Group no: 3

Project title: Music Database with Bash Scripts

Team members:

1) Chan, Yik Hei Student ID: 301347095

2) Li, Che Shing Winson Student ID: 301330329

3) Nguyen, Pham Thai Vy Student ID: 301321378

4) Tsoi, Ho Ting Edmund Student ID: 301351619

5) Uzel, Sebastiyan Student ID: 301316200

Section 1: Synopsis

Part A: Scope

The project will be based on fetching music data from a free service provided in Rapid API, which will return a response in a JSON format. However, the free service only allows a maximum of 100 API calls per month. In order to fire API requests and manipulate the content of JSON response, linux packages must be updated in prior to launching the script.

```
sudo apt-get update
sudo apt-get install curl jq
```

The project requires various skills such as API configuration, data storage management, data structure design, and data diagnosing, which offers a great chance to enrich knowledge and acquire experience related to using linux system.

The scope of the project is described below:

1. Music API research and integration
    - Research and select a suitable music API provider
    - Evaluate the APIs based on their data coverage and ease of use
    - Obtain an API key and necessary credentials
    - Write a bash script to make API requests and fetch music metadata based on parameters

2. Search by calling API
    - Allow user to input the song title / arist / lyrics as a parameter to be passed to the API call
    - Parse the API response to extract relevant metadata and display in a readable format

- After receiving a successful response, allow user to choose to pick a song from the search result to add to the local music library (mentioned in point 4 below)

3. Set up custom music library
   - Design a way and develop a script to store the important metadata from the API response if user decides to add the song to the library
   - Pair up the response data of that song to the database file
   - Allow user to view the locally saved music entries

4. Additional local custom music library features
   - Add sorting functionality to sort locally saved songs by title, artist, or release year

5. User interface (menu) and error handling
   - Develop a simple command-line user interface to interact with the music database
   - Design intuitive menu options for adding songs, searching, sorting, and accessing additional features

6. API response caching
   - Cache the response received as API usage is limited
   - Easier for development purpose, as debugging can be done on saved API responses

7. User input validation

Part B: Algorithm

- Preparation

Research a suitable API and study its documentation

Diagnose the API JSON response and pick necessary fields

Research on packages needed to break down JSON in linux environment

Research on sending API requests in linux environment

- Implementation

Accept user to input a search string to search songs

Include the search string into the API URL and fire a request

Receive the response and save it to a cache for easier development and analysis

- Saved to ./cache directory

Access the cache to display the chosen content from the search response on screen

- Only the title, artist, and full release date is displayed on screen
- Title, song ID, artist, artist ID, and normalized date string is secretly saved to a buffer, with the designed data format

Accept user input to add songs from search to local music library, or to start another search

- Invalid options are blocked
- Add songs: validate input range 1-10 inclusive
- The designated song is taken from the buffer and added to the end of the library file

Display all song entries in the local music library database file

- **Important: There is a prepared sample "music_library.db" file solely for demonstration purpose. Please find the sample music library file in the attachment.**
- The file is created automatically if it does not already exists when the application starts
- Read all rows in the file and use IFS="|" to separate the fields
- Manipulate the full release date string (YYYYMMDD) to display only the year
- Use string formatting to display the entries in a tidy manner

Accept user input to remove a song entry from the music library

- Read the total number of rows in the file and determine the input range
- Invalid options are blocked
- Remove the designated row

Accept user input to sort all song entries by title, artist, or release year

- Invalid options are blocked

Display the main menu

- Invalid options are blocked

- Add-ons

Implement development mode as a “feature switch” in order to save API usage during development and testing

- **Important: If development mode is active, it will not fire API request and will use the file “response.cache” saved under the “./cache” directory. Make sure the cache file is placed correctly. Please find the sample cache file in the attachment.**

Toggle development mode in the main menu

- Finalizing

Put all the modules together and refactor common displays to a function for consistency
Perform testing and debugging, put “clear” command appropriately for better visuals

Part C: Project Enhancements

1. Currently, the same song can be added to the library repeatedly. This issue can be resolved by checking the unique song ID (which is already included in the music library when designing the structure). The feature is not implemented because of time constraints.

2. At the time of adding a song after performing a live search, the user input is not totally validated. It is currently accepting any values from 1 to 10 (inclusive). If user inputs a decimal value that is in the range (e.g. 8.9, or 6.4), there will be an error.

