Melesio Albavera

Ethan Brushwood

Kaleb Kruse

Quentin Ponder

Professor San Yeung

Introduction to File Structures and Databases

6 December 2023

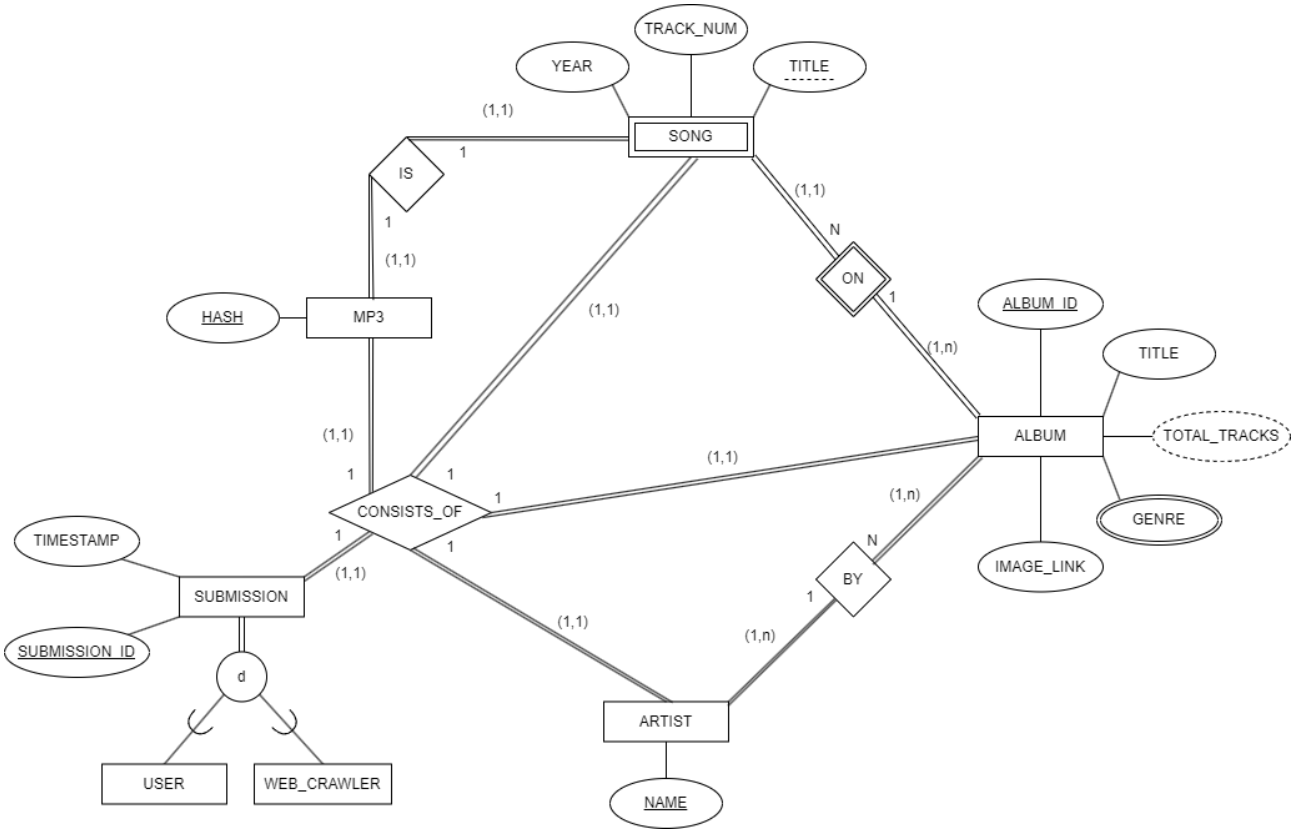
Database Application For Tunes:

ID3Tag Lookup Database

Revised Problem Statement

The metadata associated with audio files in the MP3 format, commonly adhering to the ID3 standard, has seen several iterations over the years and can be used to attribute up to 63 distinct informational fields. However, since the standard is not officially part of the MP3 format, many sources of audio tracks do not include ID3 tags, resulting in the task of categorizing falling on the shoulders of individuals compiling audio collections for personal or commercial use. This task, particularly the retrieval of the relevant information, becomes arduous even for a handful of files. We propose the development of a command-line tool for fetching a limited set of ID3 tags associated with a given user-provided query and returning the metadata in an easily parsable format.

