# ONLINE MUSIC MANAGEMENT SYSTEM

A COURSE PROJECT REPORT

By

**CHALLA AVINASH (RA2011027010107)**
**KOTHA NARASIMHA(RA2011027010120)**
**VADLAPUTI DINESH (RA2011027010121)**

Under the guidance of

**Dr. V. Vijayalakshmi.**

**Assistant professor**

**Department of Data Science and Business Systems**

In partial fulfillment for the Course

of

**18CSC303J - DATABASE MANAGEMENT SYSTEM**

in

**Department of Data Science and Business Systems**

**COLLEGE OF ENGINEERING AND TECHNOLOGY**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**Kattankulathur, Chengalpattu District**

COLLEGE OF ENGINEERING & TECHNOLOGY
SRM INSTITUTE OF SCIENCE & TECHNOLOGY
S.R.M. NAGAR, KATTANKULATHUR – 603 203

# BONAFIDE CERTIFICATE

Certified that this project report **" Music management system"** is the bonafide work of **" Challa Avinash(RA2011027010107),Kotha Narasimha Rao(RA2011027010120), Vadlaputi Dinesh(RA2011027010121)"** of III Year/VI Sem B.tech(CSE) who carried out the mini project work under my supervision for the course 18CSC303J- Database Management systems in SRM Institute of Science and Technology during the academic year 2022-2023(Even sem).

*SIGNATURE*

Faculty name
Faculty Designation
Department name and seal

# ABSTRACT

The aim of this project MUSIC MANAGEMENT SYSTEM is the development of a sample centralized relational MUSIC store application. It is aim is to formally describe the design and development. These phases are categorized into 3 main steps: Database, Application and Graphic User Interface. The listen up is online website where people can listen to music and order their favorite albums. The result is a database which unable the clerks and manager of music store to manage the information of the customers, artist, album and songs and the customers to their favorite music where we can see their listening and we can add their favourite playlists according to their music preferences. This is an eco-friendly website.

The system allows users to easily manage their music collection, including adding new songs, editing metadata, and creating playlists. Users can also search for music based on various criteria, such as artist, album, genre, or year. The system may also provide recommendations for new music based on the user's listening history.

In addition to organizing and managing music files, the system may also provide features for playing music, such as a media player with playback controls, equalizer settings, and support for various audio formats.

Overall, a music management system provides a comprehensive solution for users to manage and enjoy their music collection.

## TABLE OF CONTENTS


| Chapter No. | | Title | Page No. |
|---|---|---|---|
| | | **ABSTRACT** | **iii** |
| | | **TABLE OF CONTENTS** | **iv** |
| | | **LIST OF FIGURES** | **v** |
| | | **LIST OF TABLES** | **vi** |
| | | **ABBREVIATIONS** | **vii** |
| **1** | | **INTRODUCTION** | **1** |
| | 1.1 | Introduction | 1 |
| | 1.2 | Problem statement | 2 |
| | 1.3 | Objectives | 3 |
| | 1.4 | Scope and applications | |
| | 1.5 | General and Unique Services in the database application | |
| | 1.6 | Software Requirements Specification | |
| **2** | | **LITERATURE SURVEY** | **11** |
| | 2.1 | Existing system | 11 |
| | 2.2 | Comparison of Existing vs Proposed system | 11 |
| **3** | | **SYSTEM ARCHITECTURE AND DESIGN** | **12** |
| | 3.1 | Architecture Diagram | 12 |
| | 3.1.1 | Front end (UI)design | 13 |
| | 3.1.2 | Back end (Database) design | 19 |
| | 3.2 | ER Diagram and Use case Diagram | 25 |
| **4** | | **Modules and Functionalities** | **27** |
| | 4.1 | Connectivity used for database access | 28 |
| **5** | | **CODING AND TESTING** | **41** |
| **6** | | **RESULTS AND DISCUSSIONS** | **43** |
| | 6.1 | Result | 43 |
| | 6.2 | Discussions | 43 |

| | | |
|---|---|---|
| **7** | **CONCLUSION AND FUTURE ENHANCEMENT** | **44** |
| | **REFERENCES** | **46** |

# LIST OF FIGURES


| Figure No. | Figure Name | Page No. |
|---|---|---|
| 4.1 | Proposed Architecture Diagram | 13 |
| 4.2 | Use Case Diagram | 27 |
| 4.3 | ER Diagram | 28 |

# LIST OF TABLES


| Table No. | Table Name | Page No. |
|---|---|---|
| 1 | Login Table | 34 |
| 2 | Songs Table | 35 |
| 3 | Singers Table | 36 |
| 4 | Movies Table | 37 |
| 5 | Playlist Table | 38 |
| 6 | Trendings Table | 39 |

# ABBREVIATIONS

| | |
|---|---|
| **HTML** | Hypertext Markup Language |
| **CSS** | Cascading Style Sheet |
| **PHP** | Hypertext Preprocessor |
| **DB** | Data Base |
| **SQL** | Structured Query Language |
| **SVM** | Support Vector Machine |
| **UI** | User Interface |

# INTRODUCTION

A music management system is a software application designed to help manage various aspects of the music industry, including artist management, recording, distribution, marketing, and sales. It is an essential tool for musicians, record labels, and other music industry professionals to streamline their workflow, keep track of important data, and make informed business decisions.

The system typically includes features such as music catalog management, royalty tracking, accounting and financial reporting, contract management, and communication tools. It may also incorporate advanced analytics and data visualization capabilities to help music professionals gain insights into the performance of their music and make informed decisions about future projects.

Overall, a music management system is a critical tool for anyone looking to navigate the complexities of the music industry, maximize their revenue, and build a successful career in music.

Music management database provides the user with relevant information regarding the preferred music genre, which is popular in their respective countries. The music database can be used by various music streaming platforms like, spotify, apple music etc. It consists of various tables with the relevant entities. For the ease of this project, we have used FOUR major tables. The user can easily access any required entry from the database using some simple PL/SQL queries. DDL/DML commands, basic/advanced select statements, PL/SQL statements, functions, procedures, triggers, cursors and exception handling.

Users can search and browse the songs in the platform's database by applying specific filters, such as genre, artist, and song name. The application also provides general information about each song, such as its name, genre, and artist, making it easy for users to find the songs they are looking for.

Users can also express their opinions on the songs through a "like" system and possibly add them to their favorites list for easy access later. In addition, users can organize songs into playlists or follow other people's playlists to discover new songs and improve the listening experience.

## 1.2 Problem Statement

This music application is an online website for a musical store. The application is a virtual showcase for everything related to a music store. This site has information regarding the latest songs, albums, and artists. An online music application offers the customers, the chance to hear performances of their favorite artists or the most liked songs. And this application has to store information of customers and artists with their products. In this context the functionality is to update remove and insert different entities. The database is built for the clerks and managers of the musical store. Customers are ordering by phone or email. The clerk of the databases must be able to fulfil the the wishes of the customers. These wishes include finding the right album and ordering this album.

Furthermore, with the growing number of music streaming services and digital downloads available, users may have music files stored on multiple devices or platforms, making it even more challenging to keep track of their entire collection.

Therefore, there is a need for a music management system that can effectively organize and manage music files, providing a centralized platform for users to store, search, and play their music collection. Such a system would need to be user-friendly, able to handle a large number of files, and provide features for organizing and customizing music collections, such as creating playlists and editing metadata.

Additionally, the system should be able to integrate with various platforms and services, allowing users to easily import and export their music files, as well as stream music from their preferred services. Overall, the goal of a music management system is to simplify the process of managing and enjoying a large music collection, providing a seamless and efficient user experience.

## 1.3 Objectives

The main objective of this “listen up” (music store) is to manage the details of the music, performer, album, customer, album type, ordering details. It manages all the information about music, track, album type, ordering details. The project is totally built at administrator end. Thus, only the administrator can have the access. The purpose of the project is to build an application program to reduce the manual work for managing the music, performer, track, album, ordering details. It tracks the details about the album, customer, album type.

To achieve this objective, the system must:

1. Provide a centralized platform for storing and managing music files: The system should be able to handle a large number of music files and provide features for organizing and customizing the collection, such as creating playlists, editing metadata, and filtering by various criteria (e.g., artist, album, genre, year).
2. Provide a user-friendly interface: The system should be intuitive and easy to use, with a clear and organized layout that allows users to quickly find and play the music they want.
3. Provide seamless integration with various platforms and services: The system should be able to import and export music files from different devices and platforms, as well as stream music from popular services (e.g., Spotify, Apple Music) to provide a seamless listening experience.
4. Support various audio formats: The system should support a wide range of audio formats to ensure compatibility with different types of music files.
5. Provide high-quality playback: The system should provide a high-quality playback experience with features such as an equalizer, playback controls, and support for high-quality audio codecs.
6. Provide personalized recommendations: The system should be able to provide personalized music recommendations based on the user's listening history and preferences.
7. Overall, the objective of a music management system is to simplify the process of organizing and enjoying a user's music collection, providing a seamless and intuitive experience that enhances the user's enjoyment of music.
8. To create a user-friendly interface as the front end of the database
9. To provide wide variety of songs to listen
10. To provide an easy method for singers to distribute
11. To provide lag free listening experience

12. To curate playlists for the users based on their listening habits
13. To overcome the manual errors and make a computerzied system. In this project, there are various type of modules available to manage Album Type, Music, Album. We can also generate reports for Music, Album, Singer, Track.

## 1.4 Scope and application

It may help collecting perfect management in details. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the management of passed year perfectly and vividly. It also helps in current all works relative to Online Music Portal. It will be also reduced the cost of collecting the management & collection procedure will go on smoothly.However, the general scope of the project may include the following:

1. Design and development of a music management system: The project will involve designing and developing a software application that can effectively organize and manage a user's music collection.
2. Database design and implementation: The system will require a database to store metadata about the music files, such as artist name, album title, song title, genre, and release date. The project will involve designing and implementing a database schema to store this information.
3. User interface design and development: The system should provide a user-friendly interface for users to interact with their music collection. The project will involve designing and developing an intuitive user interface that allows users to easily navigate and manage their music collection.
4. Integration with third-party platforms and services: The system should be able to integrate with various platforms and services, allowing users to import and export their music files and stream music from their preferred services. The project will involve implementing APIs to facilitate this integration.
5. Testing and quality assurance: The project will involve testing the system to ensure that it meets the requirements and is functioning correctly. This will involve conducting unit tests, integration tests, and system tests to identify and resolve any issues.
6. By the next time I would like to add two more modules: Purchase Module and Accounting Module. Purchase Module deals with purchasing activities of music related items. Accounting Module deals with all accounting activities such as billing, ledger preparation, balance sheet preparation, profit and loss account preparation etc.
7. In the present system transaction is through a particular bank or through money orders. In future I would like to make it through credit cards. Credit card validation techniques are needed for that.

8. The developed software for the organization is flexible and it can be made to run on all kinds of platforms. The system is error free and highly portable. It can be implemented in any servers in the Internet providing an easy access to the clients. It also has more options of the future developments.

## 1.5 General and unique services in the database application

A database application for music management system can provide a range of general and unique services to help manage music data effectively. Here are some examples:

**General Services:**

**Data Entry:** Allow users to add new music data to the database, such as songs, artists, albums, genres, etc.

**Data Editing:** Allow users to edit existing music data in the database, such as updating song titles, artist names, album covers, etc.

**Data Search:** Allow users to search for music data using various search criteria, such as song title, artist name, genre, etc.

**Data Sorting:** Allow users to sort music data in various ways, such as by release date, popularity, duration, etc.

**Data Backup:** Provide a backup mechanism to protect music data from accidental loss or corruption.

**Unique Services:**

**Music Recommendation:** Provide music recommendation based on user preferences, such as recommending similar songs or artists.

**Music Analysis:** Analyze music data to provide insights into trends, popularity, user behavior, etc.

**Playlist Management:** Allow users to create and manage playlists of their favorite songs, artists, or genres.

**Music Licensing:** Manage music licensing agreements and royalties for music usage in commercial settings, such as restaurants or retail stores.

**Music Event Management:** Manage music events, such as concerts, festivals, or shows, including scheduling, ticketing, and artist management.

Overall, a database application for music management system can provide a range of services to help users manage music data effectively and efficiently, while also offering unique services to enhance user experience and value.

## 1.6 Software Requirements Specification

| | | |
|---|---|---|
| **Operating System** | **:** | WINDOWS 2000 Server |
| **Programming Language** | **:** | HTML,CSS,PHP,SQL |
| **Server Technologies** | **:** | J2EE (JSP, SERVLETS, JDBC) |
| **Application Server** | **:** | Jakarta Tomcat 4.1 |
| **Database Support** | **:** | ORACLE 8i |

## 2. Literature Survey

Literature Review: Music Technology in Education In 2009, Dammers summed up the conundrum of using Information and Communications Technology (ICT) in the field of music education, stating that because music performance is, by its very nature, synchronous, the use of ICT is problematic at best (Dammers, 2009, p. 22). Many collegiate programs are embracing the possibilities of asynchronous online education (research rather than performance), but music programs are not generally moving in that direction (Dammers, 2009, p. 22). This does not mean that technology is not being used in music education, but it does shape how it is being explored and implemented from the primary level through higher education. Before examining how technology is being used in music education, it is necessary to lay out parameters for the term. Rees (2001) defined music technology as "the systematic study of tools and techniques for music production, performance, education, and research" (Rees, 2011, p. 154). This article will deal primarily with education and research, showing how music technology – and more specifically ICT within the field of music – is being used to shape opportunities for music students of varying ages, skills and backgrounds. It will then turn to a discussion on the importance of clearly stated objectives within the research framework and consider an inherent problem in the manner in which the findings are presented.

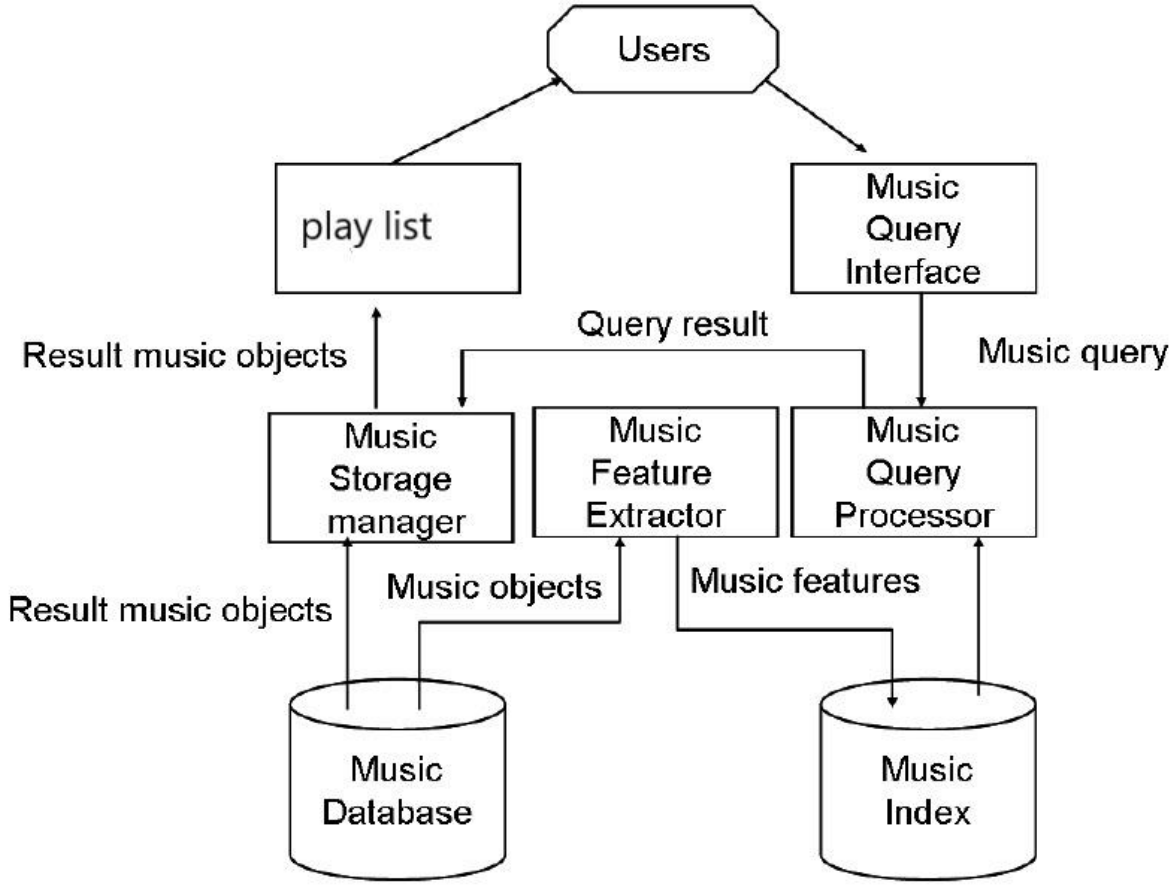
## 2.1 Existing System :

- Offers a large library of music from various artists and genres.
- Provides features such as playlist creation and sharing, song and personalized playlists.
- Allows users to search for music by artist, album, song, and genre.
- It would not allow the admin to view table in the website.