3. In the music library, feature of searching for lyrics can be implemented. It will involve getting the respective unique song ID to fire another API request. The feature is not implemented because of time constraints again.

4. In the music library, feature of arranging song entries into playlists can be implemented.

Section 2: Bash Scripts

Documentation and version history

```
#!/bin/bash

# main.sh
# must be executed with bash main.sh
# must install the following packages first
# sudo apt-get update
# sudo apt-get install curl jq

# version 1.6
# v1.0: implemented json data extraction, data buffer, and release
date normalization
# v1.1: implemented option menu after search, allow performing
search again
# v1.2: implemented add song to local music database
# v1.3: implemented display of all songs in music_library.db
# v1.4: implemented delete and all sorting functions for
music_library.db
#       refactored line separator and invalid option error message
into functions
# v1.5: implemented main menu and fixed all visual bugs

# optional param, for details, see documentation
# https://rapidapi.com/Glavier/api/genius-song-lyrics1
```

Constant declaration (API and file directory location)

```
# toggle development mode, to save API call quota
DEV_MODE=true

# Genius API info
API_BASE_URL="https://genius-song-lyrics1.p.rapidapi.com/search/?"
API_HOST="genius-song-lyrics1.p.rapidapi.com"
API_KEY="097f8f91ccmsh1ab75a582424b34p110966jsn07404ed0c4e6"

# define cache directory, cache file and local music library
CACHE_DIR="./cache"
```

```
CACHE_FILE="$CACHE_DIR/response.cache"
MUSIC_LIBRARY="music_library.db"
# set the data buffer file for json extraction
DATA_BUFFER="buffer.txt"

# create cache directory if it does not exist
mkdir -p "$CACHE_DIR"

# create the local music library if it does not exist
if [ ! -f "$MUSIC_LIBRARY" ]; then
  touch "$MUSIC_LIBRARY"
fi
```

```
Function 1: Search a song
Take user input
If development mode is ON, no API request is fired
If development mode is OFF, construct the correct URL and fire API request
Save the response to the designated path as a cache
```

```
search_from_api() {
  # prompt user to enter the search string
  read -p "Enter the search string, it can be song title, artist name, or lyrics : " search_string

  # check if development mode is active, skip api call if true
  if [[ $DEV_MODE = true ]]; then
    echo "Development mode is active. Skipping API call. Search string has no effect."
  else
    # construct the complete api url with the search string
    API_URL="${API_BASE_URL}q=${search_string}&per_page=10&page=1"

    curl --request GET \
      --url "$API_URL" \
      --header "X-RapidAPI-Host: $API_HOST" \
      --header "X-RapidAPI-Key: $API_KEY" \
      --output "$CACHE_DIR/response.cache"
  fi
```

```
}
```

```
Function 1: Search a song
Extract required fields from API response
Data normalization
Save data to a buffer for further use
Display search results
Allow user to “add song”, “perform another search”, or “exit”
```

```
extract_from_json() {
  # clear the data buffer before extraction
  > "$DATA_BUFFER"

  # read json response from file
  json_response=$(cat $CACHE_FILE)

  # extract the number of songs
  num_songs=$(echo "$json_response" | jq '.hits | length')
  echo "$num_songs song(s) found with the query"
  echo ""

  # iterate over each song
  for ((i=0; i<num_songs; i++)); do
    echo "Song $((i+1)) :"

    # extract the fields from json for the current song
    title=$(echo "$json_response" | jq -r ".hits[$i].result.title")
    song_id=$(echo "$json_response" | jq -r ".hits[$i].result.id")
    artist=$(echo "$json_response" | jq -r
".hits[$i].result.primary_artist.name")
    artist_id=$(echo "$json_response" | jq -r
".hits[$i].result.primary_artist.id")
    full_release_date=$(echo "$json_response" | jq -r
".hits[$i].result.release_date_for_display")
    release_year=$(echo "$json_response" | jq -r
".hits[$i].result.release_date_components.year")
    release_month=$(echo "$json_response" | jq -r
".hits[$i].result.release_date_components.month")
```

```
    release_day=$(echo "$json_response" | jq -r
".hits[$i].result.release_date_components.day")

    # format the release date before saving to local db
    # check if any of the release date components is null first
    if [[ "$release_year" == "null" ]]; then
      release_year="0000"
    fi
    if [[ "$release_month" == "null" ]]; then
      release_month="00"
    fi
    if [[ "$release_day" == "null" ]]; then
      release_day="00"
    fi
    # format the release date into YYYYMMDD format
    normalized_release_date=$(printf "%04d%02d%02d" "$release_year"
"$release_month" "$release_day")

    # print the extracted fields for the song
    echo "Title: $title  |  Artist: $artist  |  Release Date:
$full_release_date"

    # append chosen fields to the buffer
    echo
"$title|$song_id|$artist|$artist_id|$normalized_release_date" >>
"$DATA_BUFFER"
  done

  print_line_separator
  option=""
  while [[ $option != [Qq] ]]; do
    read -p "<A> add a song to library <S> perform another search
<Q> exit to main menu : " option
    case $option in
      [Aa])
        add_song_to_library
        ;;
      [Ss])
```

```
        clear
        search_from_api
        extract_from_json
        ;;
      [Qq])
        clear
        # go back to main menu
        break
        ;;
      *)
        print_invalid_option_message
        ;;
    esac
  done
}
```