Revised Conceptual Database Design



The model above was updated to more accurately reflect participation constraints. In the diagram above the following assumptions are made:

1. An artist is uniquely identifiable by their name.
2. Every song (including singles) belongs to an album.
3. No two songs on the same album can share the same name.
4. An artist cannot make 2 albums by the same name
5. Only one artist can contribute to an album (although, creating a separate artist i.e. "Person 1 & Person 2" would work)

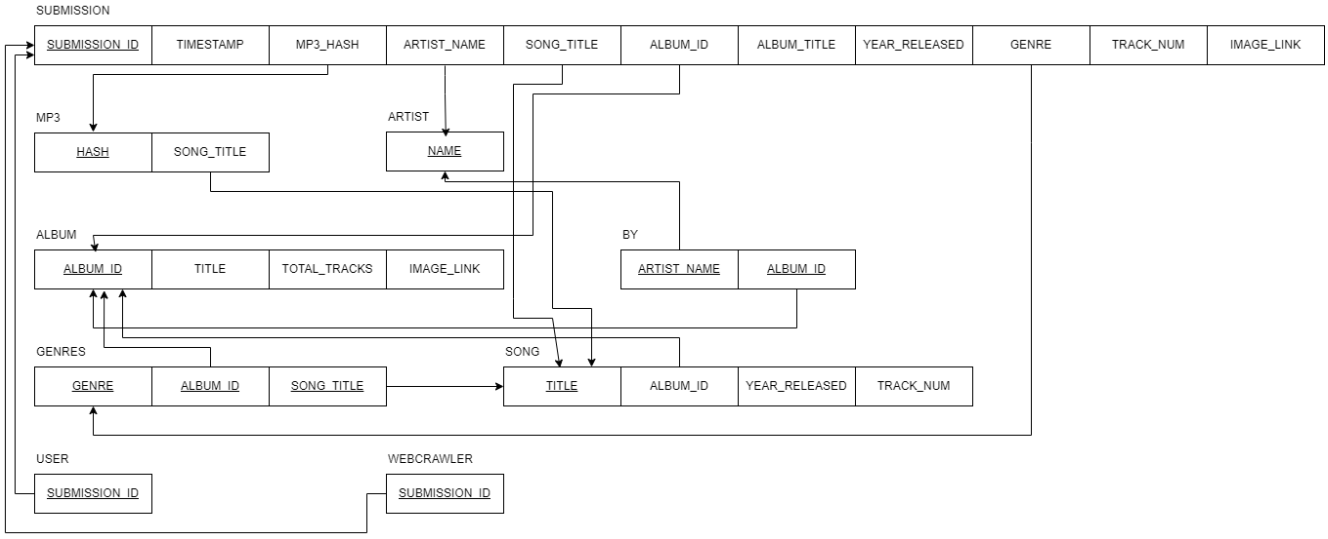
Functional Requirements

The following is a revised overview of the functions our proposed application would provide:

1. Allow users to query the database via a command line interface and return a JSON file containing either the relevant ID3 tags or an appropriate error message. We propose support for queries of arbitrary MP3 files (on which a hash search is performed.) Should the query yield a failure message, this will be noted and the user will be prompted to supply this information, should they wish to.
2. Automated data aggregation through the use of a web crawler bot. This program would acquire relevant data from relevant websites such as YouTube, Spotify, Wikipedia, Genius, etc.
    2.1. Proposed tags include Artists, Title, Album, Number of Tracks on Album, Track Number, and Album Art Image Link.

3. A script for processing MP3 files. Essentially, this would accept an MP3 file from the user, query the database via the file-hash lookup method, and allow to user to append the retrieved data to the end of the input files.
4. A graphical user interface that accepts an arbitrary MP3 file and appends the relevant metadata, with similar functionality to the command-line tool.
5. In order to verify the veracity/integrity of the metadata provided, users are allowed to adjust information that they believe to be false.

