
Presented to the Software Technology Department

De La Salle University - Manila

Term 1, A.Y. 2021 - 2022

In partial fulfillment

of the course

In **Web Application Development** (**S11**)

# MACHINE PROJECT DOCUMENTATION

Submitted by:

**Sarmiento, Raymund John M.**

Submitted to:

**Arturo P. Caronongan III**

**February 23, 2022**

## Project Background

The web application that has been developed can be used to upload, store, load, and play music files that the user uploads to the database. It is similar to something like iTunes but instead of having the files stored locally on the user's computer, it is stored remotely in a MongoDB database. It is limited in functionality but the core concept has been achieved in the application. The application is named Tasogare in reference to Mai Yamane's 1980 album of the same name. The project was initially conceptualized as a music sharing application much like SoundCloud, but due to difficulties and time constraints, the creation of user accounts was unachieved.

## Project Modules

* : module not implemented

**Register***

A visitor must register if they want to upload, comment, or download music. Here, a visitor must enter their username, their password, an avatar, and an optional short description.

**Log in***

After registering properly, a visitor may log-in. Upon logging in, the user can start uploading, commenting, and downloading. The user is given the option to be "remembered" by the website. When the user chooses this option, they will remain logged in that device until they log-out themselves.

**Log out***

The user may log out from their account. This clears any session-related data.

**View account details***

The user may view their account details including their username, avatar, and short description.

**Edit account details***

The user is allowed to edit their account details, which includes the username, password, avatar, and short description. The user will be asked to enter their current password to save their edited account details.

**Delete account***

The user is allowed to delete their account. All posts and comments related to the user will be removed from the web app.

**View all posts**
Upon visiting the web page, an unregistered visitor may see 10 rows of music and artists. Each row may be a different grouping, one may be a row of artists/posts by a certain individual, another may be of a category of music. Clicking on a song will expand it. In the case of a song, the user will be able to listen to the song and see the comments and reviews on it.

**View a user profile***
Each user has their own page which shows their profile publicly. On the same page, a visitor may see the user's username, avatar and short description. They may also see a portion of the user's latest posts and comments. The visitor may opt to see the rest of the posts and comments of the user.

**Post**
A user may upload songs to the website. The name of the song and its details will come from the embedded details in the song's file by default, but the user should have the option to change any details of the song for the website. If the uploaded audio file does not have any embedded details, the user is required to input them manually before it is posted. Only MP3's or similar compressed audio file formats will be accepted. Uncompressed audio file formats such as FLACs and WAV files will not be accepted. The user should also give the option of whether the song they uploaded can be downloaded or not by other users.

**View a song***
A user may view and listen to any upload. This will allow them to view the details of the song, and the comments that have been made on it.

**Downloading a song**
Users may download songs if the option has been placed there by the uploader.

**Edit an upload***
The owner of the post may edit the details of their uploads at any point.

**Delete an upload***

The owner of the song may delete their uploads.

**Comment***
A user can comment on any song, including their own. They may also reply to another user's comments, and the comments can nest indefinitely.

**View a comment***
A user may view any comment on a song he is currently viewing.

**Edit a comment***
The owner of the comment may edit their comments at any point.

**Delete a comment***
The owner of the comment may delete their comment.

**Upvote***
A user can upvote a song or a comment (including their own) once.

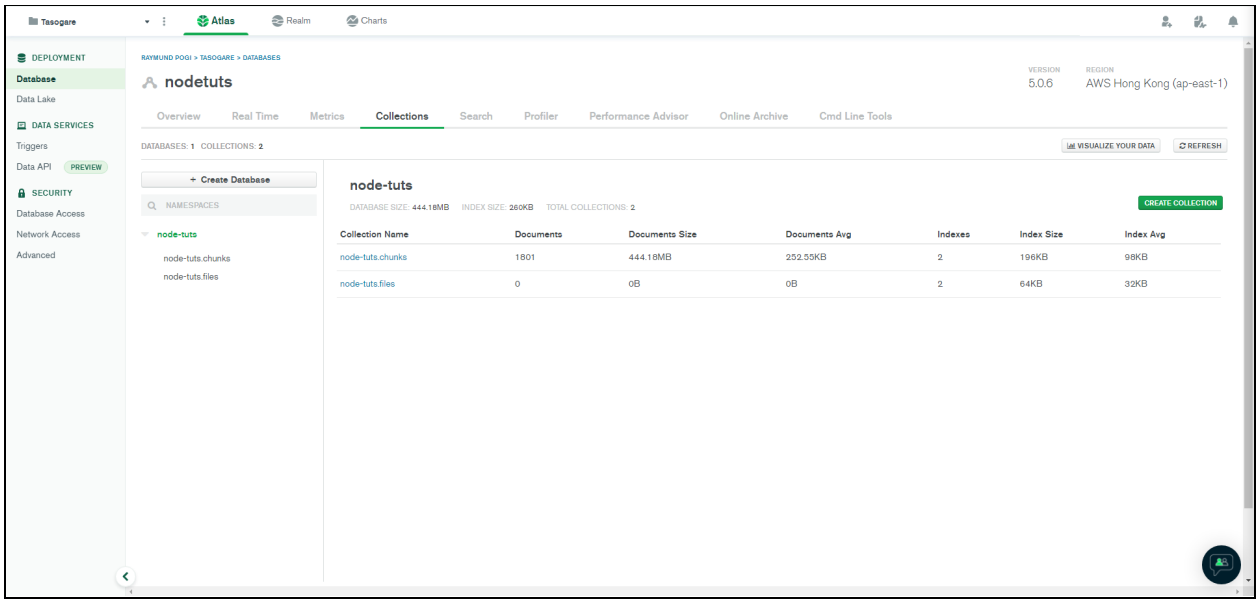
**Downvote***
A user can downvote a song or a comment (including their own) once.

**Search a song/artist***
A visitor/user can search for songs by title, category/genre, or artist. By entering a search phrase/word, all songs containing it will appear as results.

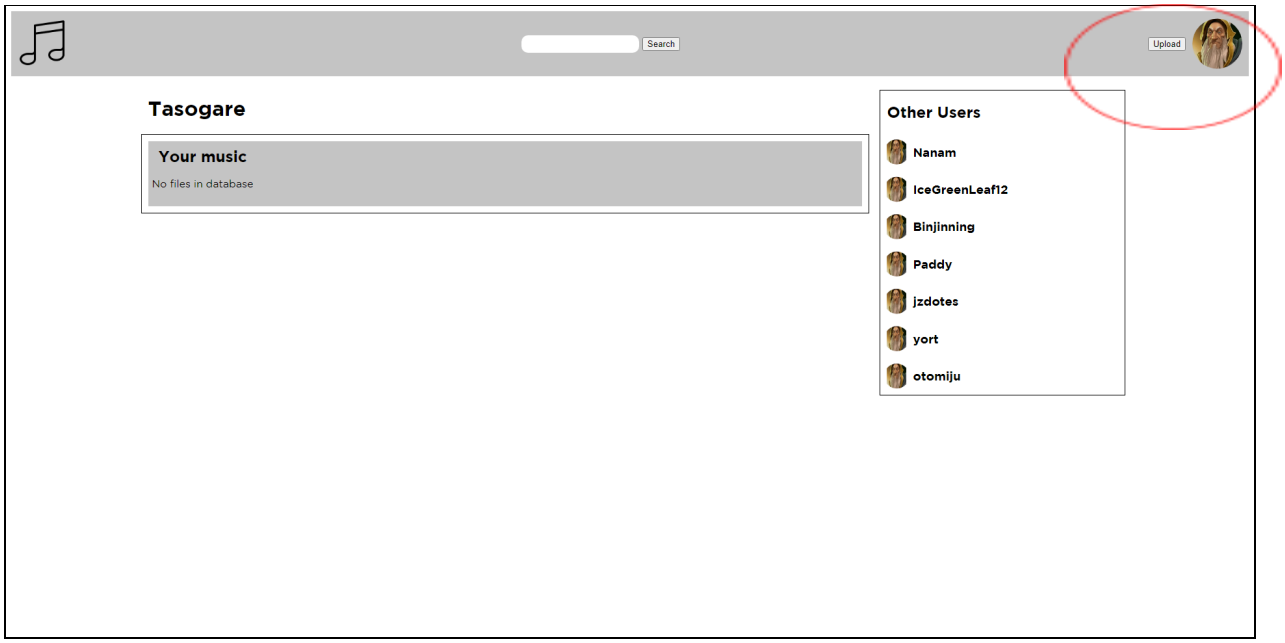
## Project Functionality

Upon opening the application, the user will be greeted with the home page which contains the header that has a search bar (non-functioning) and the upload button, all the songs that are in the database under that, and other users of the application on the right. If there are no songs in the database, the feed will display a text saying "No files in the database".

