GRAD 695 Research Methodology & Writing

Applied Project Synopsis

Project Title: MUSICON, a music player emulator

Professor: Dr. Abrar Qureshi

Author: Prafulla Chandra Munugoti

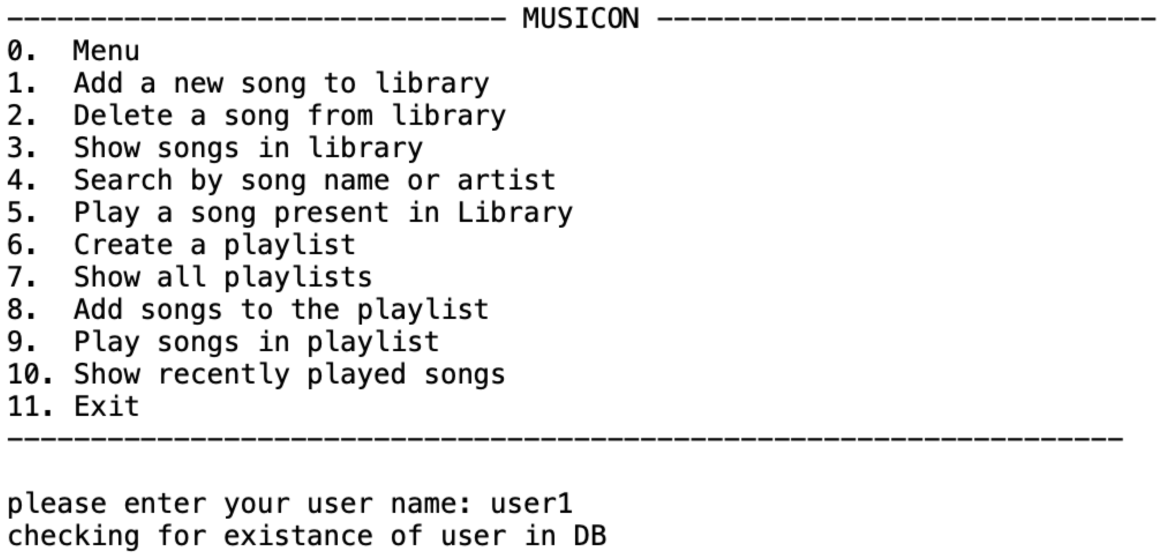
Term: Spring 2023

Submission Date: 04/09/2023

Project Synopsis:

MUSICON, a music player emulator is a software alternative to explore an option to stream audio files. It is primarily built in response to the role of music for stress-free, morale boost, and confidence-increasing purpose. Researchers tried to understand the role of music by conducting a survey with text responses for users engaged with music during the pandemic. The revelation of major correlations with participants on how their mood was regulated by their musical engagement, anxiety, and worry levels.

This survey results led me to build the MUSICON software artifact, allowing users to play, pause, resume, and skip audio files as they wish. MUSICON starts with the display of its functions via an infinite loop for the user's comfort and usability. The major functionalities cover letting users check their permissions supported by the backend MySQL database. It allows users to create their own account and super users provide relevant access and the features available will be decided based upon the permissions. Superusers are responsible for any sort of maintenance related to DB w.r.t adding, deleting, and updating users and songs as and when items are available to be put in DB. MUSICON allows users to group the songs of their liking and place them under one category named playlist. Each user can create 1 or many playlists. Users are allowed to play audio files in any order which can be the sequential, random, or looped way. These modes are applicable to single audio files or multiple audio files via creating a playlist. It can mine the DB present in the backend to well assist users with their choice made from the existing menu options. It is an interactive music player emulator that requests user interaction at each, and every choice made. MUSICON is well supported by a tightly connected, fully relational database in the back end. The interactive menu is printed below:



```
------------------------------ MUSICON ------------------------------
0.  Menu
1.  Add a new song to library
2.  Delete a song from library
3.  Show songs in library
4.  Search by song name or artist
5.  Play a song present in Library
6.  Create a playlist
7.  Show all playlists
8.  Add songs to the playlist
9.  Play songs in playlist
10. Show recently played songs
11. Exit
--------------------------------------------------------------------

please enter your user name: user1
checking for existance of user in DB
```

To develop such an interactive menu-based software artifact there are many requirements both from the software and hardware side to make it possible. Below are some of the requirements based on the machine that is used to develop the application:

Integrated Development Environment (IDE):

Anaconda Navigator 2.4.0 (Jupyter Notebook 6.3.0)

Hardware Requirements:

Processor: Intel Core i5 or Apple M1 or AMD Ryzen or equivalent
RAM: 4 GB RAM (8 GB preferred)
Hard Disk: 15 GB

Software Requirements:

Coding Platform: Developed in Mac OS Ventura 13.3.1 (portable to Linux and Windows machines)
Coding Language: Python
Compiler: Python 3.8.8 (MacOS comes pre-installed with Python version)
Libraries used: pandas, pymysql, copy, time, pygame, colorama, re, datetime, random.

Build/Installation Instructions:

1. MySQL Installation: https://dev.mysql.com/doc/mysql-installation-excerpt/8.0/en/macos-installation-pkg.html
2. Anaconda Installation: https://docs.anaconda.com/anaconda/install/mac-os/
3. Type cmd/terminal to pull a terminal and enter the command: “pip install pymysql” or “pip3 install pymysql".
4. Type cmd/terminal to pull a terminal and enter the command: “pip install pygame” or “pip3 install pygame".
5. Type cmd/terminal to pull a terminal and enter the command: “pip install pandas” or “pip3 install pandas".
6. Type cmd/terminal to pull a terminal and enter the command: “pip install colorama” or “pip3 install colorama".
7. All other libraries come standard with Python installation, else use the similar command pip3 install <library name>.

Following the above steps 1-7 will provide proper installation of libraries.

Starting MySQL server for the script to fetch data from DB:

1. https://dev.mysql.com/doc/refman/8.0/en/macos-installation-launchd.html
2. cd /usr/local/mysql/bin/
3. ./mysql -u root -p. → prompts a password of the root user

4. Enter the password of the root used in the installation process. MySQL is ready to use.
5. Run SQL commands present in “musicon.sql” file to store the data in MySQL DB.

Run the musicon code:

1. cd <directory where source code is located>
2. ./musicon.py, and follow the menu items to use the MUSICON software artifact.

Product Inventory (List of Files):

1. MUSICON.py (main Python source code)
2. musicon.sql (mail script to create/insert initial tables data)
3. songs/*.mp3 (audio files for the software artifact to work)
4. libraryTest.py (test cases of Library class)
5. playlisttest.py (test cases of Playlist class)
6. musicontest.py (test cases of MUSICON software artifact as a system)
7. README (file and explanation of each option in the MUSICON menu with step-by-step instructions for each file and option)

Known Issues:

1. No known issues found yet.