Revised Logical Database Design



Revised Summary Table of Data Types

Table	Attribute	Type	Constraint
SUBMISSION	SUBMISSION_ID	INT	Primary Key
SUBMISSION	TIMESTAMP	TIMESTAMP	NOT NULL
SUBMISSION	MP3_HASH	STRING	Foreign Key
SUBMISSION	ARTIST_NAME	VARCHAR(255)	Foreign Key
SUBMISSION	SONG_TITLE	VARCHAR(255)	Foreign Key
SUBMISSION	ALBUM_ID	INT	Foreign Key
SUBMISSION	ALBUM_TITLE	VARCHAR(255)	NOT NULL
SUBMISSION	YEAR_RELEASED	INT	NOT NULL
SUBMISSION	GENRE	VARCHAR(255)	Foreign Key
SUBMISSION	TRACK_NUM	INT	
SUBMISSION	IMAGE_LINK	TEXT	
MP3	HASH	VARCHAR(255)	Primary Key
MP3	SONG_TITLE	VARCHAR(255)	Foreign Key
ARTIST	ARTIST_NAME	VARCHAR(255)	Primary Key
ALBUM	ALBUM_ID	INT	Primary Key
ALBUM	ALBUM_TITLE	VARCHAR(255)	NOT NULL
ALBUM	TOTAL_TRACKS	INT	NOT NULL
ALBUM	IMAGE_LINK	TEXT	NOT NULL
ALBUM_BY_ARTIST	ARTIST_NAME	VARCHAR(255)	Foreign Key
ALBUM_BY_ARTIST	ALBUM_ID	INT	Foreign Key
GENRES	GENRE	VARCHAR(255)	Primary Key
GENRES	ALBUM_ID	INT	Primary Key
GENRES	SONG_TITLE	VARCHAR(255)	Primary Key
SONG	TITLE	VARCHAR(255)	Primary Key
SONG	ALBUM_ID	INT	Foreign Key
SONG	YEAR_RELEASED	INT	NOT NULL
SONG	TRACK_NUM	INT	
USER	SUBMISSION_ID	INT	Foreign Key
WEBCRAWLER	SUBMISSION_ID	INT	Foreign Key

Revised Application Program Design

```
submitMP3() //user has submitted an MP3
//checks for duplicate hashes in database
if (duplicates):
        //gather and display information for user, including a sum of all the songs submitted into the
database from that album
else:
        //switch to information gathering screen to get data from user

submitNewData(): //user has manually entered information to be stored on info gathering screen
//checks for duplicate album in database, adds new entry if it doesn't exist
//checks for duplicate artist, then adds entry if it doesn't already exist
//checks for duplicate song on album, adds entry if it doesn't exist
//inserts genre
//inserts mp3
//inserts into submission and user_submission
//switches to main screen

editSubmission()  //Ran if the user needs to edit an entry into the database
//pulls up current data from submission in question (found via MP3)
//deletes old entry, while also cleaning things up, like deleting an album with no songs or an artist with no
albums
//creates new entry, already having made sure the new entry wouldn't violate any key constraints before
deleting the old entry

deleteSubmission()  //Ran if the user needs to delete an entry from the database
//deletes old entry, while also cleaning things up, like deleting an album with no songs or an artist with no
albums


appendTags()  //Ran if the user wants to append the tags stored in the database to their MP3
//appends tags to the MP3 in question
```

Aggregation Functions

Counting Submitted Songs in an Album: Iterates through the table of songs for a particular albumID and counts the number of entries. Used to gauge the size of an album and is displayed to the user when they submit an MP3 that has already been registered in the database.

Get Max ID: Used to generate a new ID for both submissions and albums. Iterates through the list of submissions/albums and finds the highest ID value, then attribute the album/submission to be added this highest value plus one.

Installation Guide

1. Downloading DAFT and Installing Python:

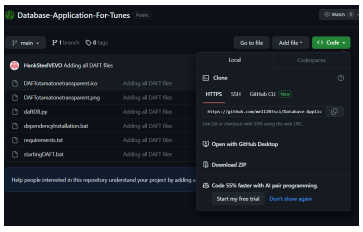
a. First, click the Code button and download the ZIP file from the GitHub page at: https://github.com/me11203sci/Database-Application-For-Tunes

b. If Python isn't already installed on your computer, the first step is to make that happen! Go to this link: https://www.python.org/downloads/release/python-3117/ and download the Windows installer for your system (Windows installer (64-bit) is recommended).

