### Synopsis for Music Streaming Platform Mini Project

#### 1. Problem Statement:

In the era of digital entertainment, music streaming platforms have become the primary means for users to access and enjoy music content. However, managing a vast collection of music, including songs, artists, playlists, and user histories, poses a challenge for both users and service providers. Users need an efficient and user-friendly platform to manage their music collections, track play history, create personalized playlists, and discover new music. Music streaming services often struggle with data organization, scalability, and integration of user preferences.

This project aims to create a user-friendly Music Streaming Platform that enables seamless management of users, artists, songs, playlists, and play history, addressing the need for a simple but scalable interface for both users and administrators.

#### 2. Objectives:

The primary objectives of the Music Streaming Platform mini project are:

* To provide a system that allows users (listeners) to register, create, and manage playlists.
* To enable users to add songs to playlists, track their listening history, and explore different music genres or artists.
* To implement CRUD (Create, Read, Update, Delete) functionalities for managing data related to listeners, songs, artists, playlists, and play history.
* To design a robust backend system that ensures data integrity, secure user management, and scalability.
* To create an easy-to-use and responsive front-end interface, allowing users to interact with the platform efficiently.
* To allow administrators to manage the platform, ensuring smooth operation by adding, updating, and deleting records.

#### 3. Functional Requirements:

The system should meet the following functional requirements:

* **User Management:**
  + Listeners should be able to register with their names and email addresses.
  + Listeners should have the ability to update or delete their profiles.
* **Artist Management:**
  + Artists should be able to be added to the platform with relevant information such as artist name.
  + Administrators should be able to update or delete artist records.
* **Song Management:**
  + Songs should be linked to artists with the ability to display song details.
  + Administrators should be able to add, update, and remove songs.
* **Playlist Management:**
  + Listeners should be able to create personalized playlists.
  + Listeners can add or remove songs from their playlists.
  + Listeners should be able to update and delete playlists.
* **Play History:**
  + The platform should track the songs a listener plays, including the time/date of each play.
  + The play history should be accessible to listeners and allow for easy tracking of songs played over time.
* **Search Functionality:**
  + Listeners should be able to search for songs and artists within the platform.
  + Search results should display relevant information like song name, artist name, and playlist association.
* **Database:**
  + The platform will use a relational database to store and manage data on listeners, artists, songs, playlists, and play history.
  + Tables will include Listener, Artist, Song, Playlist, and PlayHistory, ensuring data integrity and consistency.
* **Admin Interface:**
  + An administrative interface will allow the management of users, artists, songs, and playlists.
  + Admin users can view and modify data for all aspects of the platform, including user details, song catalogs, and playlists.

#### Technologies Used:

* **Frontend:** HTML, CSS, JavaScript (for a responsive and interactive user interface).
* **Backend:** Python (Flask for server-side logic), MariaDB (for database management).
* **Database:** MariaDB to store all relevant data like users, songs, playlists, etc.

The system will be built as a web application with an easy-to-use interface, ensuring a great user experience for both administrators and listeners. The backend logic will ensure the smooth functionality of adding, updating, and deleting records.

#### Diagrams:

