

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING AND
INFORMATION TECHNOLOGY



SANGEET: A MUSIC MANAGEMENT SYSTEM

Enrollment No.	Name
Alokik Garg	22103302
Siddharth Agrawal	22103279
Manav Upadhyay	22103278

Course Name: Database System and Web Lab

Course Code: I5BI7CI37I

Program: B. Tech. CS&E

2nd Year 3rd Sem

2023 – 2024

INTRODUCTION

OVERVIEW

The Music Player Management System is a comprehensive web development project aiming to replicate the functionality of popular music streaming platforms like Spotify. Using a combination of HTML, CSS, JavaScript, PHP, and MySQL, the goal is to create an intuitive and feature-rich web application for users to manage and stream their music library.

SCOPE

This system will provide users with a user-friendly interface to organize, upload, and manage their music collection. It will include functionalities such as user registration and authentication, music library management, playlist creation, search and discovery options, real-time music streaming, and an admin panel for user and content management.

TECHNOLOGIES USED

The project leverages several technologies:

- **HTML/CSS:** For building the frontend user interface, ensuring a responsive and visually appealing design.
- **JavaScript:** Used to implement interactive features, real-time streaming functionalities, and client-side validations.
- **PHP:** Manages server-side logic, handles authentication, and interacts with the database.
- **MySQL:** Stores user data and music metadata in a relational database.

OBJECTIVES

PROJECT OBJECTIVES

The primary objectives of the Music Player Management System include:

- Creating a web-based music player with an intuitive interface.
- Implementing user registration and secure authentication mechanisms.
- Enabling users to upload, organize, and manage their music library efficiently.
- Providing features for creating and managing playlists.
- Facilitating music search and discovery functionalities.
- Enabling real-time music streaming capabilities.
- Developing an admin panel for user account and content management.

NORMALIZATION OF DATABASES

The database structure is normalized to ensure efficient data organization and reduce redundancy. This involves breaking down data into separate related tables to minimize data duplication and inconsistencies. Each table is structured to store specific types of information, promoting data integrity and scalability.

USERS TABLE:

- **Attributes:** user_id, email_id, username, password, phone_no, registration_date, user_type
- **Purpose:** Stores user information including login credentials, contact details, registration date, and user type (listener, artist, admin).

ALBUMS TABLE:

- **Attributes:** album_id, artist_id, album_no, album_name
- **Purpose:** Manages album-related data, including album number, name, and their association with artists through foreign key artist_id.

ARTISTS TABLE:

- **Attributes:** artist_id, artist_name
- **Purpose:** Contains information about artists, primarily their names, and serves as a reference for albums through the artist_id.

COMMENTS TABLE:

- **Attributes:** comment_id, user_id, song_id, comment
- **Purpose:** Stores user comments related to specific songs through the song_id and user_id references.

LIKES TABLE:

- **Attributes:** user_id, song_id
- **Purpose:** Manages user likes on individual songs through references to user_id and song_id.

MUSIC TABLE:

- **Attributes:** music_id, album_id, song_id, title, artist, genre, file_path, thumbnail, likes
- **Purpose:** Stores song-specific data including title, artist, genre, file paths, associated albums, and likes.

PLAYLISTS TABLE:

- **Attributes:** playlist_id, user_id, playlist_name, description
- **Purpose:** Manages user-created playlists along with their names, descriptions, and association with users through user_id.

PLAYLIST_SONGS TABLE:

- **Attributes:** playlist_id, song_id
- **Purpose:** Establishes the relationship between playlists and songs by mapping the playlist_id to the song_id.

This database structure allows for efficient organization, management, and retrieval of data related to users, music, artists, albums, comments, likes, and playlists. Relationships between tables facilitate the seamless interaction between different entities within the music player management system.

