password

Email: Email

**Figure 2**: Registration page with dynamic password strength monitor

In the case a user tries to create an account with a username that already exists, an error message will be displayed, and the user will be asked to choose another username. If the registration process is successful, the user will automatically be logged in and gain access to the logged-in pages.



successfully logged in as t12

Music Share

A collaborative music sharing platform

Music Share allows you to share your favorite music with other users and discover what others are listening to.

All Music

Curious to see what other users are sharing and listening to? Check out the newest music that other users have added to our ever expanding database!

All Artists

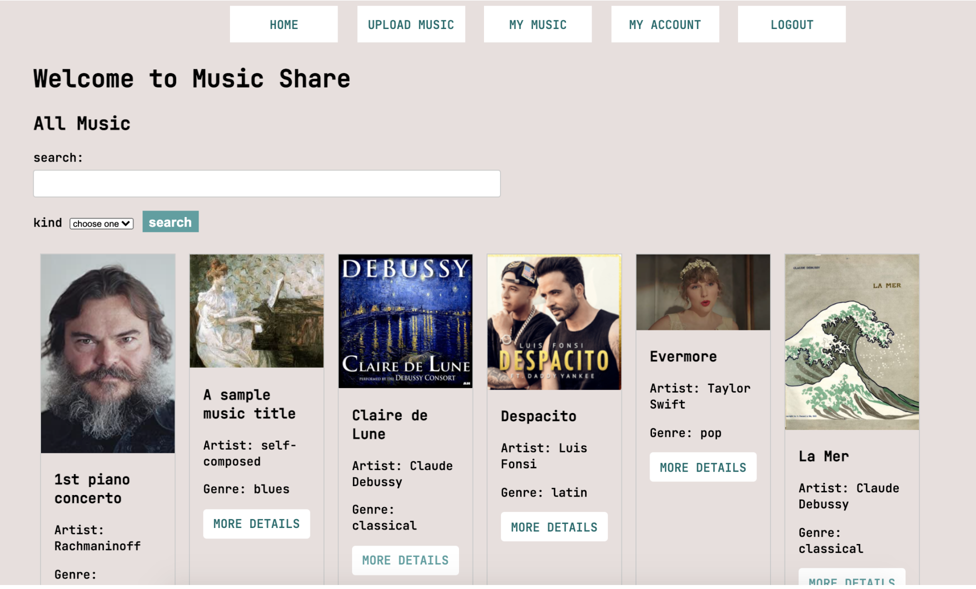
Want to see what new artists have been added to our platform? Check out all the artists and their works!

Users

Music Share is there to help you share music with friends and other like-minded individuals. Curious to see if your friends or favorite artists have joined yet? Check out all the users on our platform!

**Figure 3**: Homepage displayed when a user successfully is logged in.

Once logged in, the user is redirected to the logged-in home page (see Figure 3). From there, the user has the option to navigate to different pages. The user can access the all music page that displays the music currently part of MusicShare. They can also explore the current artists part of MusicShare and other users that have an account on MusicShare.



Welcome to Music Share

All Music

1st piano concerto
Artist: Rachmaninoff
Genre:

A sample music title
Artist: self-composed
Genre: blues

Claire de Lune
Artist: Claude Debussy
Genre: classical

Despacito
Artist: Luis Fonsi
Genre: latin

Evermore
Artist: Taylor Swift
Genre: pop

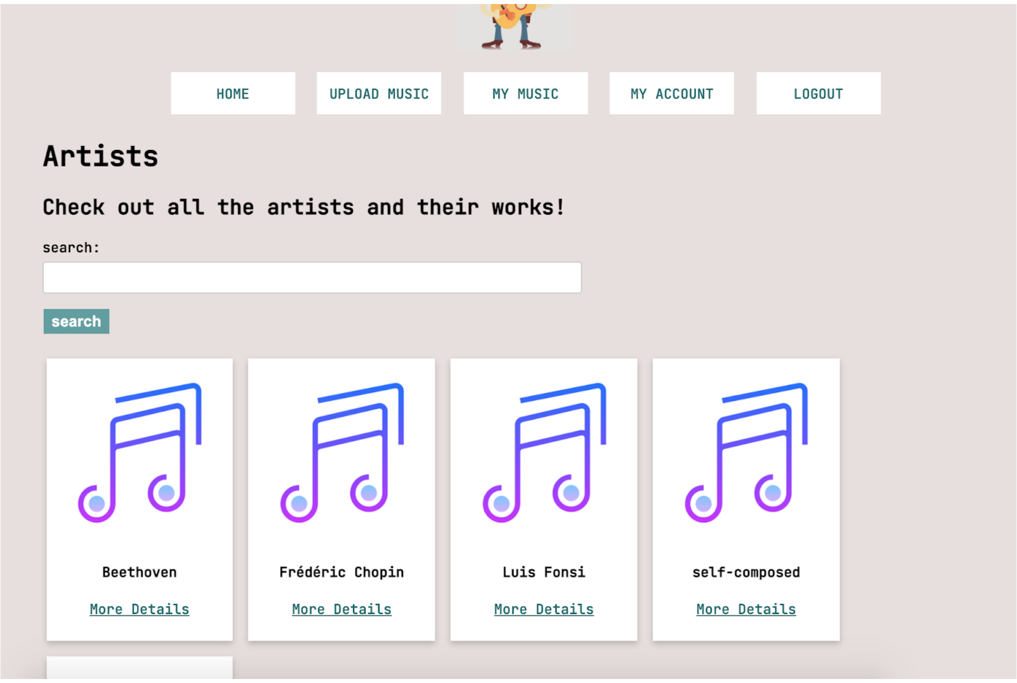
La Mer
Artist: Claude Debussy
Genre: classical