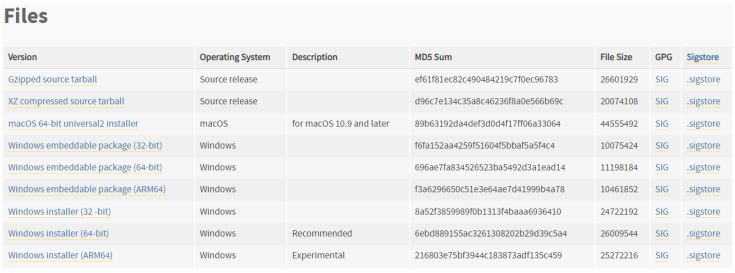
c. Open the installer and click "Install Now"

d. Python has been installed! Move on to the next step.

1a)



1b)

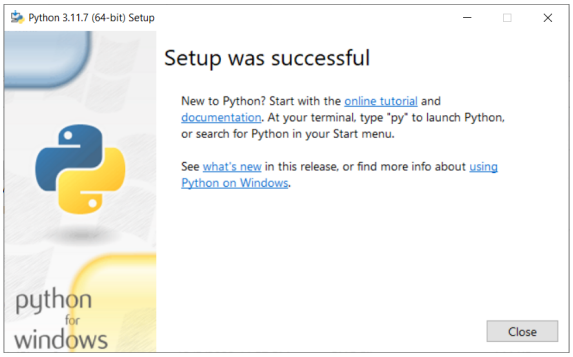


Files

Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore
Gzipped source tarball	Source release		ef61f81ec82c490484219c7f0ec96783	26601929	SIG	.sigstore
XZ compressed source tarball	Source release		d96c7e134c35a8c46236f8a0e566b69c	20074108	SIG	.sigstore
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	89b63192da4def3d0d4f17ff06a33064	44555492	SIG	.sigstore
Windows embeddable package (32-bit)	Windows		f6fa152aa4259f51604f5bbaf5a5f4c4	10075424	SIG	.sigstore
Windows embeddable package (64-bit)	Windows		696ae7fa834526523ba5492d3a1ead14	11198184	SIG	.sigstore
Windows embeddable package (ARM64)	Windows		f3a62966650c51e3e64ae7d41999b4a78	10461852	SIG	.sigstore
Windows installer (32-bit)	Windows		8a52f3859989f0b1313f4baaa6936410	24722192	SIG	.sigstore
Windows installer (64-bit)	Windows	Recommended	6ebd889155ac3261308202b29d39c5a4	26009544	SIG	.sigstore
Windows installer (ARM64)	Windows	Experimental	216803e75bf3944c183873adf135c459	25272216	SIG	.sigstore

1c)

1d)



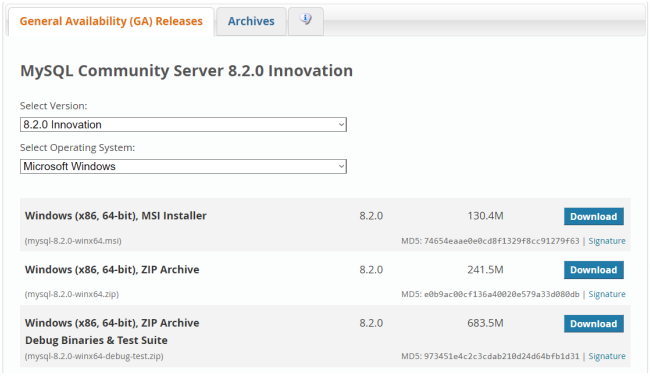
2. Downloading your MySQL database

a. First, we will be downloading a MySQL Community Server, do so at the following link via the MSI Installer: https://dev.mysql.com/downloads/mysql/