USERS TABLE:

- **First Normal Form (1NF):**
Attributes like **user_id**, **email_id**, **username**, **password**, **phone_no**, **registration_date**, and **user_type** store atomic values, ensuring there are no repeating groups.
- **Second Normal Form (2NF):**
No partial dependencies exist. All non-key attributes depend on the entire primary key (**user_id**).
- **Third Normal Form (3NF):**
No transitive dependencies. Attributes like **phone_no** are independent of non-key fields and only depend on the primary key (**user_id**).

ALBUMS TABLE:

- **1NF:**
The **album_id**, **artist_id**, **album_no**, and **album_name** attributes hold atomic values without repeating groups.
- **2NF:**
All non-key attributes are fully functionally dependent on the primary key (**album_id**).
- **3NF:**
There are no transitive dependencies, ensuring attributes like **album_no** and **album_name** are directly dependent on the primary key (**album_id**).

ARTISTS TABLE:

- **1NF:**
artist_id and **artist_name** fields contain atomic values without repeating groups.
- **2NF:**
All attributes are fully functionally dependent on the primary key (**artist_id**).
- **3NF:**
There are no transitive dependencies, maintaining direct dependence on the primary key (**artist_id**).

COMMENTS TABLE:

- **1NF:**
comment_id, **user_id**, **song_id**, and **comment** hold atomic values without repeating groups.
- **2NF:**
All non-key attributes are fully functionally dependent on the primary key (**comment_id**).
- **3NF:**

There are no transitive dependencies. The **comment** field relies solely on the primary key (**comment_id**).

LIKES TABLE:

- **1NF:**
user_id holds atomic values, and **user_id, song_id** together form a composite primary key.
- **2NF:**
All non-key attributes depend on the entire composite primary key (**user_id, song_id**).
- **3NF:**
There are no transitive dependencies, ensuring each attribute directly relies on the composite primary key.

MUSIC TABLE:

- **1NF:**
music_id, album_id, song_id, title, artist, genre, file_path, thumbnail, likes are atomic without repeating groups.
- **2NF:**
All non-key attributes are fully functionally dependent on the primary key (**music_id**).
- **3NF:**
There are no transitive dependencies, maintaining direct dependence on the primary key (**music_id**).

PLAYLISTS TABLE:

- **1NF:**
playlist_id, user_id, playlist_name, description are atomic without repeating groups.
- **2NF:**
All non-key attributes are fully functionally dependent on the primary key (**playlist_id**).
- **3NF:**
There are no transitive dependencies, maintaining direct dependence on the primary key (**playlist_id**).

PLAYLIST_SONGS TABLE:

- **1NF:**
playlist_id and **song_id** hold atomic values without repeating groups, together forming the composite primary key.
- **2NF:**

All non-key attributes depend on the entire composite primary key (**playlist_id**, **song_id**).

- **3NF:**

There are no transitive dependencies, ensuring each attribute directly relies on the composite primary key.

FEATURES

FEATURE LIST

1. **User Registration and Authentication:** Users can securely register and log in using email and password credentials.
2. **Music Library Management:** Users can upload, organize, edit metadata, and delete songs categorized by album, artist, and genre.
3. **Playlist Creation and Management:** Custom playlists can be created, edited, and managed by adding or removing songs.
4. **Music Search and Discovery:** Users can search for songs, artists, and albums, receive recommendations based on preferences and access artist and album profiles.
5. **Real-time Music Streaming:** Enables users to stream music with playback controls and queue management for continuous listening.
6. **Admin Panel:** Admins have control over user accounts, content moderation, and access to user statistics and analytics.

CODE SUMMARY

PHP ALGORITHMS

USER AUTHENTICATION:

STORING PASSWORDS:

- **Hashing Passwords:** Utilize hashing algorithms (like bcrypt or SHA) to securely hash and store user passwords in the database.
- **Salting:** Incorporate unique salts for each user before hashing passwords to enhance security against precomputed hash attacks.

AUTHENTICATING USERS:

- **Login Verification:** Compare the hashed input password with the stored hashed password during the login process.
- **Session Creation:** Upon successful verification, create a session for the user, allowing access to authenticated functionalities.