**Figure 4**: Display of the page with all the music on MusicShare

If a user wants to see all the music currently on MusicShare, they can navigate to the page displaying all the music from the homepage. The page displaying all music (see Figure 4) offers information on the name of a music title, its genre, and the artist. In addition, by clicking on the more details link that is part of each music title, a user can access more information about that particular music title. The additional information page also allows the user to easily navigate to the composer of the specific piece and the user who added it. Both of these links create an easy way for a user to discover more titles by an artist and find users who are adding music that they might enjoy as well.

Because the page displaying all the music can become quite large, the user can choose to search for an artist or a music title thanks to a search bar. If there are no search results that match a user's query, an error page will appear and tell the user to search again. If only one artist or music title matches the user's search query, they will automatically be redirected to that artist's or music title's details page. If, instead, multiple titles or artists match the query, all of them will be displayed. A user can then navigate to the pages of artists or music titles to see more details.

Besides viewing all the music titles that are part of MusicShare, a user can also access all the artists that are part of the platform via the homepage. Similarly to the all music page, a user can click on a musician's name to learn more about a particular artist and use the search bar to search for an artist (see Figure 5).



Artists

Check out all the artists and their works!

search:

Beethoven

More Details

Frédéric Chopin

More Details

Luis Fonsi

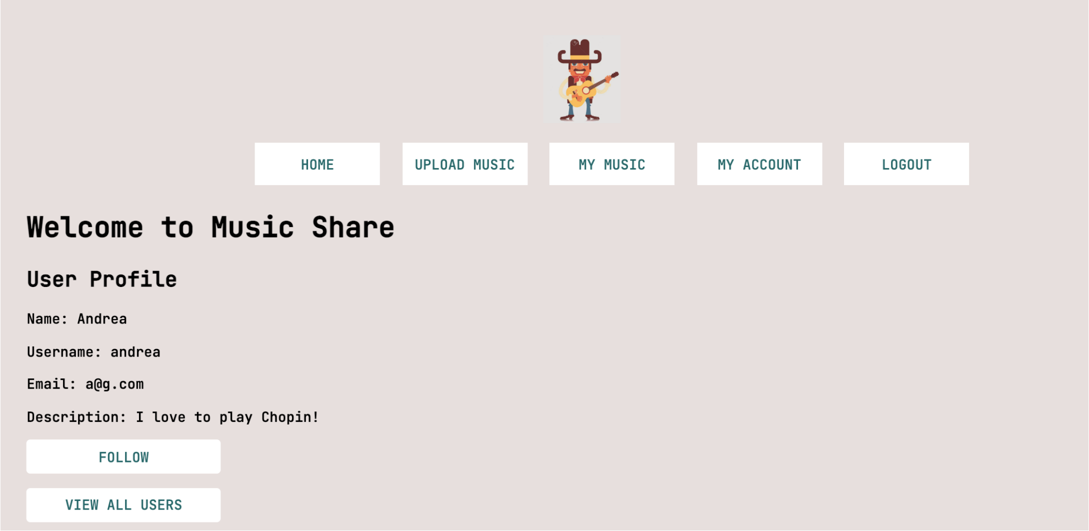
More Details

self-composed

More Details

**Figure 5:** Page displaying all of the artists.

The homepage also allows users to explore all the users registered on MusicShare. If a user knows a friend's username, they can use the search bar to access their friends' profiles or scroll through all the user profiles. A user profile includes the basic information a user used to register, including their username and email. However, a user can also decide to add more information to their profile, including their name and a short description of themselves. Since MusicShare is there to foster a collaborative environment, a user can choose to follow users, which they can easily do through other users' detailed user profile pages. In addition, if a user no longer wishes to follow another user, they can also unfollow the person when they can go to that user's page and click on unfollow (see Figure 6).



Welcome to Music Share

User Profile

Name: Andrea

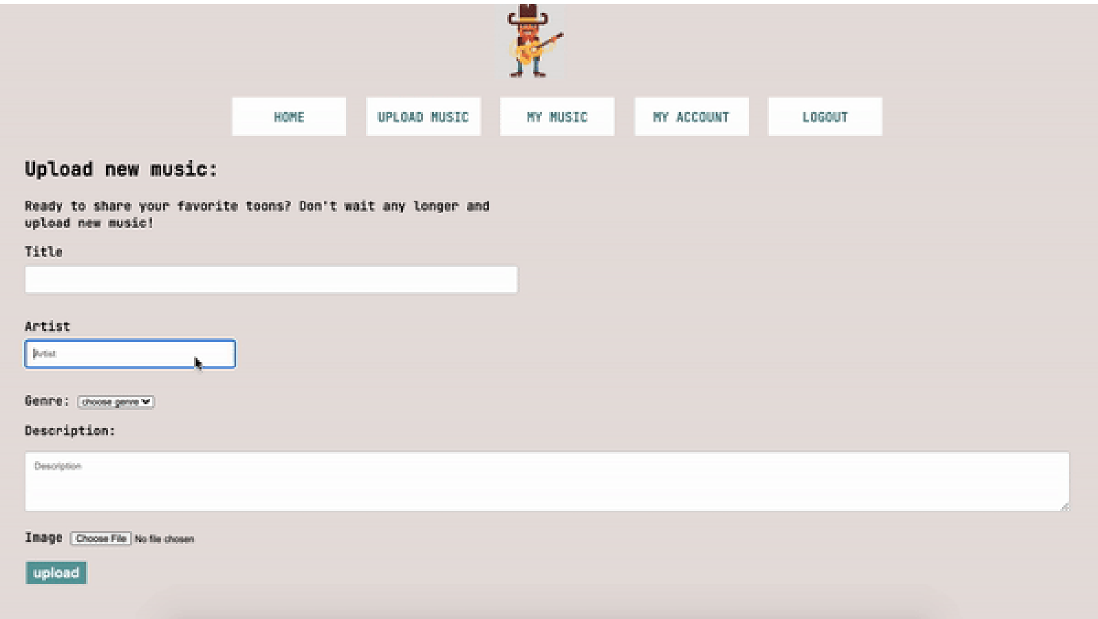
Username: andrea

Email: a@g.com

Description: I love to play Chopin!

**Figure 6**: Example of the a user's profile page

The navigation bar offers the user easy access to pages pertaining to their profile and music. Thus clicking on the my music box in the navigation bar, the user will be directed to a page where they can see the music they have added to MusicShare. If a user is new to the platform, they will not have any music shown on and instead see a message telling them to add music to get started. Also, part of the navigation bar is a link to upload new music.



Upload new music:

Ready to share your favorite toons? Don't wait any longer and upload new music!

Title

Artist

Artist

Genre: choose genre

Description:

Description

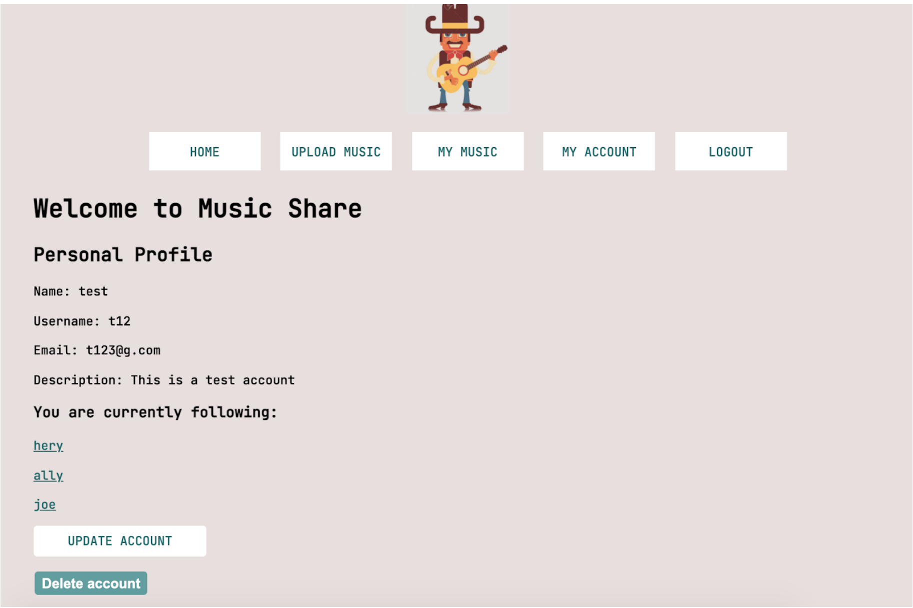
Image Choose File No file chosen

**Figure 7**: The upload music page that allows user to add music to MusicShare

When accessing the upload new music page (see Figure 7), a user is prompted to enter the name of the music they are adding, the artist, genre, an image, and an optional description to upload a music title. The image can have a .png, .jpeg, or .gif extension. If a user uploads a file that does not have one of these formats, they will not be able to upload any music.

A user can also update the music information that they have added to the database. By navigating to the detailed view of a music title they have added (either through the ‘my music’ page or accessing it through the all music page), they can update the genre, artist, and title. By allowing the user only to edit and delete titles they have added themselves, it avoids the issue of maliciously updating and deleting music that other users have added.

A user can access their profile page via the navigation bar. A user’s profile page allows users to update their profile and add additional information, including their name and a short description of themselves. The personal profile page also displays whom the user is currently following (see Figure 8). If a user is not following anyone, a message saying start following other users will appear. If a user no longer wishes to be part of the platform, they can delete their account. The navigation bar allows the user to log out easily, in which case they will be redirected to the welcome page displayed before they were logged in.



Welcome to Music Share

Personal Profile

Name: test

Username: t12

Email: t123@g.com

Description: This is a test account

You are currently following:

hery

ally

joe

**Figure 8**: View of the personal profile page

## 3. Technical Details

In part 2, I discussed the general way a user can interact with the MusicShare platform. Next, I will delve deeper into the technical aspects of the implementation of MusicShare. MusicShare was implemented using Flask, a Python web application framework. The data collected and used in MusicShare was stored using a MariaDB database instance running on the Wellesley College Tempest server.

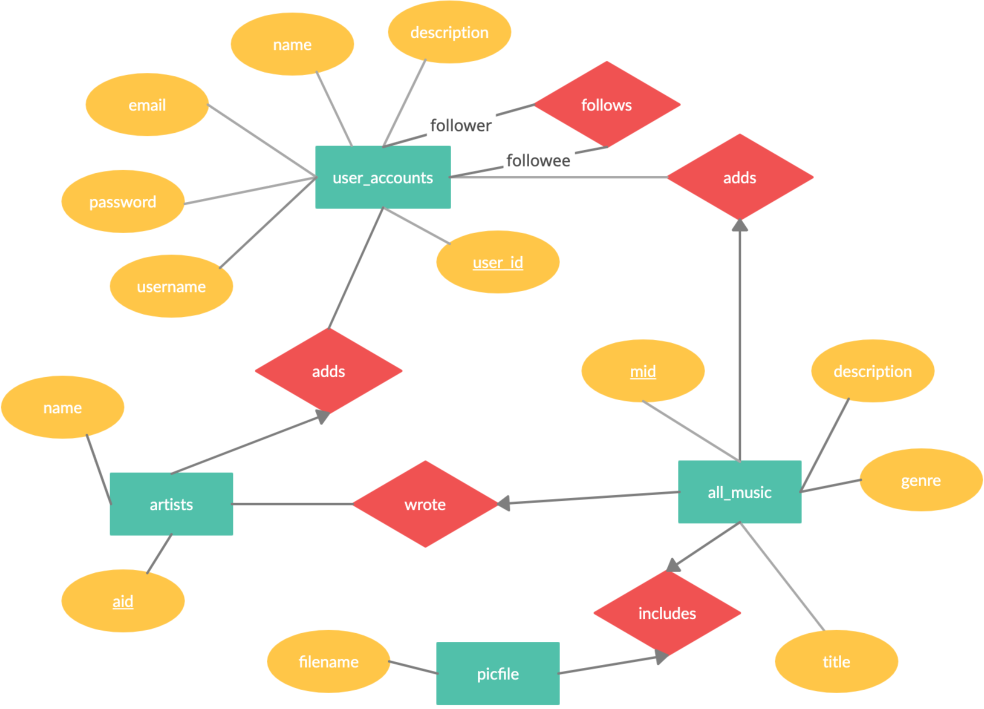


**Figure 8**: ER diagram corresponding to the MusicShare platform.

Figure 8 shows the entity-relationship diagram related to the MusicShare platform. I will go through the most critical aspects of the diagram and its implementation, beginning with the user_accounts.