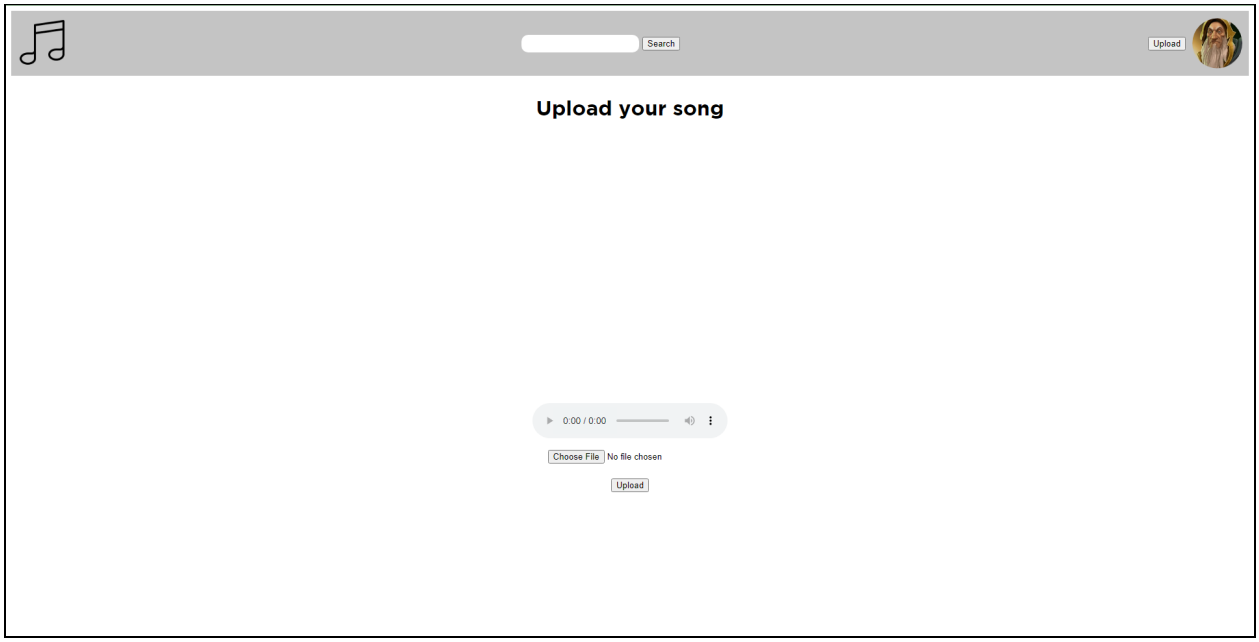

The database used is with MongoDB and it holds two collections to store the uploaded music/audio files. This will be further expanded upon later in the document.



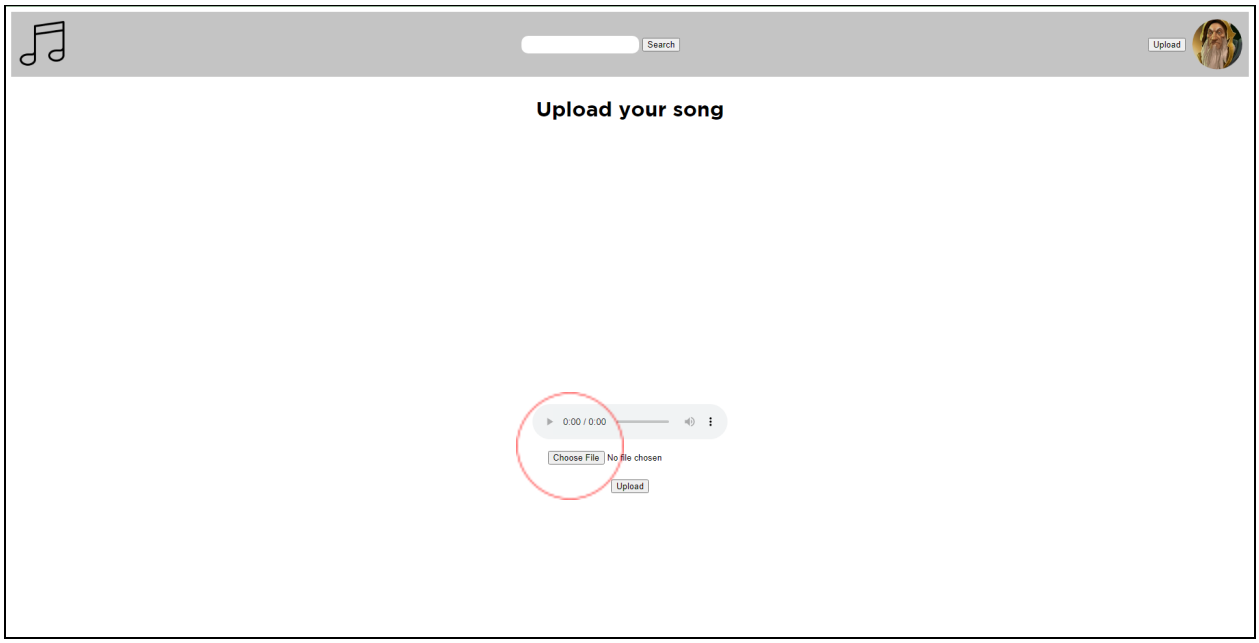

To upload a song, click the upload button located on the top right of the screen



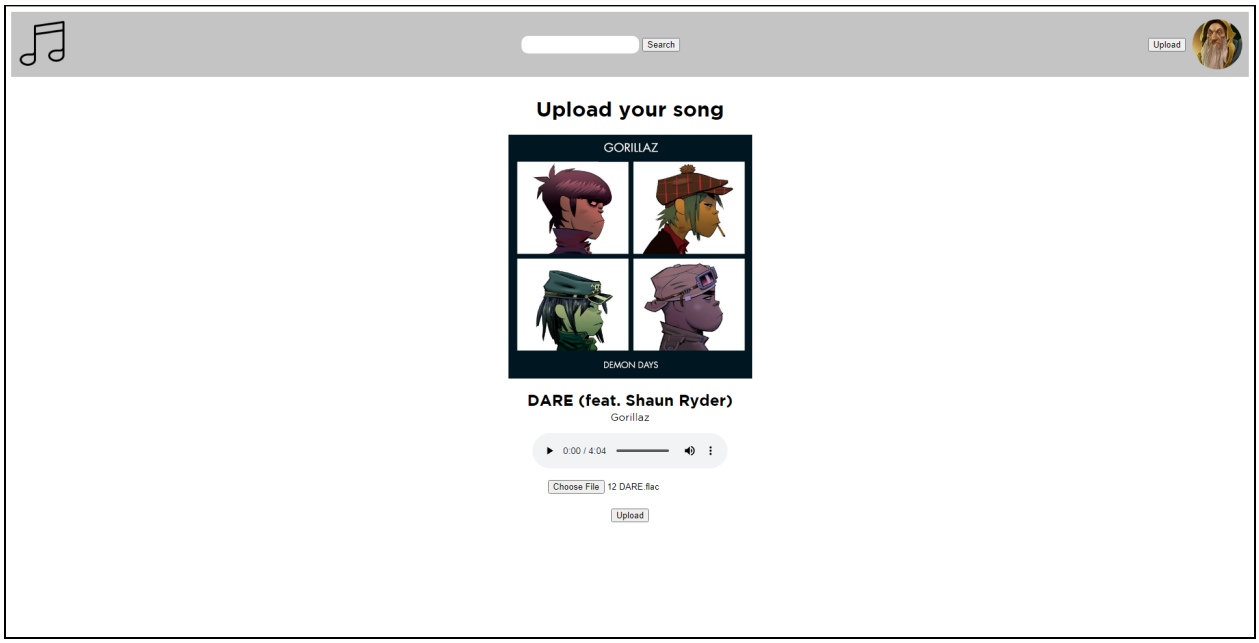

After clicking on upload, the upload screen will appear and this is where the user can select and upload a song from their computer.



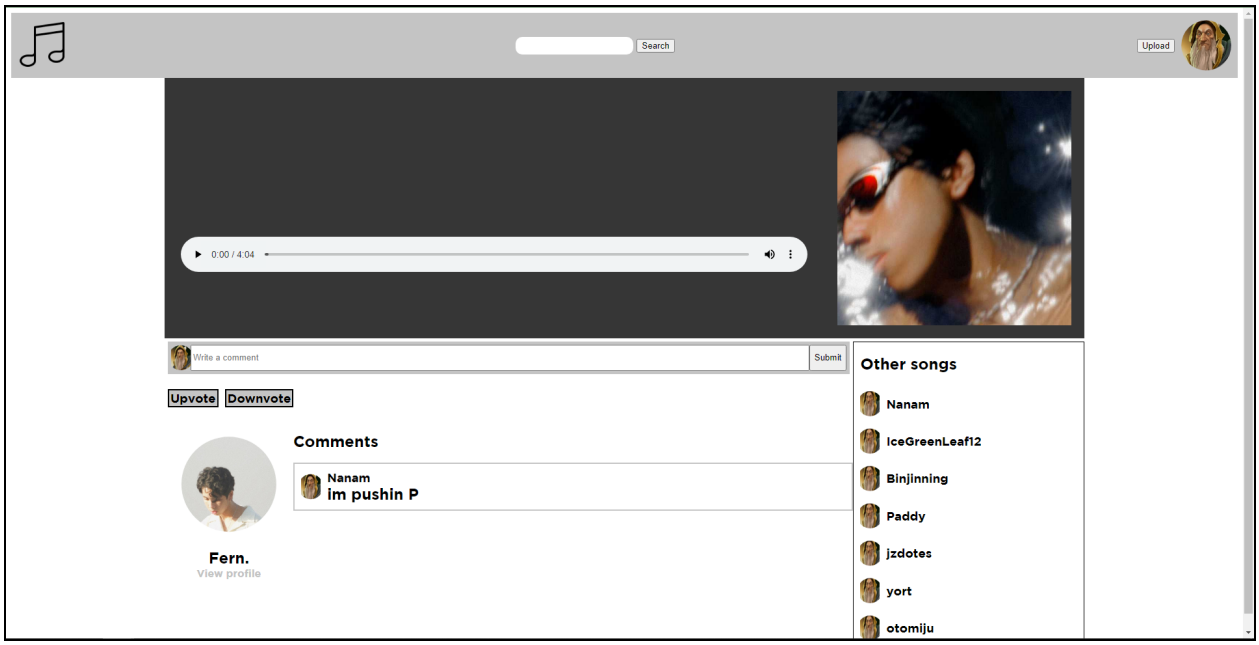
Click the ‘Choose File’ button to select a file. The application will only accept music/audio files that have metadata in it. It can accept audio files in mp3, m4a, flac, and wav formats.




The selected file will appear on screen with the song title, artist, and cover art. All of this data is read and accessed through the files metadata or id3 tags. From here, the user can already play the audio file that they have selected.