DATABASE INTERACTIONS:

CRUD OPERATIONS:

- **Create:** Functions to create new user accounts, upload music metadata, create playlists, and register likes/dislikes.

- **Read:** Retrieve user data, music metadata, playlists, and liked tracks from the database based on user interactions.
- **Update:** Update user profiles, edit music metadata, modify playlists, and manage likes/dislikes.
- **Delete:** Functionality to remove user accounts, delete songs, playlists, or undo likes from the database.

SESSION MANAGEMENT:

USER SESSIONS:

- **Session Creation:** Upon successful authentication, create a session token or identifier and store it in the user's browser.
- **Session Validation:** Validate session tokens on each user interaction to ensure authorized access to functionalities.
- **Expiration & Renewal:** Implement session expiration mechanisms and renew sessions on user activity to maintain active sessions securely.

ACCESS CONTROL:

- **Authorization Levels:** Define user roles (listener, artist, admin) and implement access controls based on these roles.
- **Permission Handling:** Set permissions for various operations (e.g., CRUD) based on user roles to control database access.

JAVASCRIPT FUNCTIONALITIES

REAL-TIME STREAMING:

PLAYBACK CONTROL:

- **Play / Pause / Stop:** Functions to handle play, pause, and stop actions for music tracks.
- **Skip Track:** Functionality to move to the next or previous track in the queue.
- **Volume Control:** Functions to adjust the volume of the streaming audio.

QUEUE MANAGEMENT:

- **Add to Queue:** Functionality to add songs to the streaming queue.
- **Remove from Queue:** Removing songs from the queue while maintaining the playback flow.
- **Reorder Queue:** Functions allowing users to rearrange the song order in the queue.

USER INTERACTION:

INPUT HANDLING:

- **Form Submission:** Functions handling form submissions for user interactions (e.g., login, registration, music uploads).
- **Search Functionality:** Implementing search functionality for songs, artists, or albums with live updates.
- **UI Interaction:** Functions to handle user interactions within the UI elements (buttons, dropdowns, etc.).

PLAYLIST MANAGEMENT:

- **Create Playlist:** Functions for creating new playlists.
- **Add / Remove from Playlist:** Adding or removing songs from the playlists.
- **Edit Playlist Details:** Functions to modify playlist names, descriptions, or cover images.
- **Playlist Sorting:** Sorting songs within a playlist based on user preferences.

ANTICIPATED OUTCOME

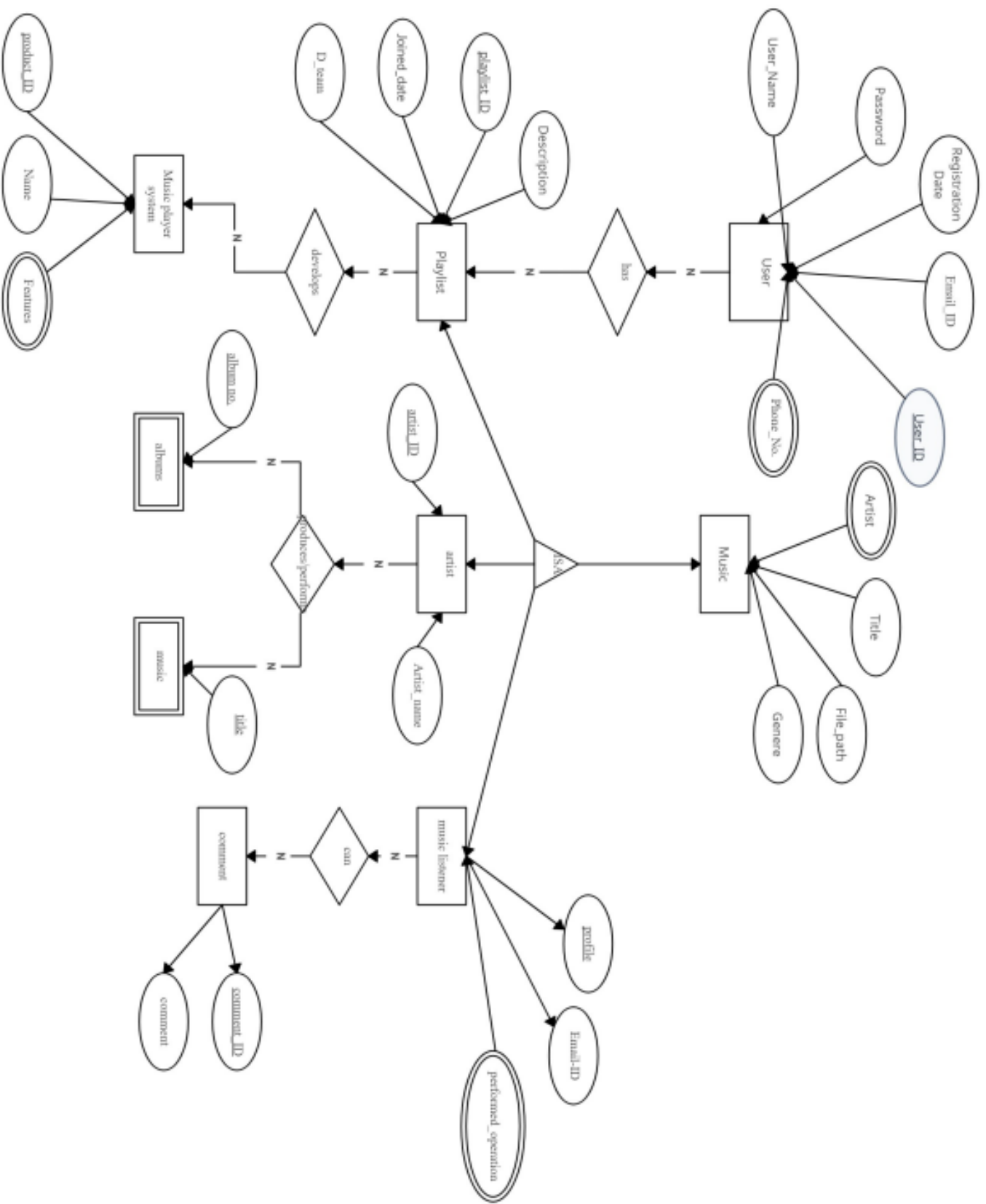
The project aimed for a feature-rich music player with user management, playlist creation, search functionalities, real-time streaming, and an admin panel. Achieved milestones include a functional music player with basic user management, playlist features, search capabilities, streaming controls, a responsive app design, initial security measures, successful deployment, and early maintenance procedures.

CONCLUSION

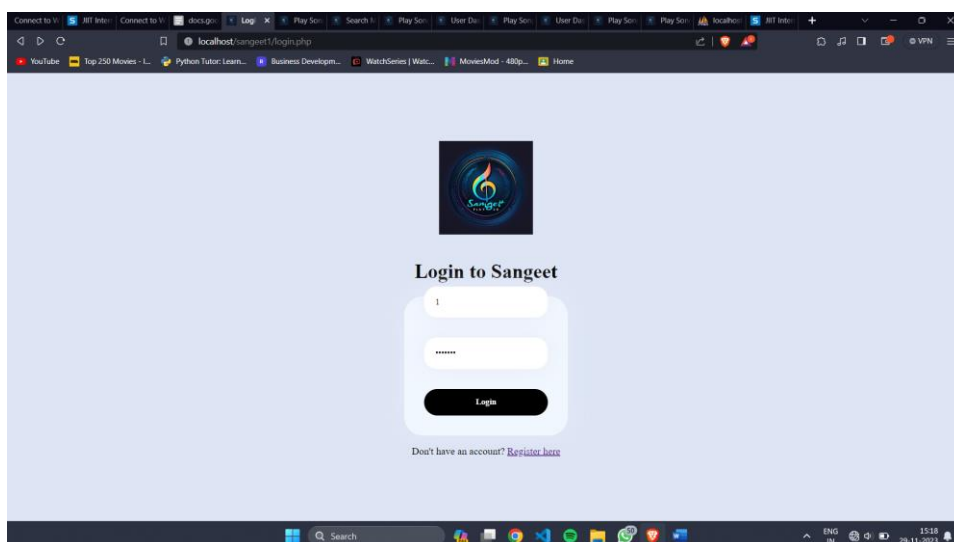
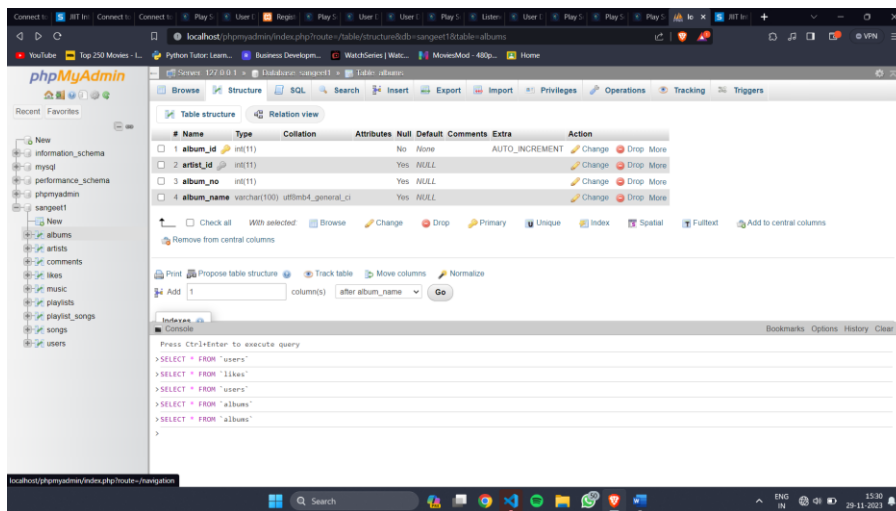
The Music Player Management System endeavors to provide an all-encompassing music streaming experience akin to popular platforms. Its feature-rich architecture, robust backend, and user-friendly interface aim to enhance music enthusiasts' interactions. Through meticulous planning, rigorous development, and adherence to security and scalability, the project aspires to offer a seamless musical journey to its users.

