Programming Project / C++ - Playlist Management System

Student 1: Tănase Vlad Mihai
Student 2: Pușa Gabriel Valer

I. Task Description

Student 1 (Tănase Vlad Mihai) was responsible for building the playlist management system. This included designing and implementing the core functionality, such as:

- Creating and deleting playlists.
- Adding and removing songs from playlists.
- Viewing playlists sorted by rating (highest to lowest).
- Viewing songs within a playlist sorted alphabetically by title.
- Implementing file input/output to persist playlist data.
- Developing a menu-driven interface with color-coded output for user interaction.

Student 2 (Pușa Gabriel Valer) was responsible for testing and debugging the system. This involved:

- Adding, deleting, and modifying playlists to ensure functionality.
- Adding and removing songs to test playlist management features.
- Navigating through main and submenu options to verify usability.
- Debugging issues related to input validation, menu navigation, and file operations.

II. Data Structures Used by the Team

The following classes were used:

- **Song**: Contains:
  - `title` (string): The title of the song.
  - `artist` (string): The artist of the song.
- **Playlist**: Contains:
  - `name` (string): The name of the playlist.
  - `rating` (float): The rating of the playlist (0.0 to 5.0).
  - `songs` (vector of `Song`): A collection of songs in the playlist.
- **PlaylistManager**: Contains:
  - `playlists` (vector of `Playlist`): A collection of all playlists.

  - Methods to manage playlists, including loading/saving to file, creating, deleting, and managing playlists and songs.

III. File Structure

The following file is used:

- `file.txt`: Stores playlist data in the following format:

```
#<playlist_name>,<rating>
<song_title>,<artist>
<song_title>,<artist>
...
<empty line>
#<next_playlist_name>,<rating>
...
```

IV. Interacting with the Executable

The application (`Playlist_Management_System`) offers a menu-driven console interface with the following functionality:

Main Menu

Displayed with a light orange header "Playlists main menu" and the following options:

- **Create playlist**: Prompts for a playlist name (max 99 characters) and rating (0.0 to 5.0), then adds the playlist.
- **View all playlists**: Displays all playlists sorted by rating (highest to lowest) with a "Press Enter to continue..." prompt, then refreshes to the main menu.
- **Delete playlist**: Displays playlists sorted by rating, prompts for a number to select, and confirms deletion (y/n).
- **Manage playlist**: Displays playlists sorted by rating, prompts for a number to select, then enters the submenu.
- **Exit**: Saves playlists to `file.txt` and exits.

Songs Submenu

Displayed with a light orange header "Songs submenu" after selecting a playlist, with the following options:

- **Add song**: Prompts for song title and artist, then adds the song to the playlist.
- **View songs**: Displays songs sorted alphabetically by title with a "Press Enter to continue..." prompt, then refreshes to the submenu.
- **Delete song**: Displays songs sorted alphabetically, prompts for a number to select, and removes the song.
- **Back to main menu**: Returns to the main menu, refreshing the terminal.

The interface uses color-coded output (red for errors, green for success, yellow for ratings, cyan for headers, orange for menu titles) and refreshes the terminal when switching between menus or after viewing lists.