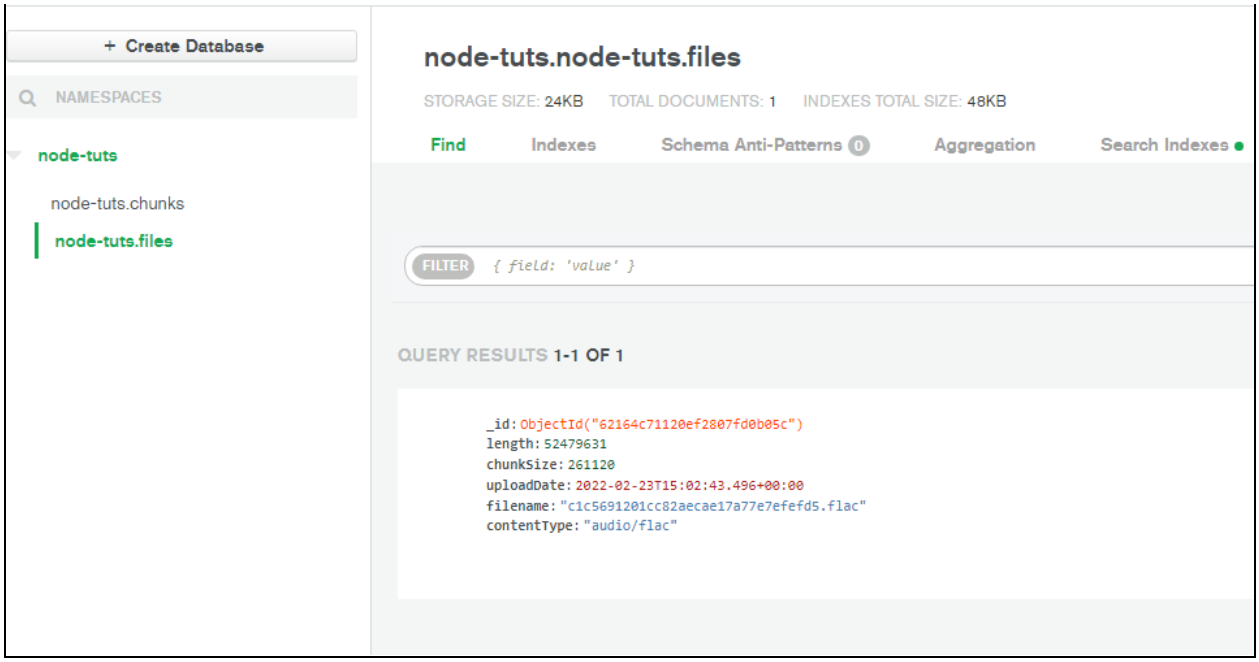



After clicking ‘Upload’, the user will be redirected to the ‘View Song’ screen where they will be able to play the audio file that they have uploaded. The file will be loaded from the MongoDB database in this page. The cover art and other data of the song could not be retrieved and are not accurate in this page. The user can also download the song.

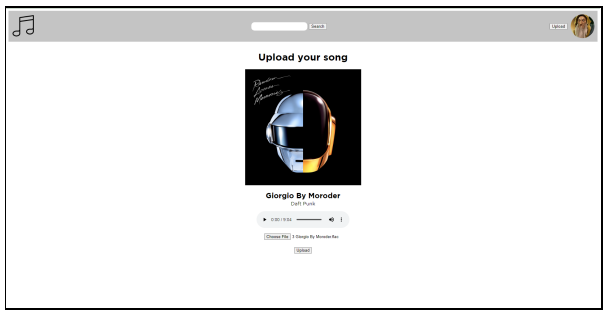


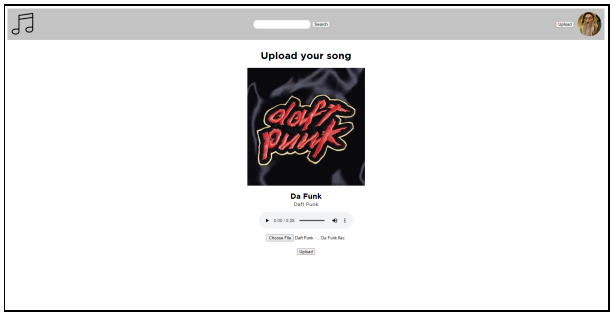
The user can return to the home page by clicking on the musical note icon on the top left of the screen. The audio file that has been uploaded can now be seen here in the home page.

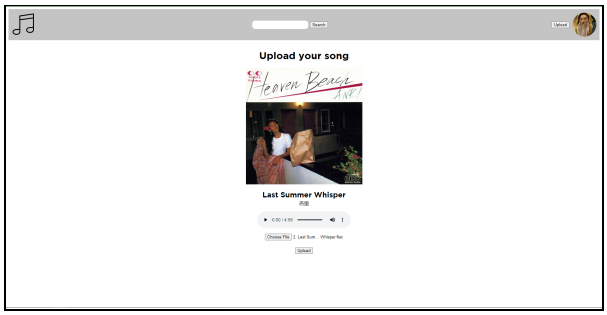

When looking at the database, we can now see that the ‘node-tuts.files’ has one document in it. This is representative of the audio file that has been uploaded.

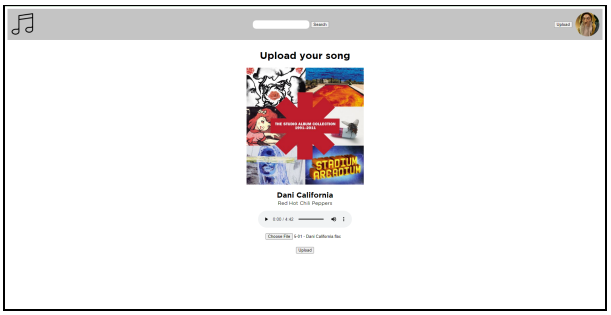


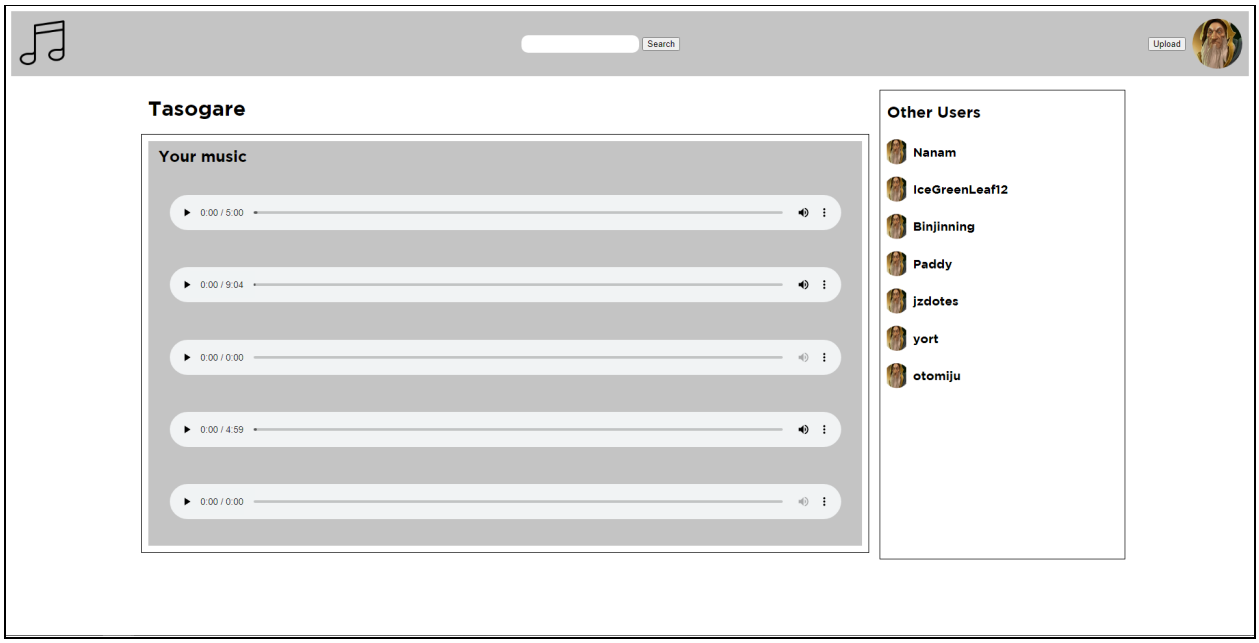

We can upload more files to see the database filled with more files and have more songs in the home page.

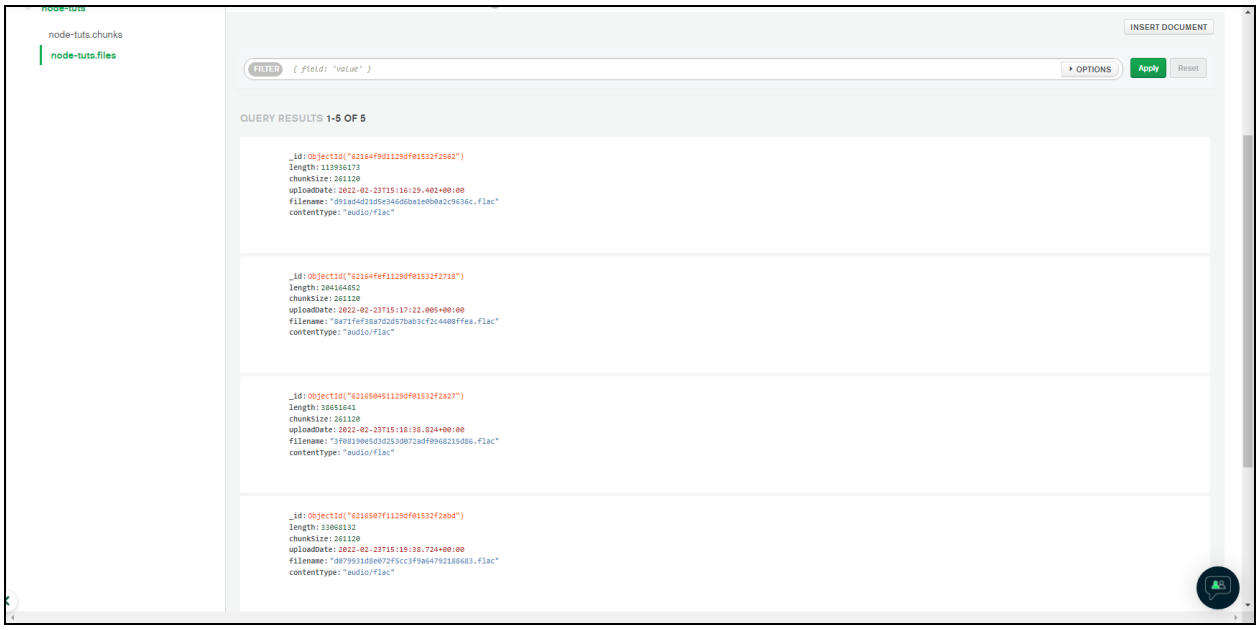


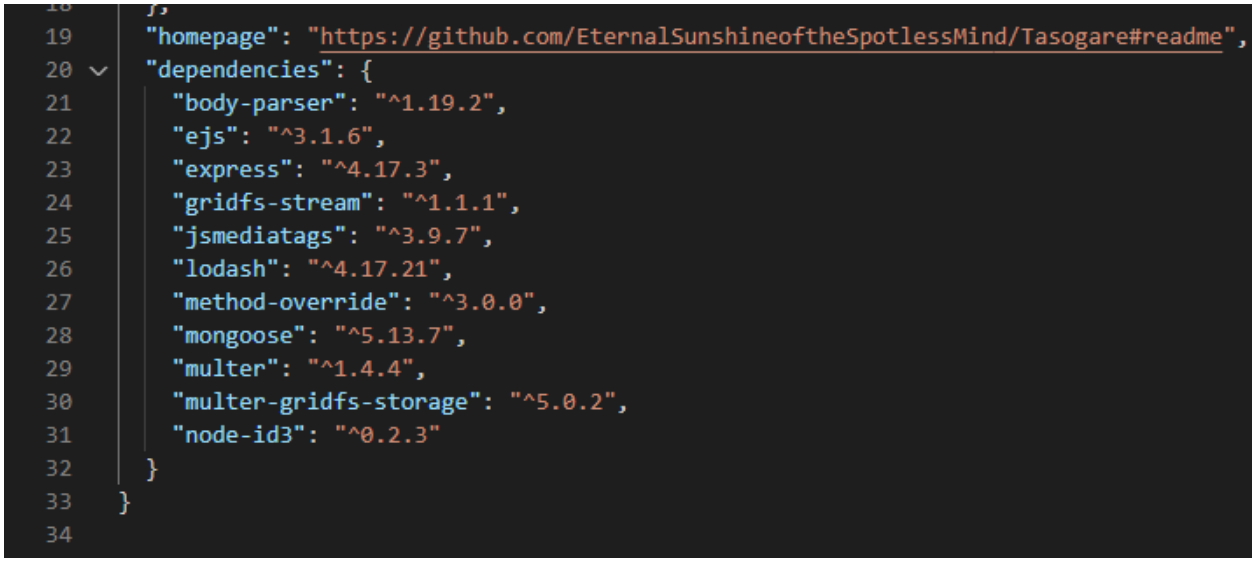



All the songs in the homepage can be played and downloaded freely.

## Backend Development

Dependencies used:




An older version of mongoose was used due to an error occurring when using a gridfs-stream function.