Handling and saving user information:

MusicShare allows users to add, delete and update music. These are actions not everyone should be able to do. Thus users are required to login to perform such actions. The rationale behind having a user login is to track who is doing what on MusicShare. Each user has a username, password, and email. The username is the unique identifier for each user and corresponds to an id they are automatically given when creating an account. If a user tries to sign up with a username that already exists, an error message will be displayed telling them to choose another username. In Flask, this means that I used a try-except clause. First, try to insert the new user information into the database. However, if one gets a duplicate key error, one knows that this user already exists and thus can display this to the user trying to register for a new account. In addition, when a user logs in or creates an account, their password is encrypted using the Python library bcrypt. The hashed password can then be stored and retrieved from the database. It safeguards the private information of each user and makes hacking into someone's account harder.

Using sessions:

Sessions are used to check the login status of a user. Once a user logs in, the login status is changed to true in the session. In addition, the user id is saved in the session. By keeping the user id in the session, when the user tries to access their profile or the music they have added, it makes sure that it only retrieves the music and user information corresponding to the user id saved in the session. Also, the session allows us to check if the user is authorized to access particular routes. For example, a user cannot change the profile of another user by simply changing the URL to access the personal profile page of another user since the Flask app will always check if they are supposed to have access to that particular URL. If not, they will be redirected to the home page. Also, the session storing the user id will help display the music a user has added to the database. It does so by looking at the all_music table, a table that includes the music information contains a foreign key matching the user who added the particular title. One can then make an SQL query that retrieves all the music titles added by that user.

Once a user decides to log out, the logout code clears the user information stored in the session and sets the logged-in status to false, meaning that the user will no longer be able to access pages reserved to logged-in users.

Operations on the music and artists table:

One of the most important tables in our database is the all_music table which holds the information surrounding all the music titles in our database. Each music item has a unique id, as well as details including genre, the filename of the image corresponding to the music title, and foreign keys that reference the user who added a music title and the artist id of the artist who wrote that particular music. With the music id (mid), one can access the page displaying information corresponding to that specific mid. In addition, by using a left outer join on the all_music table and the artists' table holding the artists' information, one can retrieve an artist's name.

The artist information is saved in an additional table since it allows for easier searching for a particular artist, accessing their data, and avoiding duplicates. When a user decides to add or update a music title, they will be asked to enter or edit the artist information. The forms used to add or update a music title automatically suggest matching artist names already part of the database to avoid having duplicate artists stored in the database. The autocomplete feature avoids the issue of users accidentally misspelling a name and adding the same artist under a different spelling to the database. The autocomplete feature is implemented with jQuery, which dynamically displays the names of artists matching what the user is typing (see Figure 7).

Because multiple users will be able to access the site simultaneously, meaning our Flask app is multi-threaded, one will also have to think about concurrent database transactions. When adding or updating a music piece, one first checks if the entered artist already exists. If not, add the artist to the artist table and automatically assign them an artist id (aid). Since another user could simultaneously add the same artist, the all_music table and artists table are both locked before trying to check if a particular artist is part of the database. If not, they are added to the database. After these two queries are executed, the tables can be unlocked.

Handling Files:
When a user adds music to the database, they are also uploading an image that corresponds to the music title. When submitting the form for upload music, the image is saved to the filesystem in an “uploads” subdirectory of our Flask app. To avoid having files with the same name, which could be overwritten, a random id using the UUID library is generated, which produces a universally unique identifier that guarantees with almost 100 percent certainty that each file will have a unique name. Before a file is added to the filesystem and the name is saved to the database, one checks the size of the file to avoid people from uploading content that is too large (threshold of 1 MB) as well as only allowing uploads of certain kinds (jpeg, gifs and png files). By having these safeguards in place and making users log in before they upload files, one avoids many security concerns related to unrestricted file uploads.

Handling errors and pages that do not exist:
Lastly, I will briefly cover how I dealt with routes that do not exist. If a user tries to access routes that do not exist will display a 404 not found page if a user is logged in. Otherwise, an error message will be flashed, and the user will be redirected to the welcome page. Similarly, if they try accessing another user’s update profile page, they will be redirected and not allowed to perform such an update.

## 4. Conclusion

MusicShare allows people to crowdsource their favorite music and find new music to explore. In this paper, I briefly discussed the different functionalities of the site and some of the technical implementation details. The simplicity of the site makes it user-friendly and allows users to contribute information about their treasured music tracks. MusicShare, unlike many other music sharing platforms, is committed to storing a user’s information safely and not collecting unnecessary information. The registration form offers a password strength meter for convenience, and the user only has to input minimal information to create an account.