Function 1: Search a song
Add a song from search result to local music library

```
add_song_to_library() {
  while true; do
    read -p "Enter the song number (1-10) to add : " desired_song
    if [ $desired_song -gt 0 ] && [ $desired_song -le 10 ]; then
      song=$(sed -n "$desired_song"p $DATA_BUFFER)
      echo $song >> $MUSIC_LIBRARY
      echo "Song $desired_song is added to the music library."
      break
    else
      print_invalid_option_message
    fi
  done
}
```

Function 2: View local music library
Display all entries from the library file
Allow user to “remove song”, “perform sorting”, or “exit”

```
browse_music_library() {
  # read content from music library file
  music_library_file=$(cat "$MUSIC_LIBRARY")
```

```
  # display header row with predefined widths
  # use %-(n)s flags to fix the width of each fields, left-aligned
text
  printf "%-5s | Title%-75s | Artist%-34s | Release Year%s\n"
  print_line_separator

  # initialize the list counter
  counter=0

  # loop the entries
  while IFS="|" read -r song_title song_id artist_name artist_id
release_date; do
    ((counter++))
    printf "%-5s | %-80s | %-40s | %s\n" "$counter" "$song_title"
"$artist_name" "${release_date:0:4}"
  done <<< "$music_library_file"

  print_line_separator
  option=""
  while [[ $option != [Qq] ]]; do
    read -p "<R> remove a song from the library <S> sort the
entries <Q> exit to main menu : " option
    case $option in
      [Rr])
        remove_song
        ;;
      [Ss])
        sort_song
        ;;
      [Qq])
        clear
        # go back to main menu
        break
        ;;
      *)
        print_invalid_option_message
        ;;
```

```
    esac
  done
}
```

Function 2: View local music library
Remove a song entry from the music library
Locate the correct row to be removed and update the library file
Display the updated library

```
remove_song() {
  # split the music library content into an array of rows by line
break (\n)
  IFS=$'\n' read -rd '' -a rows <<< "$music_library_file"

  local remove_option_valid=false
  while [[ $remove_option_valid = false ]]; do
    read -p "Enter the song number to remove : " remove_song_number

    # validate user input
    if ((remove_song_number < 1 || remove_song_number >
${#rows[@]})); then
      print_invalid_option_message
    else
      remove_option_valid=true
    fi
  done

  # remove the selected row from the array
  unset "rows[remove_song_number-1]"

  # join the remaining rows back into a single string
  local updated_music_library=$(printf "%s\n" "${rows[@]}")

  # overwrite the music library file with the updated content
  echo "$updated_music_library" > "$MUSIC_LIBRARY"

  # reprint the music library after the remove confirmation
message
  clear
```

```
  echo "Song entry has been removed from the music library
successfully."
  echo ""
  browse_music_library
}
```

Function 2: View local music library
Allow user to sort the entries by title, artist, or release year
Sort by the chosen field and update the library file
Display the updated library

```
sort_song() {
  local sort_column=0

  local sort_option_valid=false
  while [[ $sort_option_valid = false ]]; do
    read -p "Sort music library by <T> title <A> artist <Y> release
year : " sort_song_option
    case $sort_song_option in
      [Tt])
        sort_column=1
        sort_option_valid=true
        ;;
      [Aa])
        sort_column=3
        sort_option_valid=true
        ;;
      [Yy])
        sort_column=5
        sort_option_valid=true
        ;;
      *)
        print_invalid_option_message
        ;;
    esac
  done

  # sort the music library by the specified column
```

```
  local sorted_music_library=$(echo "$music_library_file" | sort -
t "|" -k "$sort_column")

  # overwrite the music library file with the sorted content
  echo "$sorted_music_library" > "$MUSIC_LIBRARY"

  # reprint the music library after the soft confirmation message
  clear
  echo "Song entry has been sorted successfully."
  echo ""
  browse_music_library
}
```

```
Function 3: Toggle development mode
Allow convenient toggling of development mode within application
By default, development mode is ON when the program starts
```

```
toggle_development_mode() {
  if [ "$DEV_MODE" = true ]; then
    DEV_MODE=false
  else
    DEV_MODE=true
  fi
}
```

```
Common functions for display consistency
```

```
print_line_separator() {
  # 150 units
  echo "-----------------------------------------------------------
------------------------------------------------------------------
--------------------------"
}

print_invalid_option_message() {
  echo "Invalid choice. Please check your input and try again."
}
```

```
Display of main menu
```

```
print_main_menu() {
```

```
  echo "==================== Main Menu ====================="
  echo "1 - Search a song"
  echo "2 - View my local music library"
  echo "3 - Toggle development mode (currently : $DEV_MODE)"
  echo "Q - Exit the application"
  echo "====================================================="
}
```

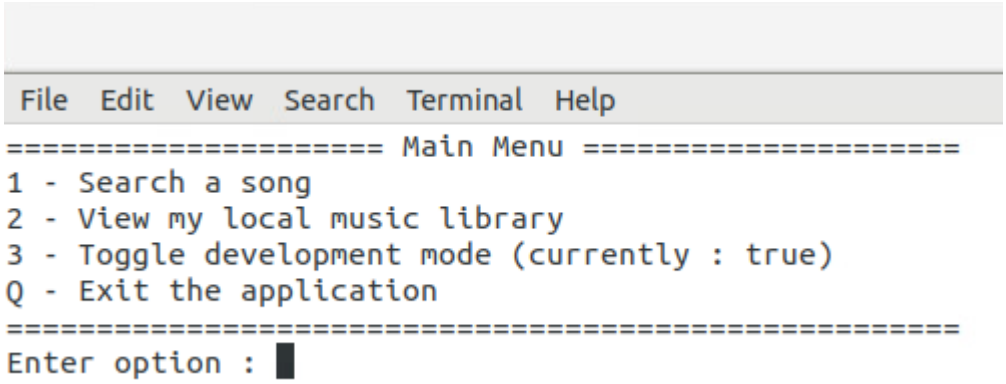
Main program

```
# main program starts
clear
main_option=""
while [[ $main_option != [Qq] ]]; do
  print_main_menu

  read -p "Enter your choice: " main_option
  case $main_option in
  1)
    clear
    search_from_api
    extract_from_json
    ;;
  2)
    clear
    browse_music_library
    ;;
  3)
    toggle_development_mode
    clear
    ;;
  [Qq])
    echo "See you again!"
    exit
    ;;
  *)
    clear
    print_invalid_option_message
    ;;
```

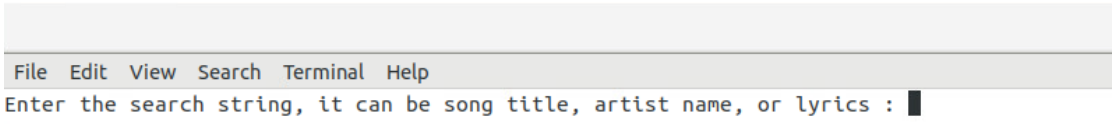
```
  esac
done
```

Section 3: Outputs Received

Unless otherwise stated, both lower-case or upper-case letters are accepted in all options.