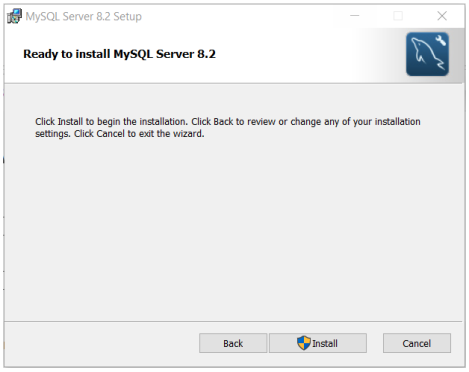
b. Open the installer, click next until the Choose Setup Type screen, then select Typical, then Install. Once the download is finished, it will open the MySQL Configurator.

c. Once the configurator is open, keep selecting next until the Accounts and Roles screen is available. It will ask for a root password, pick a password and remember it. Continue clicking next, until you are prompted with Execute, which you will also click. Click Next, and then Finish. Your database is almost ready to go!

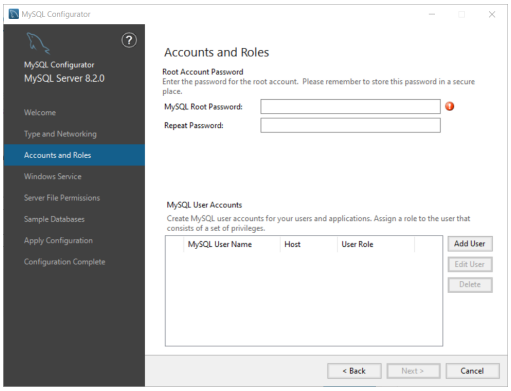
2a)



2b)

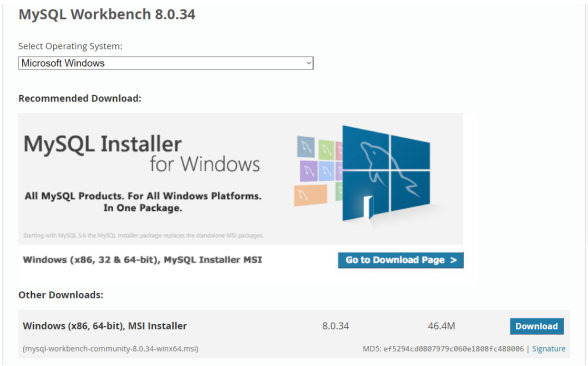


2c)

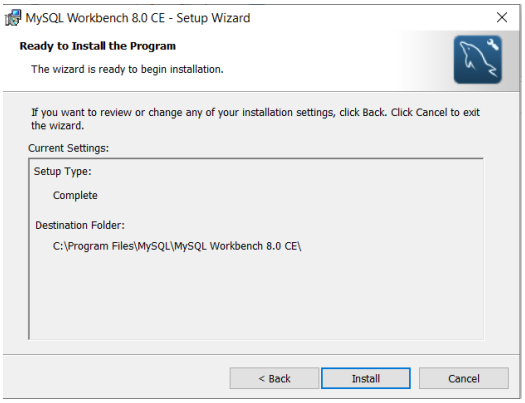


3. **Downloading MySQL Workbench**
    a. To download MySQL Workbench, go to the following link and download via the MSI Installer: https://dev.mysql.com/downloads/workbench/
    b. Open the MySQL Workbench Installer and click Next until you reach the Ready to Install the Program screen. Click Install. Once it is done, let it launch MySQL Workbench.
    c. Once you've opened MySQL Workbench, if there is not already a MySQL Connection, click Rescan servers, and one should appear. Click on it.
    d. It should prompt you for a password, and you should enter the password you created in Step 2c. If, after clicking on your server, it instead takes a while to load and you are taken to a screen that says "Server Status: Stopped", continue with step 3d, otherwise, move to Step 4.
        i. First, open the Task Manager and go to the Services tab. If you do not see the Services tab, select More Details at the bottom. Once you've done this, scroll down until you see "MySQL##". If it is stopped, right click it and select Start.
        ii. Once this is done, go back to your MySQL Workbench, select the home icon in the top left corner, and continue with Step 3c.

