# User Guide Documentation

**Goal of the Program**

The **Songs** program is designed to help users manage a personal database of songs. It allows you to store, modify, and retrieve detailed information about songs, including title, performer, album, release year, genre, and length. The program provides functionalities to add new songs, edit existing entries, delete songs, and display songs based on various criteria.

**Instructions on How to Use the Program**

**Starting the Program**

1. **Launch the Program**: Run the executable file named songs.exe from your command line or by double-clicking it in your file explorer.

**Main Menu Options**

Upon starting, you'll see the main menu:

1. **Load Database**
2. **Add New Song**
3. **Delete Song**
4. **Edit Song**
5. **Save Database**
6. **Display Songs by Artist**
7. **Display Songs by Album**
8. **List Songs by Year**
9. **List Songs by Genre**
10. **Exit**

**Option Descriptions**

**1. Load Database**

- **Purpose**: Loads an existing database from a file.
- **Instructions**:

- Enter the filename when prompted.
- If the file exists, the database will be loaded into memory.
- If the file doesn't exist, a new database will be created in memory.

**2. Add New Song**

- **Purpose**: Adds a new song to the database.
- **Instructions**:
  - Input the song's title, performer, album, release year, genre, and length (minutes and seconds) when prompted.
  - The new song will be added to the database.

**3. Delete Song**

- **Purpose**: Removes a song from the database.
- **Instructions**:
  - Enter the title of the song you wish to delete.
  - The program will remove the song if it exists.

**4. Edit Song**

- **Purpose**: Modifies details of an existing song.
- **Instructions**:
  - Enter the title of the song you want to edit.
  - Update the song's details as prompted.

**5. Save Database**

- **Purpose**: Saves the current database to a file.
- **Instructions**:
  - Enter the filename where you want to save the database.
  - The database will be written to the specified file.

**6. Display Songs by Artist**

- **Purpose**: Shows all songs by a specific artist.

- **Instructions**:
    - Enter the artist's name.
    - The program will display all songs performed by that artist.

**7. Display Songs by Album**

- **Purpose**: Shows all songs from a specific album by an artist.
- **Instructions**:
    - Enter the artist's name.
    - Enter the album title.
    - The program will display all songs from that album.

**8. List Songs by Year**

- **Purpose**: Lists all songs released in a particular year.
- **Instructions**:
    - Enter the release year.
    - The program will display all songs from that year.

**9. List Songs by Genre**

- **Purpose**: Lists all songs of a specific genre.
- **Instructions**:
    - Enter the genre (e.g., rock, pop, jazz).
    - The program will display all songs matching that genre.

**10. Exit**

- **Purpose**: Exits the program.
- **Instructions**:
    - Ensure you've saved your database before exiting to avoid losing any changes.

**Where the Files are Saved and Loaded**

- **Default Directory**: By default, the program saves and loads files from the directory where the executable is located.

- **Custom Paths**: You can specify a full path when entering filenames to save or load databases from different directories.
- **File Format**: The database is saved in a text file format, which can be named as desired (e.g., songs_db.txt).

**Detailed Specification of the Problem**

**Program Objective**

To create a user-friendly application that allows individuals to maintain and manage a personal database of songs with detailed information. The application should provide functionalities that enable users to:

- Efficiently store and retrieve song data.
- Modify the database by adding, editing, or deleting entries.
- Query the database based on specific criteria like artist, album, year, or genre.
- Persist data between sessions through file handling.

**Expected Behaviour and Output**

- **Loading Database**:
    - If a file exists, the program should correctly read and load all song entries into memory.
    - If the file doesn't exist, the program should initialize an empty database.
- **Adding New Songs**:
    - The program should accept all required song details from the user.
    - The new song should be immediately available in the database for any subsequent operations.
- **Deleting Songs**:
    - Upon entering a valid song title, the program should remove the song from the database.
    - If the song doesn't exist, the program should inform the user accordingly.
- **Editing Songs**:
    - The program should allow the user to update any of the song's details.

    - Changes should reflect immediately in the database.
- **Saving Database**:
    - The program should write all current song entries to the specified file.
    - Data should be stored in a format that can be accurately reloaded in future sessions.
- **Displaying and Listing Songs**:
    - The program should display song information clearly and accurately based on the user's query.
    - For artist and album searches, the program should handle partial matches if implemented.
    - For year and genre listings, all matching songs should be displayed.

**User Input Handling**

- The program should validate user inputs where applicable (e.g., ensuring the release year is a valid number).
- In case of invalid inputs, the program should prompt the user to re-enter the information.
- The user interface should be intuitive, guiding users through each step with clear instructions.

**Data Persistence**

- All changes made during a session should be saved to a file when the user chooses to save the database.
- The program should handle file read/write errors gracefully, informing the user if an operation fails.