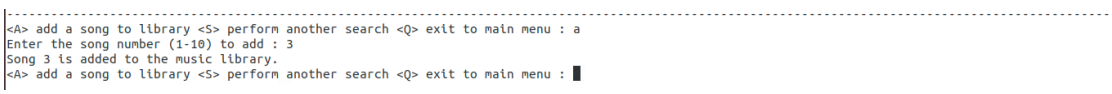


Option 1: Search a song



User is prompted to input the search string. The string can include the song title, artist name, or lyrics.

Under development mode, API request will not be fired and the search string has no effect. A message is displayed to inform user. The search result is taken from the cache file.

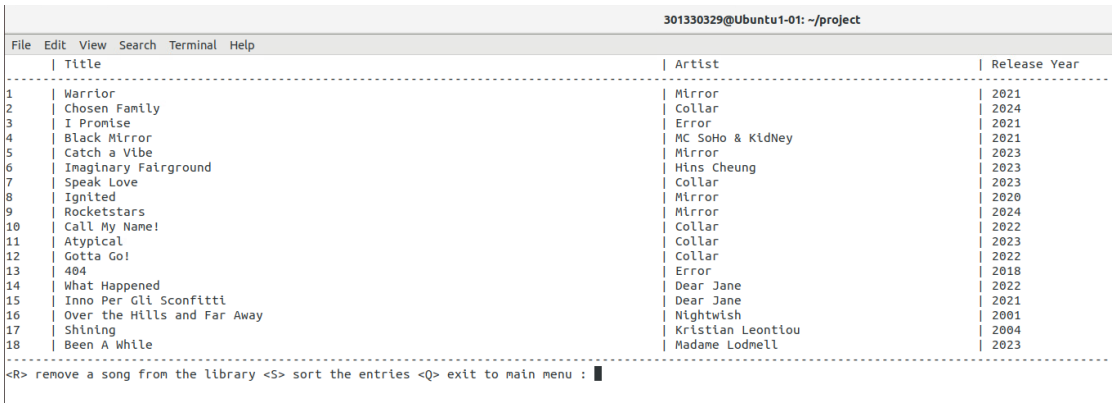


If user decides to add a song to the local library, press A and then input the song number to complete the action. The song will be added to the library file. As an illustration, song number 3 is added.

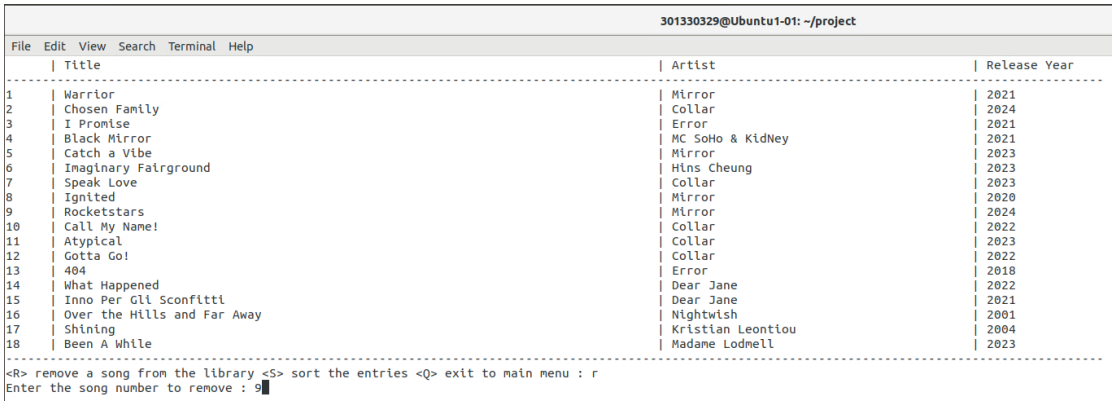
If user decides to perform another search, press S and then the screen will be cleared. User will be prompted to input the search string again.

Press Q to go back to the main menu.

Option 2: View local music library



It displays all the entries in a tidy manner from the library file. The song that is added above as an example can be seen in the music library.



If user wants to remove a song from the library, press R and then input the row number to complete the action. As an illustration, user is going to remove entry number 9 (i.e. Rocketstars by Mirror).

The chosen entry is removed from the music library. The screen is cleared and a confirmation message is displayed. The content of the library is printed again.

```
301330329@Ubuntu1-01: ~/project
File  Edit  View  Search  Terminal  Help
      | Title                                                     | Artist                         | Release Year
-----------------------------------------------------------------------------------------------------------------
1     | Warrior                                                   | Mirror                         | 2021
2     | Chosen Family                                             | Collar                         | 2024
3     | I Promise                                                 | Error                          | 2021
4     | Black Mirror                                              | MC SoHo & KidNey               | 2021
5     | Catch a Vibe                                              | Mirror                         | 2023
6     | Imaginary Fairground                                      | Hins Cheung                    | 2023
7     | Speak Love                                                | Collar                         | 2023
8     | Ignited                                                   | Mirror                         | 2020
9     | Call My Name!                                             | Collar                         | 2022
10    | Atypical                                                  | Collar                         | 2023
11    | Gotta Go!                                                 | Collar                         | 2022
12    | 404                                                       | Error                          | 2018
13    | What Happened                                             | Dear Jane                      | 2022
14    | Inno Per Gli Sconfitti                                    | Dear Jane                      | 2021
15    | Over the Hills and Far Away                               | Nightwish                      | 2001
16    | Shining                                                   | Kristian Leontiou              | 2004
17    | Been A While                                              | Madame Lodmell                 | 2023
-----------------------------------------------------------------------------------------------------------------
<R> remove a song from the library <S> sort the entries <Q> exit to main menu : s
Sort music library by <T> title <A> artist <Y> release year :
```

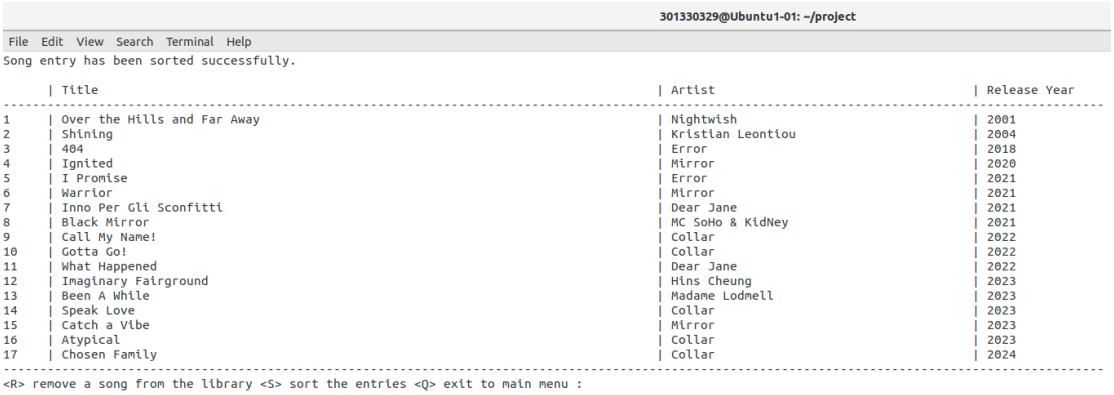
If user wants to sort the entries in the library, press S and user will be prompted the criteria for sorting. Currently the system offers 3 sorting method: by title, artist, and release year.

```
301330329@Ubuntu1-01: ~/project
File  Edit  View  Search  Terminal  Help
Song entry has been sorted successfully.

      | Title                                                     | Artist                         | Release Year
-----------------------------------------------------------------------------------------------------------------
1     | 404                                                       | Error                          | 2018
2     | Atypical                                                  | Collar                         | 2023
3     | Been A While                                              | Madame Lodmell                 | 2023
4     | Black Mirror                                              | MC SoHo & KidNey               | 2021
5     | Call My Name!                                             | Collar                         | 2022
6     | Catch a Vibe                                              | Mirror                         | 2023
7     | Chosen Family                                             | Collar                         | 2024
8     | Gotta Go!                                                 | Collar                         | 2022
9     | Ignited                                                   | Mirror                         | 2020
10    | Imaginary Fairground                                      | Hins Cheung                    | 2023
11    | Inno Per Gli Sconfitti                                    | Dear Jane                      | 2021
12    | I Promise                                                 | Error                          | 2021
13    | Over the Hills and Far Away                               | Nightwish                      | 2001
14    | Shining                                                   | Kristian Leontiou              | 2004
15    | Speak Love                                                | Collar                         | 2023
16    | Warrior                                                   | Mirror                         | 2021
17    | What Happened                                             | Dear Jane                      | 2022
-----------------------------------------------------------------------------------------------------------------
<R> remove a song from the library <S> sort the entries <Q> exit to main menu :
```

Sort by title

```
301330329@Ubuntu1-01: ~/project
File  Edit  View  Search  Terminal  Help
Song entry has been sorted successfully.

      | Title                                                     | Artist                         | Release Year
-----------------------------------------------------------------------------------------------------------------
1     | Chosen Family                                             | Collar                         | 2024
2     | Speak Love                                                | Collar                         | 2023
3     | Call My Name!                                             | Collar                         | 2022
4     | Atypical                                                  | Collar                         | 2023
5     | Gotta Go!                                                 | Collar                         | 2022
6     | What Happened                                             | Dear Jane                      | 2022
7     | Inno Per Gli Sconfitti                                    | Dear Jane                      | 2021
8     | I Promise                                                 | Error                          | 2021
9     | 404                                                       | Error                          | 2018
10    | Imaginary Fairground                                      | Hins Cheung                    | 2023
11    | Shining                                                   | Kristian Leontiou              | 2004
12    | Been A While                                              | Madame Lodmell                 | 2023
13    | Black Mirror                                              | MC SoHo & KidNey               | 2021
14    | Warrior                                                   | Mirror                         | 2021
15    | Catch a Vibe                                              | Mirror                         | 2023
16    | Ignited                                                   | Mirror                         | 2020
17    | Over the Hills and Far Away                               | Nightwish                      | 2001
-----------------------------------------------------------------------------------------------------------------
<R> remove a song from the library <S> sort the entries <Q> exit to main menu :
```

Sort by artist



Sort by release year

Press Q to go back to the main menu.

Option 3: Toggle development mode

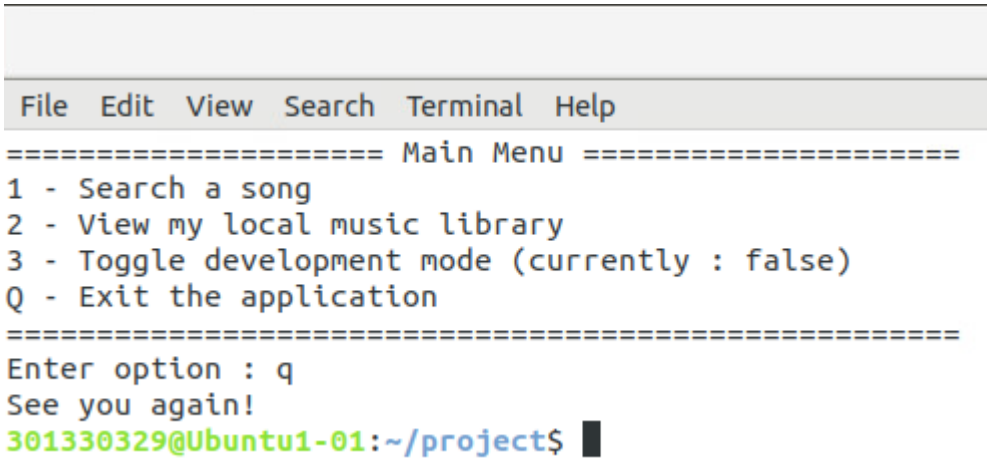
```
File  Edit  View  Search  Terminal  Help
====================== Main Menu ======================
1 - Search a song
2 - View my local music library
3 - Toggle development mode (currently : true)
Q - Exit the application
=======================================================
Enter option : 3
```

By default, the variable DEV_MODE is set to true when the application starts. The development mode is used to preserve API usage as the free API service only allows a maximum of 100 API calls per month. As mentioned in the previous section, there will be no real API calls in the song search function if the value of DEV_MODE is true. The status of the variable is reflected at the option 3 in the main menu. User (or mostly, developers) choose option 3 to toggle the value of DEV_MODE.

```
File  Edit  View  Search  Terminal  Help
====================== Main Menu ======================
1 - Search a song
2 - View my local music library
3 - Toggle development mode (currently : false)
Q - Exit the application
=======================================================
Enter option :
```

When option 3 is chosen, the value of DEV_MODE is flipped.

Option 4 (Q): Exit the application



User simply submit Q as the option to leave the application.