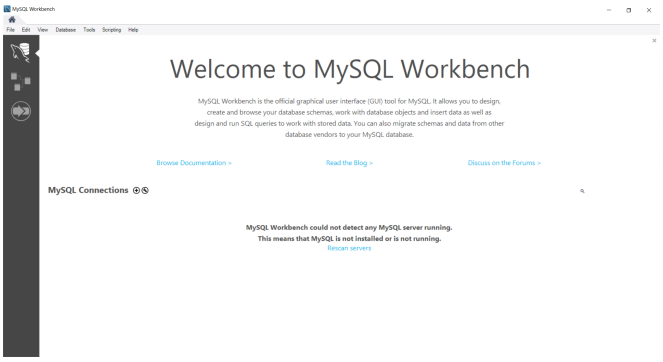
3a)



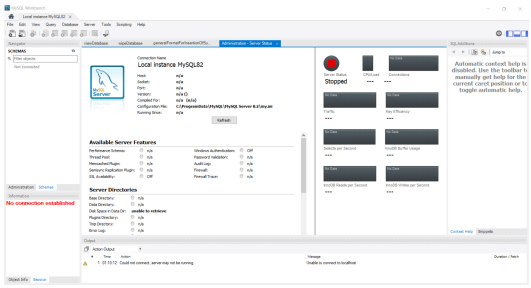
3b)



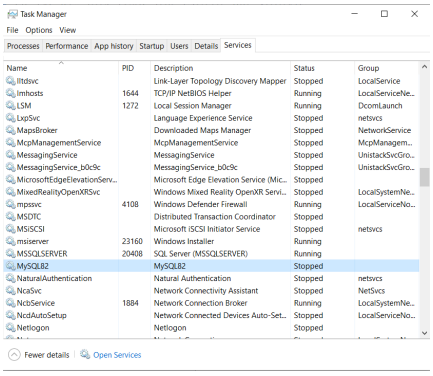
3c)



3d)



3di)



4. Loading the premade tables and entries

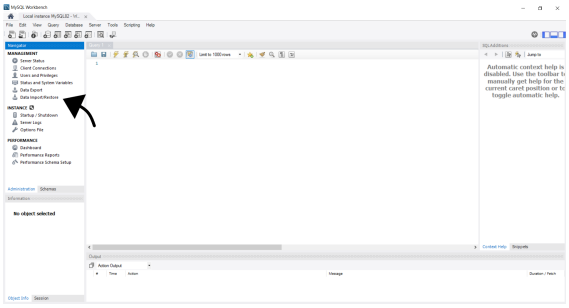
a. Now that you have MySQL Workbench open and connected to your local database, it might seem a bit empty, and that's because it is. In order to populate this database, we need to import the premade schema. To do this, first go to the Administration tab located around the middle left of the screen.

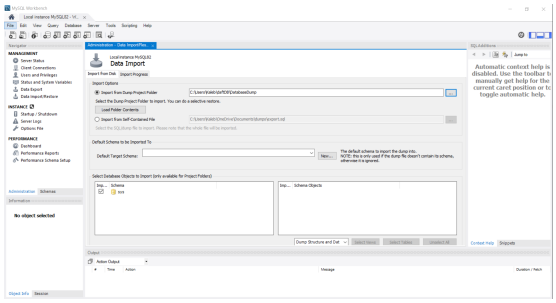
b. Once here, select the Data Import/Restore option from the Management section.

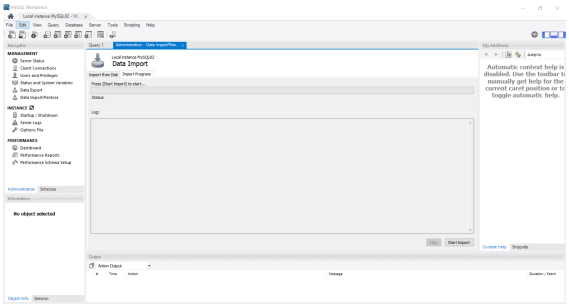
c. This screen may seem to be complicated, but we don't need to change much. The "Import from Dump Project Folder" option should already be selected, but we need to point it towards the right directory. Click the button labeled "…" at the end of that row, and browse for the folder labeled DatabaseDump that was downloaded at the beginning in the ZIP file.

d. Go to the Import Progress tab, and select Start Import from the bottom right corner. The tables and entries have been loaded!