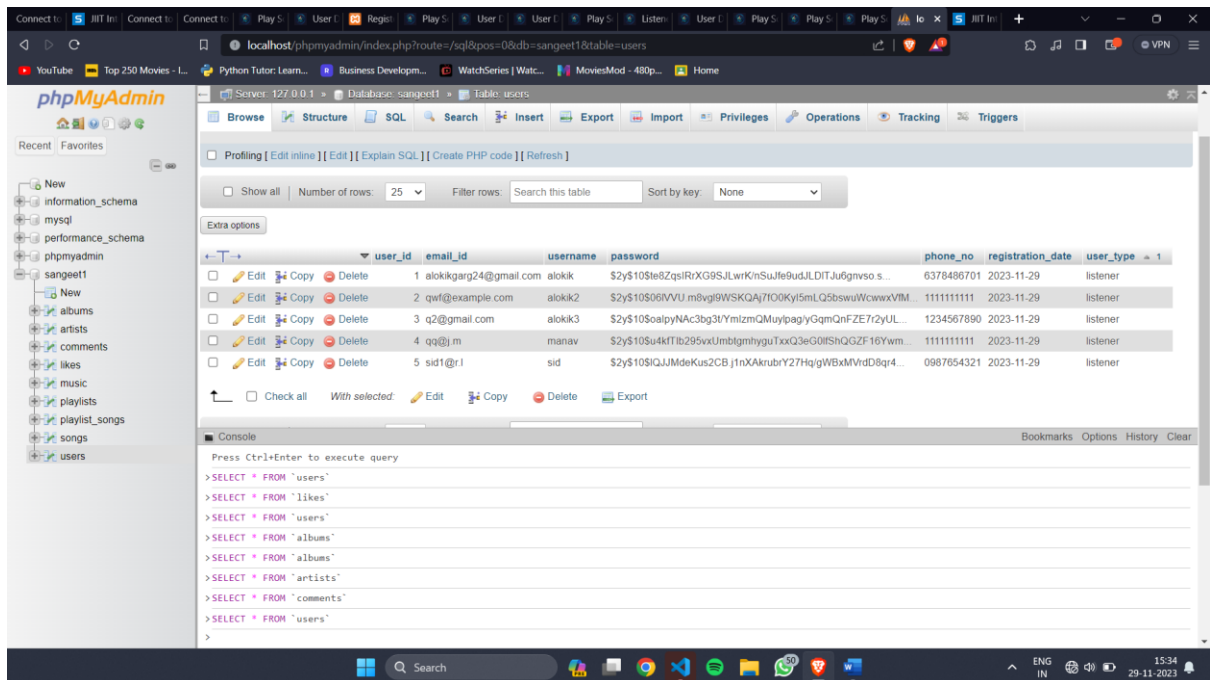
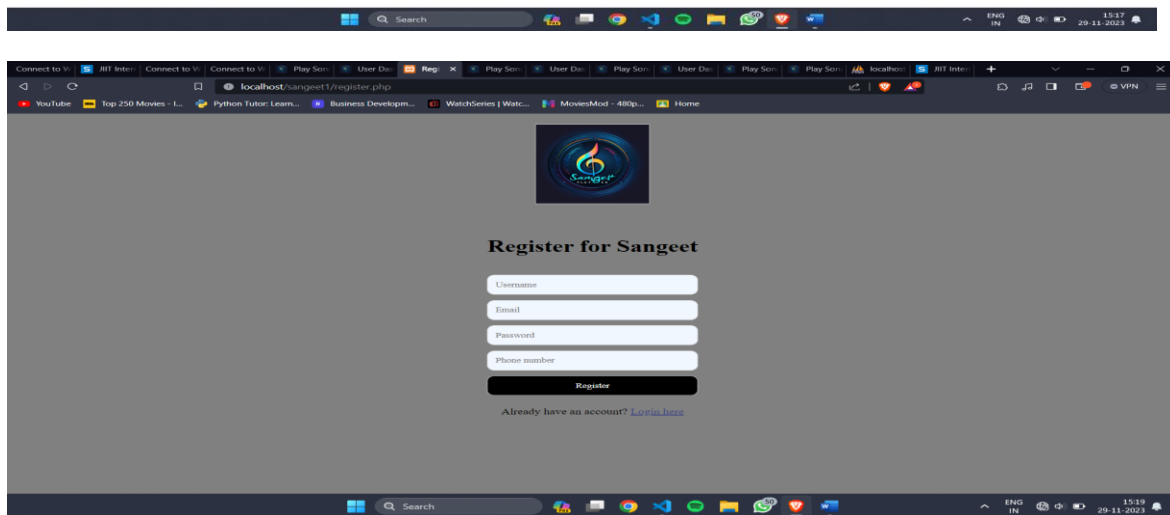
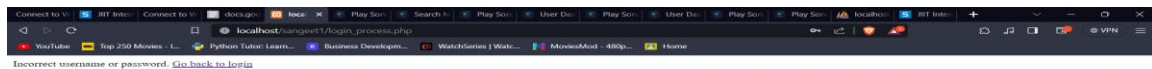
APPENDIX

Source Code along with MySQL Database dump attached in Zip Folder



SNAPSHOTS





localhost/phpmyadmin/index.php?route=/sql&pos=0&db=sangeet1&table=comments

phpMyAdmin

Database: sangeet1

	comment_id	user_id	song_id	comment
<input type="checkbox"/>	1	3	5	s
<input type="checkbox"/>	2	3	5	dd
<input type="checkbox"/>	3	3	8	ss
<input type="checkbox"/>	4	3	1	hiu
<input type="checkbox"/>	5	4	3	good
<input type="checkbox"/>	6	5	2	qwe

Query results operations

Console

```
Press Ctrl+Enter to execute query
> SELECT * FROM `users`
> SELECT * FROM `likes`
> SELECT * FROM `users`
> SELECT * FROM `albums`
> SELECT * FROM `albums`
> SELECT * FROM `artists`
> SELECT * FROM `comments`
```

localhost/sangeet1/dashboard.php

Search: Search for songs, artists

Welcome to Your Music Dashboard

Listener Account Trending Logout Search








Playlists

Liked Songs


New Playlist:

Create Playlist

Recommendations

-  [Maahi Ve - A.R. Rahman](#)
-  [Pehla Pyaar - Armaan Malik](#)
-  [Tera Zikr - Darshan Raval](#)
-  [Dance The Night \(From Barbie The Album\) \[Official Music Video\] - Dua Lipa](#)
-  [Reliever - Imagine Dragons](#)
-  [the magical woods - Kaazoom](#)
-  [Iktara - Kavita Seth, Amit Trivedi, Amitabh Bhattacharya](#)

localhost/sangeet1/song-details.php?song_id=3



Tera Zikr
Darshan Raval

0:00 / 3:28

You've already liked this song.


Comments
Add your comment...

Add Comment
manav: good

- Maahi Ve - A.R. Rahman
- keeps me high - Kellepics
- Pehla Pyaar - Armaan Malik
- Tera Zikr - Darshan Raval
- Song "hum" Water - Tyla, Travis Scott
- the magical woods - Kaazoom
- ghost - urban beatsss
- Khwab ho tum ya koi haqeeqat - kishore Kumar
- Kisi Ki Muskurahat pe - kishore
- Song "hum" I Remember Everything (feat. Kacey Musgraves) - Zach Bryan
- Taarif Karoon Kya Uski - Mohammed Rafi
- Song "hum" Aa chalke tujhe - Kishore kumar
- Iktara - Kavita Seth, Amit Trivedi, Amitabh Bhattacharya

localhost/sangeet1/listener_account.php

Search: Search for songs, artists Search



Welcome, sid!
Dashboard Trending Logout Search

User Information


User ID	5
Email ID	sid1@r.l
Phone Number	0987654321
Registration Date	2023-11-29
User Type	listener

Liked Songs

Connect to: JIT Int. | Connect to: | Connect to: | Play Sc. | User D. | Regist. | Play Sc. | User D. | Play Sc. | Listene | x | Play Sc. | Play Sc. | localh: | JIT Int. | + | - | x

localhost/sangeet1/trending.php

Search: Search for songs, artists Search





Spotlight


Listener Account Dashboard Logout Search

Trending Songs

Songs by Likes

 Maahi Ve - A.R. Rahman
Likes: 8

 Pehla Pyaar - Armaan Malik
Likes: 3


 keeps me high - Kellepics
Likes: 3

ENG IN 15:23 29-11-2023

Connect to: JIT Int. | Connect to: | Connect to: | Play Sc. | User D. | Regist. | Play Sc. | User D. | Play Sc. | Listene | x | Play Sc. | Play Sc. | localh: | JIT Int. | + | - | x


localhost/sangeet1/search.php


Search: Search for songs, artists, or Search



Listener Account Trending Logout Dashboard

Thumbnail
Aa chalke tujhe - Kishore kumar

 Kisi Ki Muskurahaton pe - kishore

 Khwab ho tum ya koi haqeeqat - kishore Kumar

ENG IN 15:23 29-11-2023