4a) 

4b) 

4c) 

4d) 

5. Python Script Setup

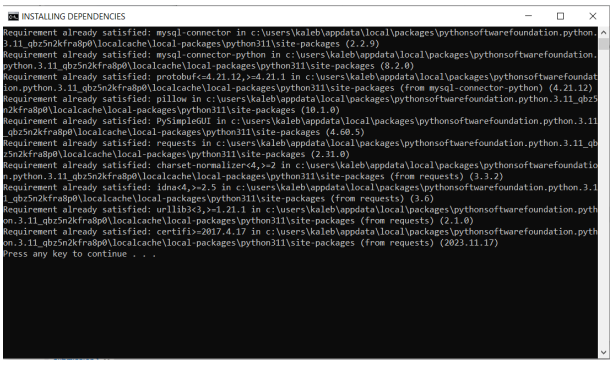
a. Alright, homestretch, this is the last step of installation. First, navigate to the ZIP file you downloaded and open “dbPassword.txt”. Once here, type in the password you created back in Step 2c. Then, click File, and Save.

b. Once that’s done, go back to the daftDB ZIP file and run “dependencyInstallation.bat”. This script installs all the necessary dependencies that the DBMS relies on.

c. Finally, run “startingDAFT.bat”. You’ll never guess what this one does. Congrats! You’ve successfully installed DAFT!

5a)

5b)



User Manual

Thank you for downloading DAFT! This will be a user guide on how to manipulate your database of MP3s with the software.

Guide

Introduction to DAFT

So, just what the heck does this software do anyways? DAFT stands for Database Application For Tunes, and it is a Database Management System that keeps track of all the tags on various songs so that if one were submitted to the software, it could have those tags appended to it. But what do I mean when I say tags? Tags, or more accurately, metadata, are pieces of information stored with an .mp3 file that provide information for it such as artist, album, title, and more. The de facto standard format for these tags among MP3s is called ID3, which stores the information about an mp3 as (usually) just a few bytes either at the beginning or the end of the file. This does not affect the audio content of an .mp3 file. DAFT supports the following ID3 tags: Title, Release Year, Genre, Artist, Album Track Number, and Album Art. So, if you've downloaded a song as an .mp3 file and aren't sure of the release year, genre, title or really any of those main tags for it, you can submit it to DAFT. If that mp3 is in our system, you can quickly and easily add those tags on your .mp3 file to match.

How to submit an MP3 to the database manually

In order to find the tags for an MP3, we need to submit it first. To do this, either type the file path to the .mp3 file in the input box provided, or select the browse button to find it through File Explorer. Once you've located your .mp3 file, select Submit. From here, there are two possibilities:

- **The MP3 is already in the database**
    - Great! The information has been displayed for you.

- **The MP3 is not yet in our database**
    - Unfortunately, we do not have this mp3 stored within the database yet. To help grow DAFT, please take the time to submit the information for the song to be stored if you would like to.

How to submit JSON files into the database

Get to this soon…

How to append the tags of an entry to your MP3

First, submit your mp3 by following the steps in the **How to submit an MP3 to the database manually** section. If the tags for your mp3 are not in the database, then we have no information for you to use. However, if your mp3 is already in the database, then all you need to do is click the “Append Tags To Submitted MP3” button, and your MP3 will have the metadata appended to it.

How to edit a submission

First, submit your mp3 by following the steps in the **How to submit an MP3 to the database manually** section. If the tags for your mp3 are not in the database, then there is no submission to edit. However, if your mp3 is already in the database, then click the “Edit Submission” button. The information that is currently attributed to that MP3 will be auto filled so as to make small changes to prior entries easier. Once you are satisfied with your changes, you can select “Submit Data”.

How to delete a submission

First, submit your mp3 by following the steps in the **How to submit an MP3 to the database manually** section. If the tags for your mp3 are not in the database, then there is no submission to delete. However, if your mp3 is already in the database, then all you need to do is click the “Delete Submission” button, and that submission will be deleted.