## 2.2 Comparison of Existing vs Proposed system:

- In terms of features and functionality, the proposed system offers more options for music discovery, customization, and personalization than the existing system.
- Offers more personalized and curated playlists that take into account user preferences, listening habits, and mood.
- Offers more options for customization and personalization, such as the ability to create custom playlists and mood-based radio stations.
- Overall, the proposed system could provide a more engaging and personalized music experience for users, but it would need to be carefully designed and tested to address potential technical and ethical challenges.

## 3.1Proposed Architecture Diagram



## 3.1.1Front end (Database) design, software used with explanation

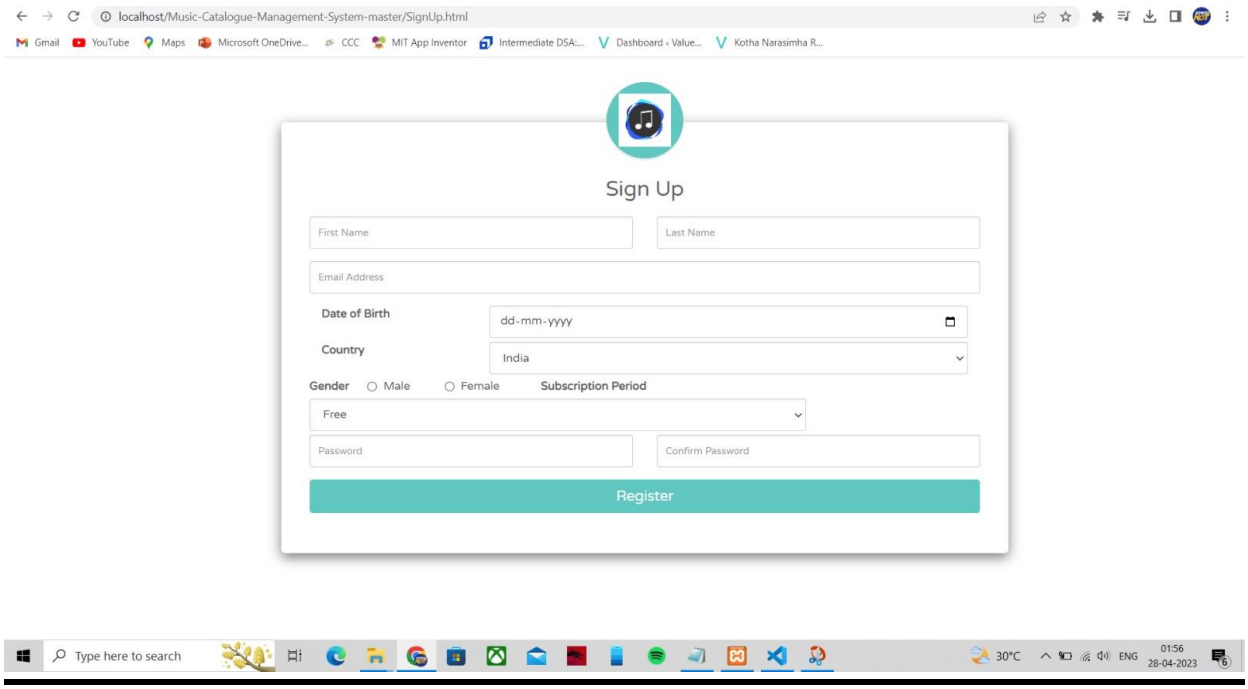



FIG.1: This is signup page

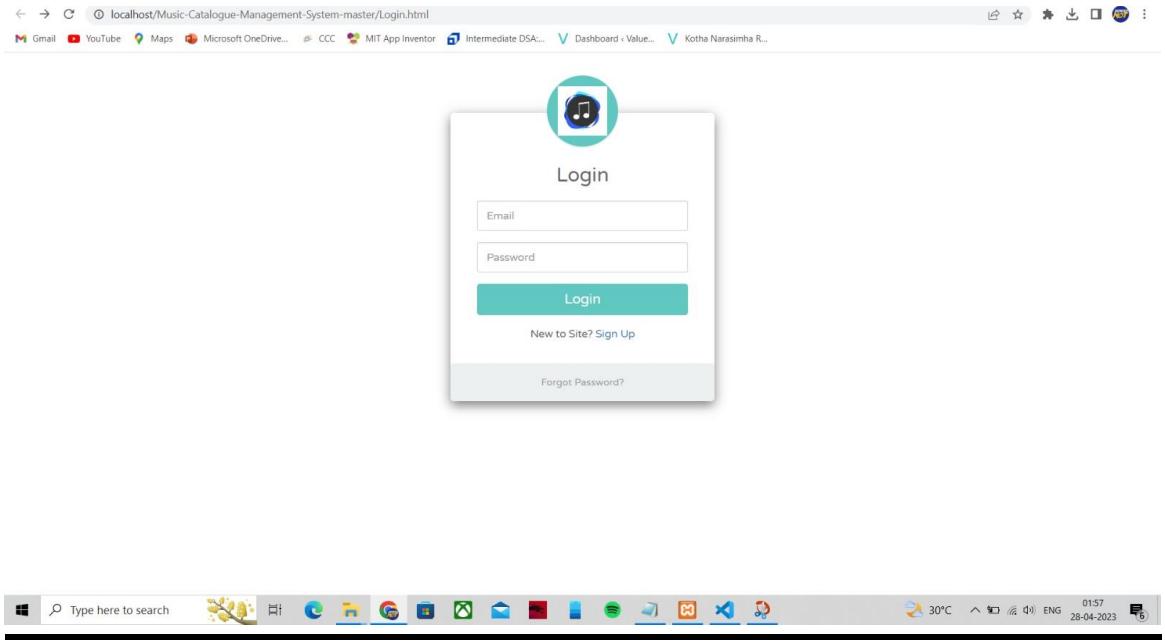



FIG.2: This is Login page

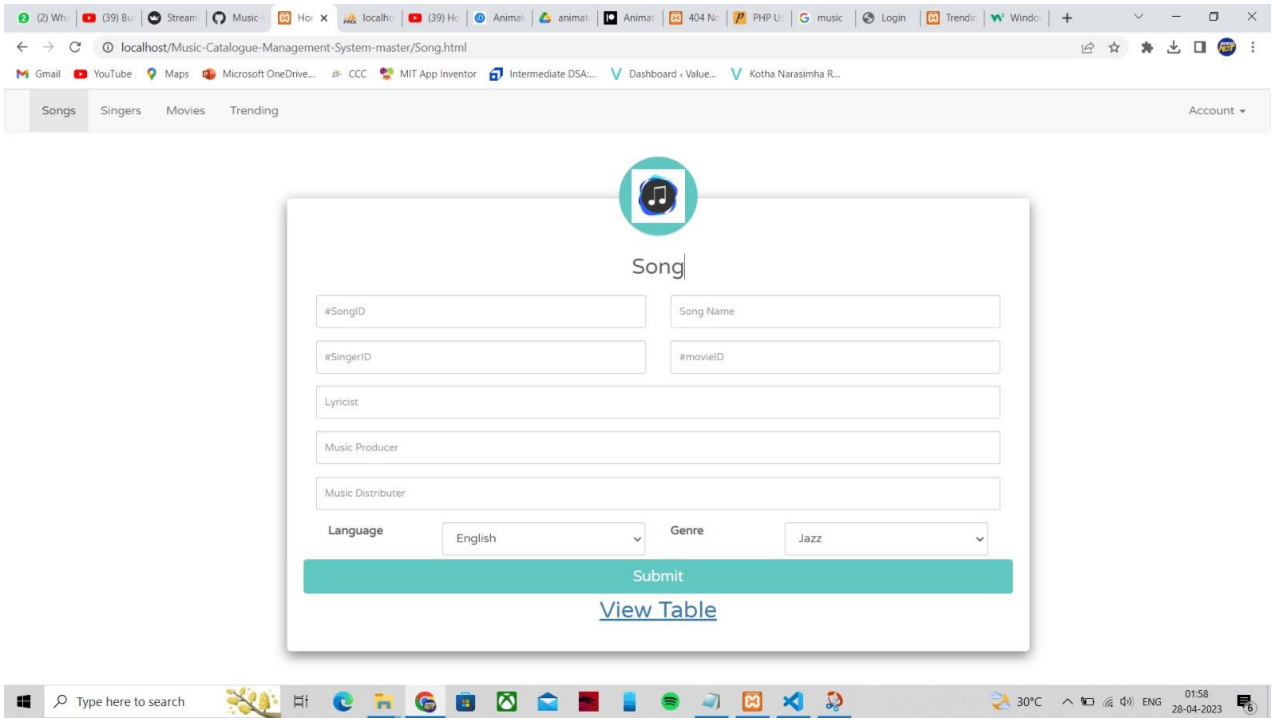



FIG.3: This is songs page

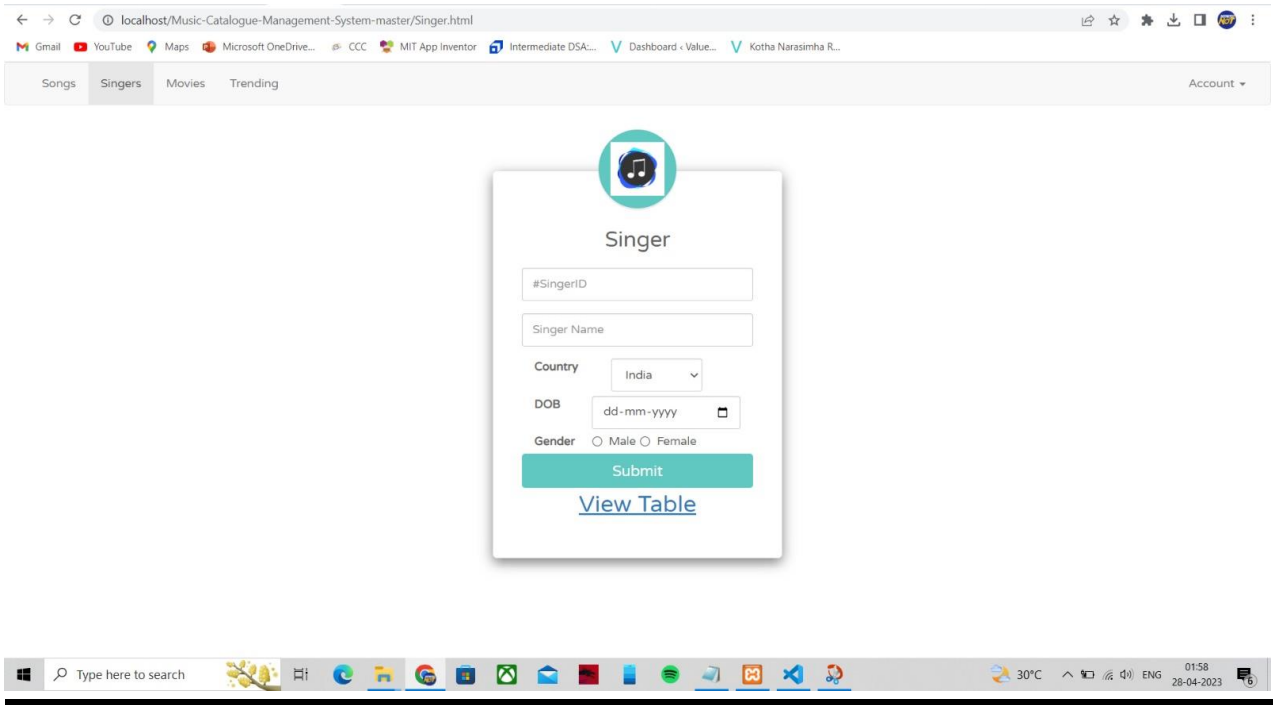



FIG.4: This is singers page

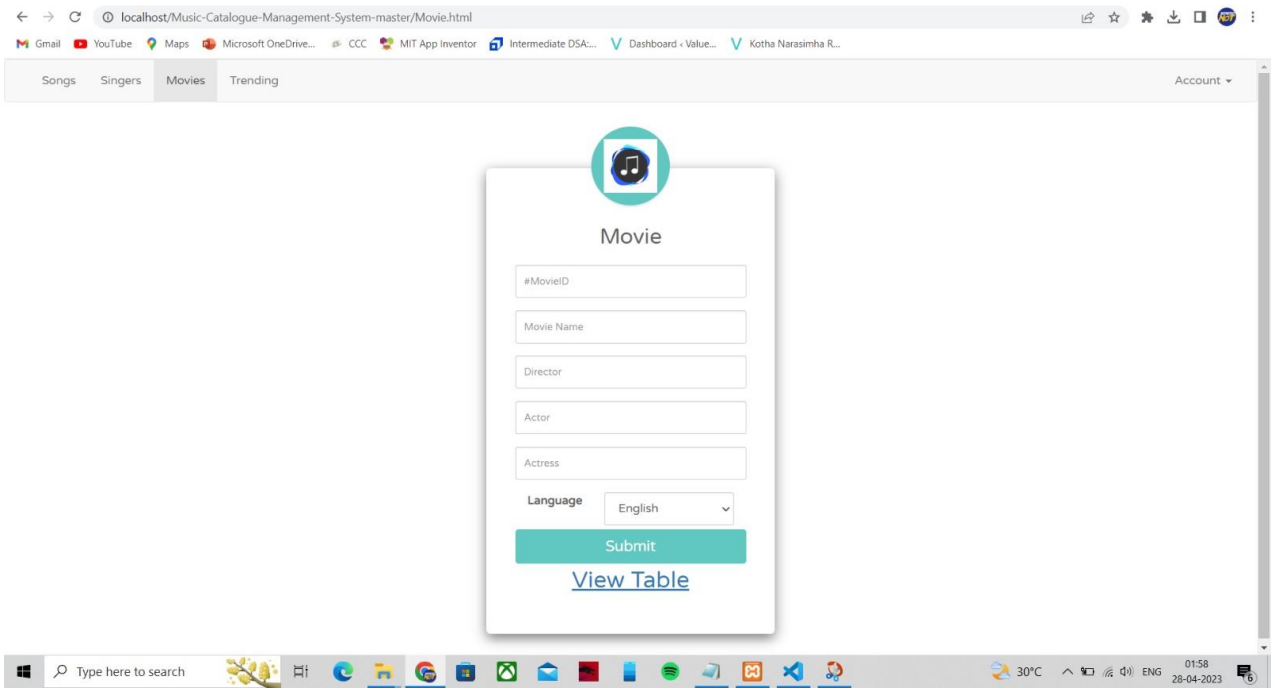



FIG.5: This is movies page

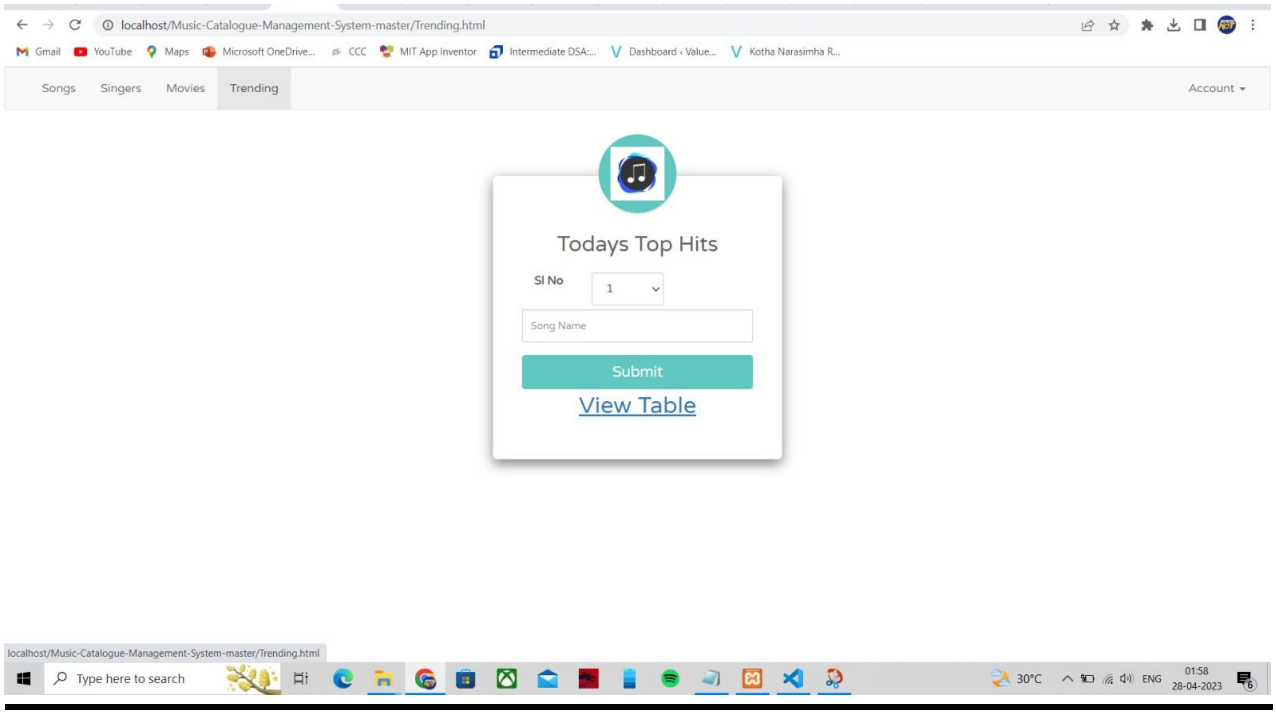



FIG.6: This is trending page

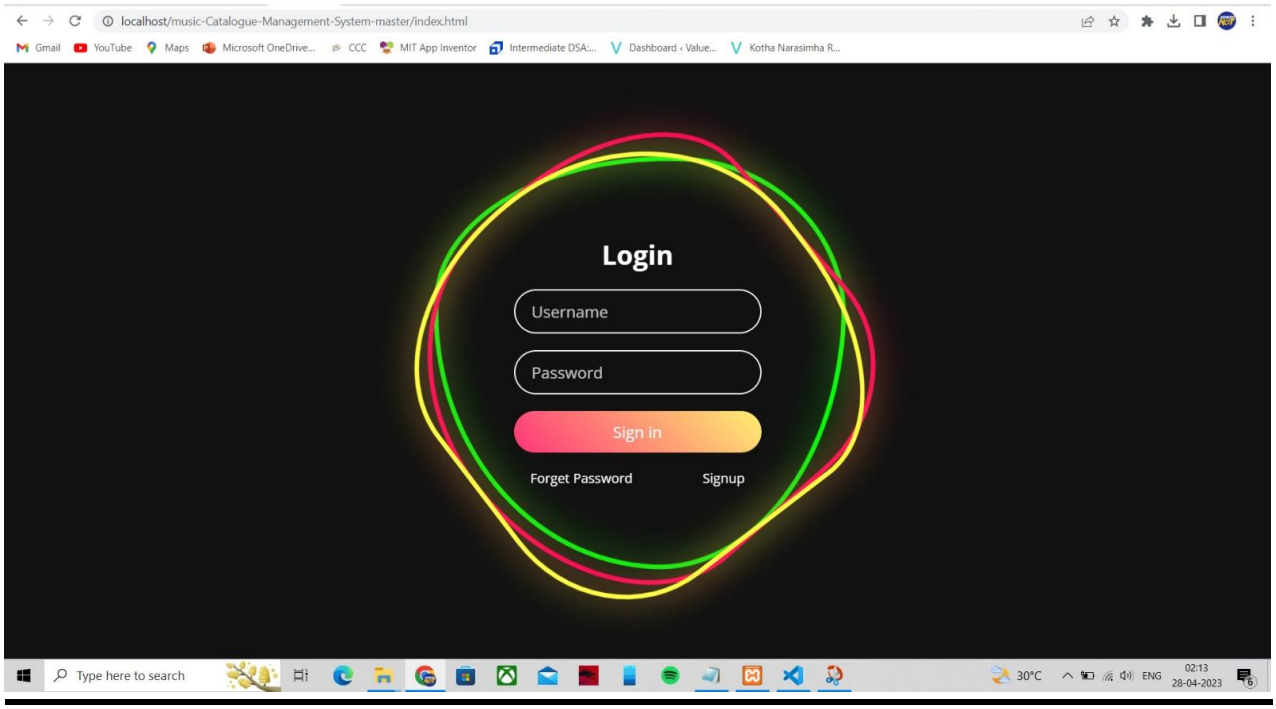



FIG.7: This is user login page

We have used the following front-end languages:

## • HTML

Hypertext Mark-up Language (HTML) is the standard mark-up language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other elements are delineated by tags, written using angle brackets. Tags such as surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

- HTML is used to define the structure of web pages, including text, images, videos, and links. It uses a set of markup tags, also known as elements, to define the structure and content of the page. HTML tags are enclosed in angle brackets and are typically used in pairs, with the opening tag indicating the start of an element and the closing tag indicating the end.

- HTML includes a range of elements, such as headings, paragraphs, lists, tables, and forms. It also provides support for multimedia elements, such as images, audio, and video, and allows for the inclusion of scripts and stylesheets for enhanced functionality and styling.

- HTML is widely used by web developers to create static and dynamic web pages. It is constantly evolving, with new versions and features being released regularly. HTML5 is the latest version of HTML, which provides a range of new features and functionality, including support for multimedia elements and enhanced accessibility

## • CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a mark-up language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

- CSS, which stands for Cascading Style Sheets, is a style sheet language used to describe the presentation of web pages. It is used to control the layout, formatting, and appearance of web pages, including fonts, colours, spacing, and positioning.
- CSS is used in conjunction with HTML and JavaScript to create responsive and visually appealing web pages. CSS styles are typically defined in separate files or within the HTML file using the "style" tag. They are applied to HTML elements using selectors, which target specific elements or groups of elements on the page.
- CSS includes a wide range of features, such as layout and positioning of elements, typography, and colour management. It also provides support for responsive design, allowing web pages to adapt to different screen sizes and devices.
- CSS is widely used by web developers to create visually appealing and user-friendly web pages. It is supported by all modern web browsers and is constantly evolving to include new features and functionality.

## 3.1.2Back end (Database) design, software used with explanation

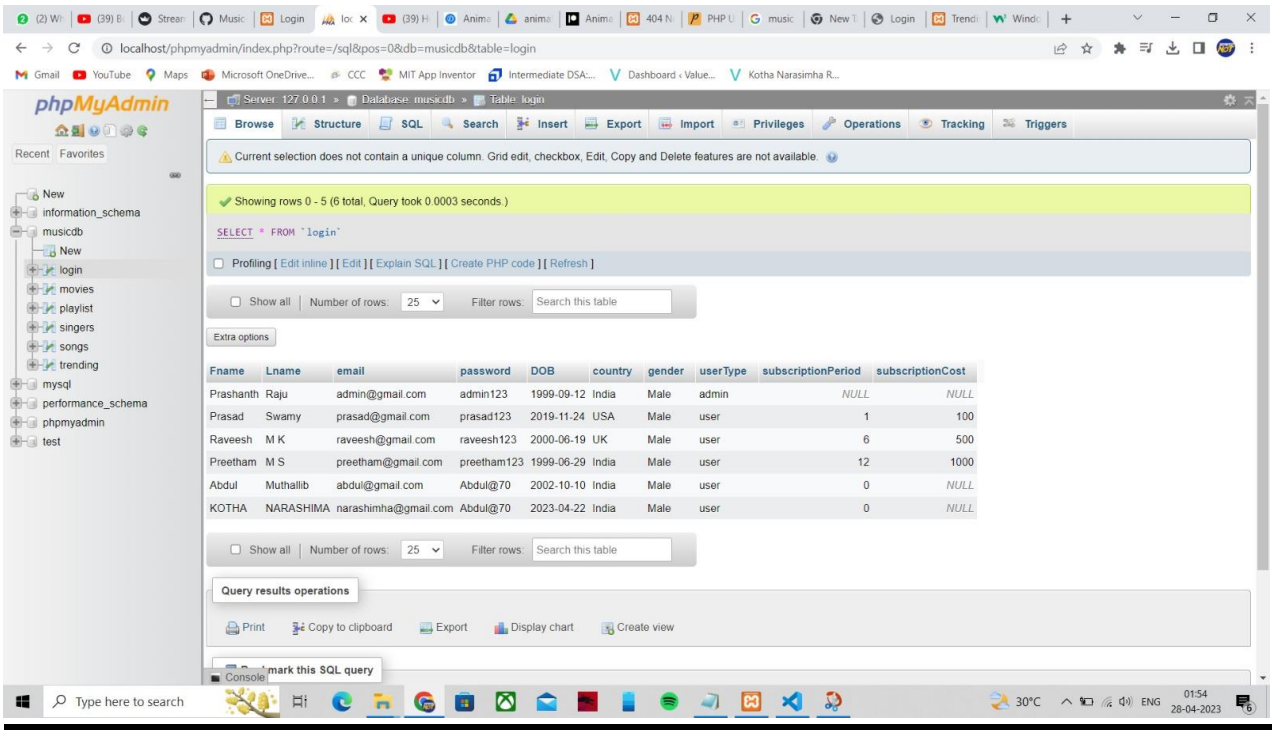



FIG.7: This table shows the login details of user

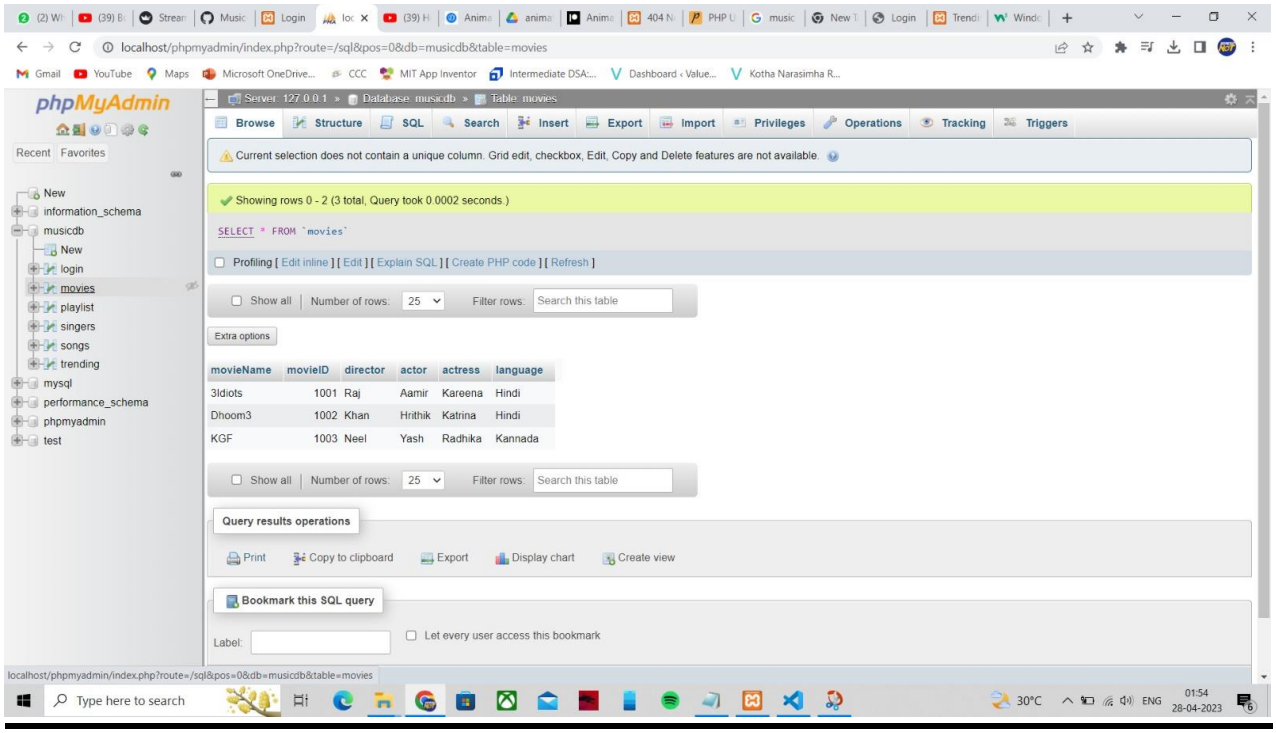



FIG.8: This table shows the movie details of the song

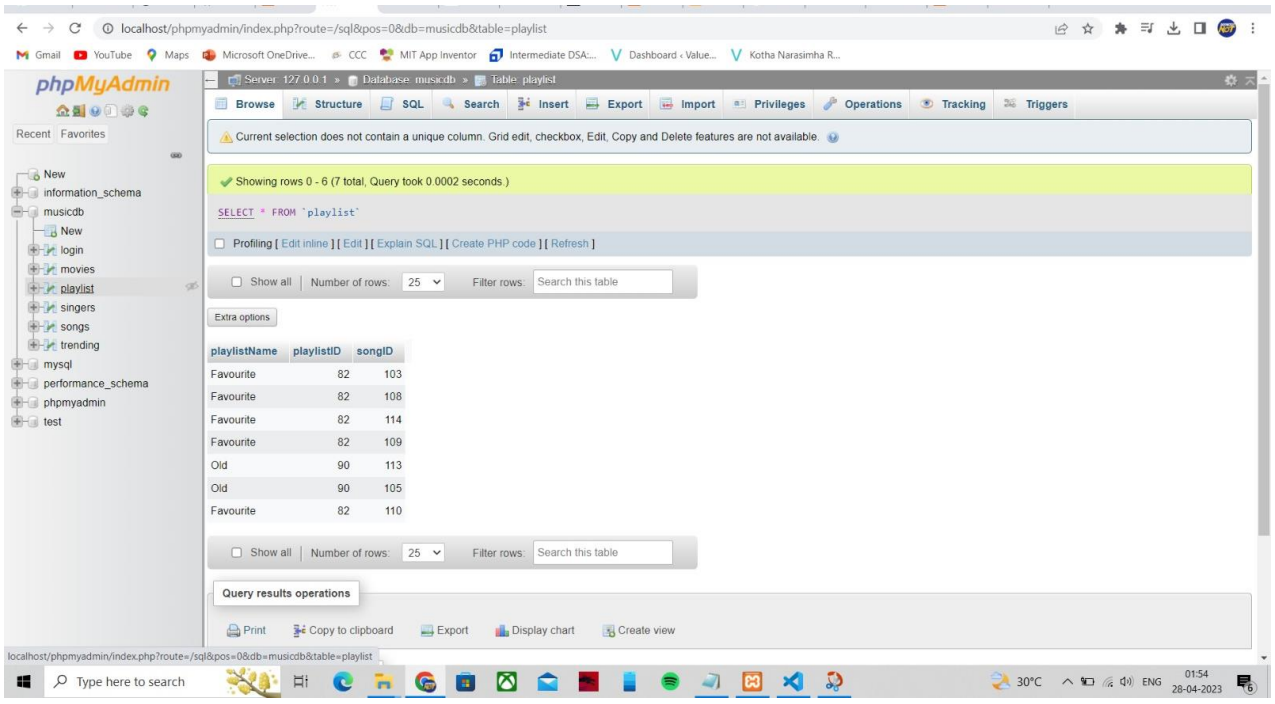



FIG.9: This table shows the details of the playlist

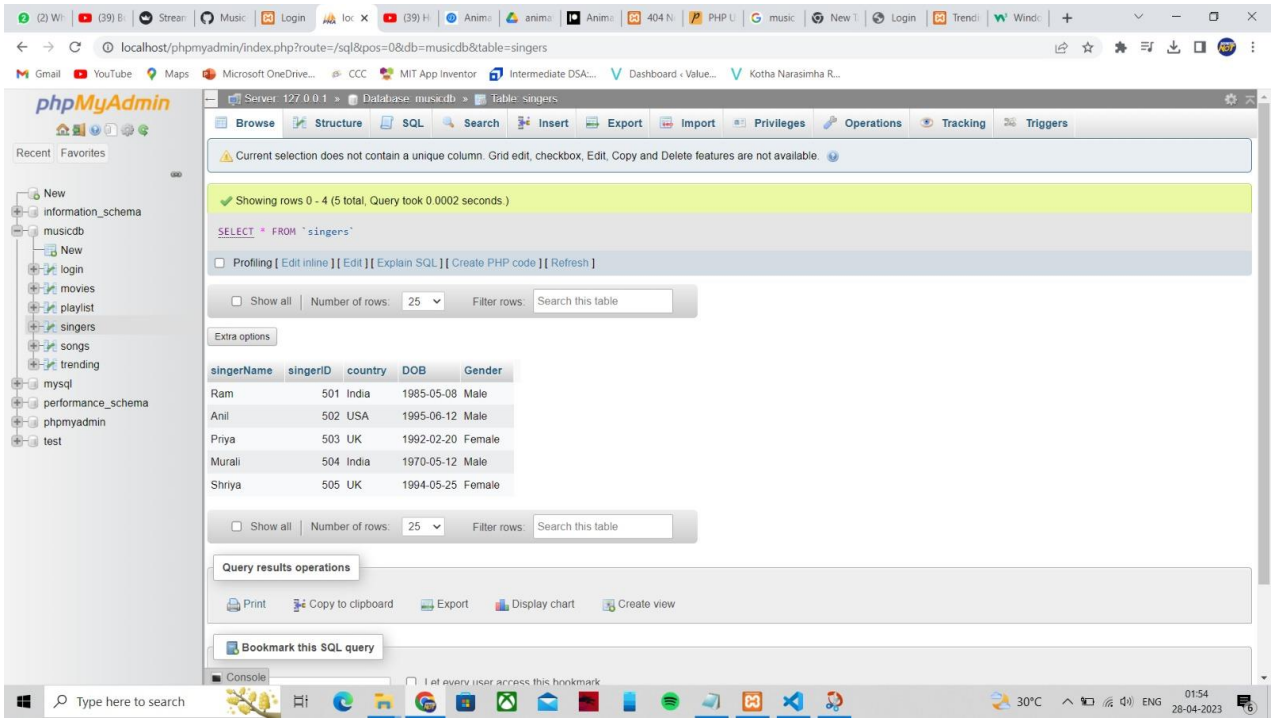



FIG.10: This table shows the details of the singers

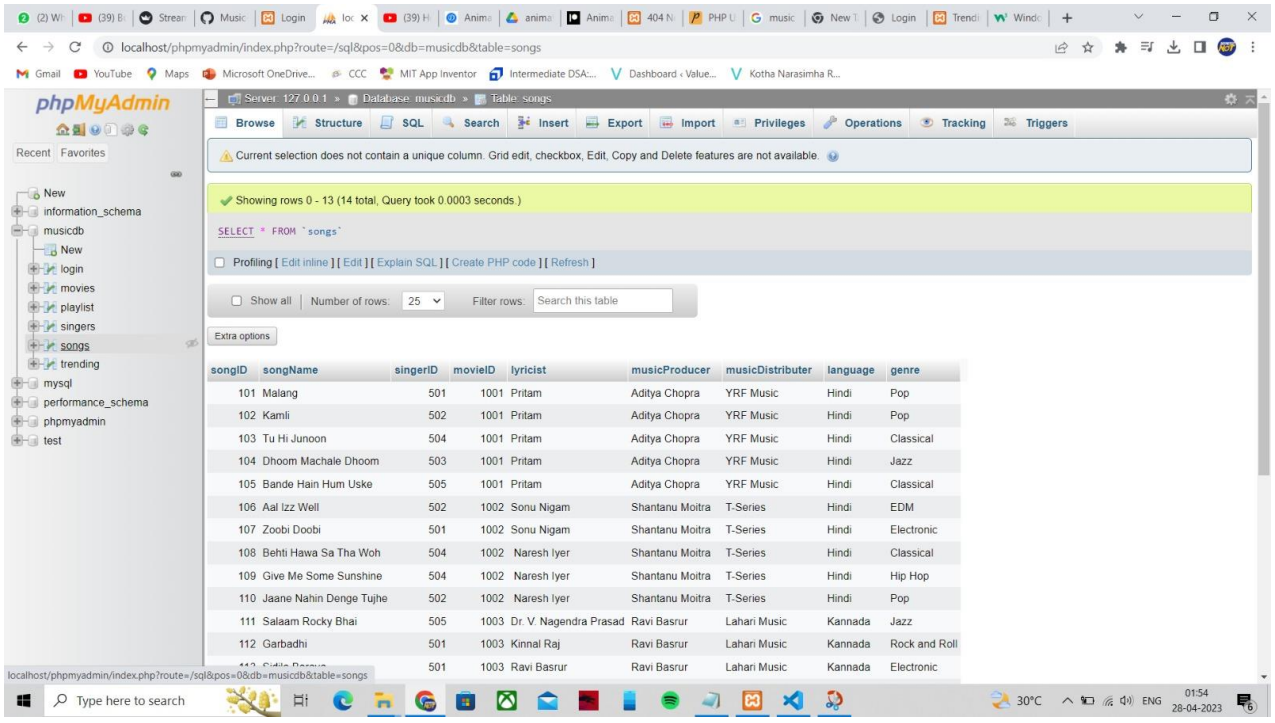



FIG.11: This table shows the details of the songs

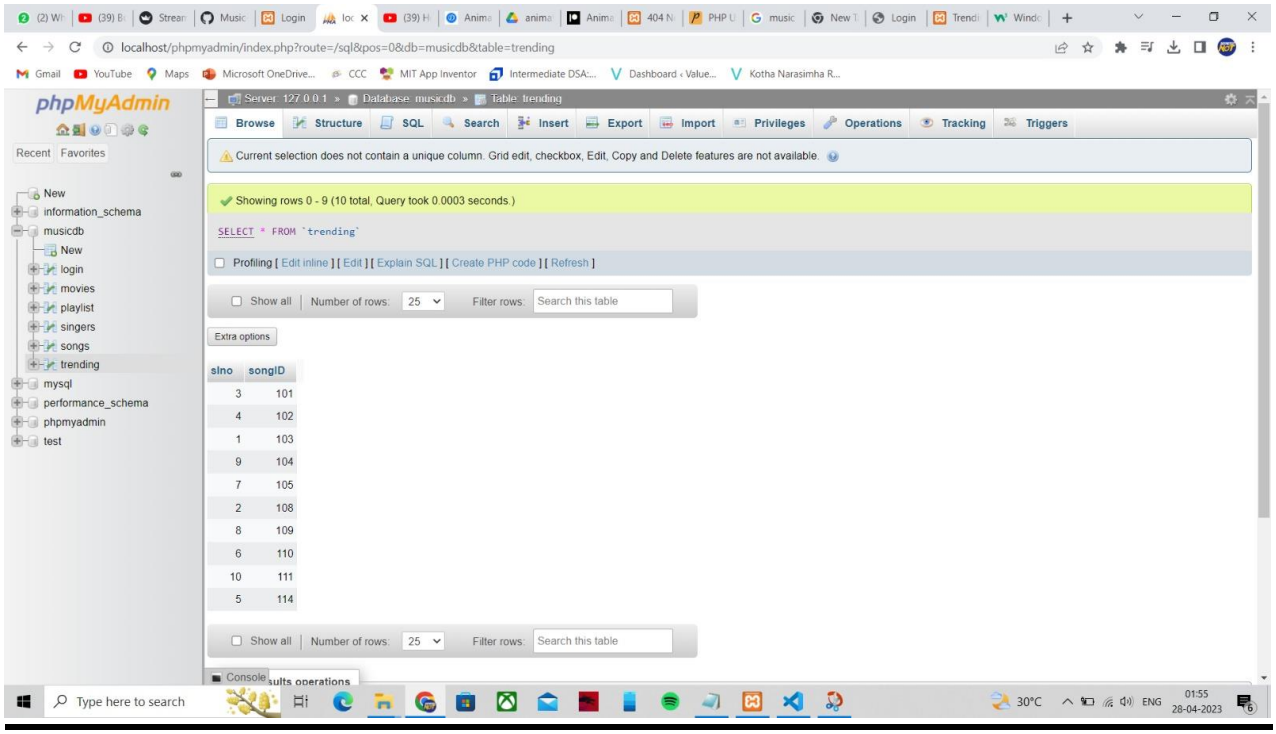



FIG.12: This table shows the details of the songs

We have used the following Back-end languages:

## • PHP MY ADMIN

- PhpMyAdmin is used for connecting the database to the apache server. phpMyAdmin is a free and open-source administration tool for MySQL and MariaDB. As a portable web application written primarily in PHP, it is currently one of the most popular MySQL administration tools, especially for web hosting services.

- PhpMyAdmin offers a range of features, including support for multiple languages, import and export of data in various formats, visual representation of database relationships, and the ability to create and execute SQL queries directly from the interface.

- It also includes a set of security features, such as the ability to restrict access to specific databases and tables, and an audit log that records all actions performed by users.

- PhpMyAdmin is widely used by web developers, system administrators, and database administrators for managing and maintaining MySQL and MariaDB databases. It is available for download from the official website and can be installed on various web servers, including Apache and Nginx.

## • MY SQL

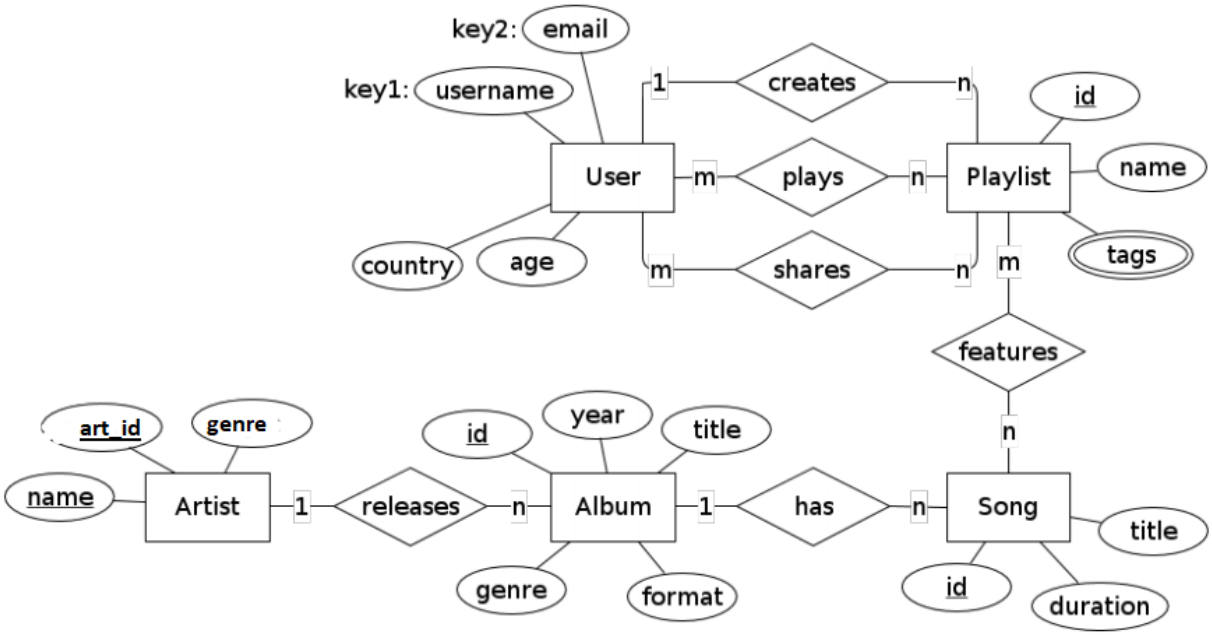
- We have used, in this project, MySQL which is an open-source relational database management system. MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), Facebook, Twitter, Flickr, and YouTube.

- MySQL supports various platforms, including Windows, Linux, and macOS, and provides a range of features, such as support for multiple databases, multi-threading, and indexing for efficient data retrieval. It also includes a set of programming interfaces and tools, such as JDBC, ODBC, and MySQL Workbench, to facilitate application development and database management.

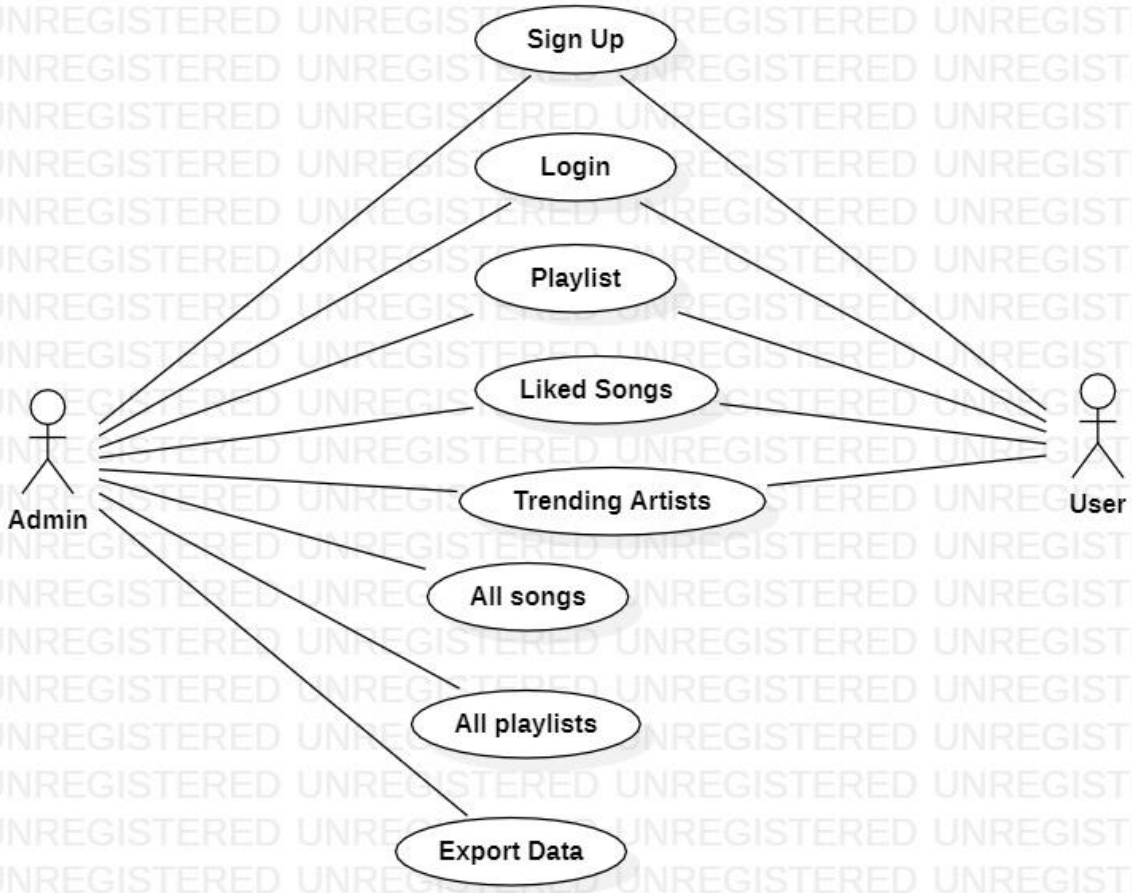
- MySQL uses a structured query language (SQL) for managing data, which allows users to create, modify, and retrieve data in a structured and efficient manner. It also supports transactions and provides various security features, such as encryption and authentication, to ensure the security of data stored in the database.

- MySQL is widely used by web developers, system administrators, and database administrators for managing and maintaining data in various web applications. It is available for download from the official website and can be installed on various web servers, including Apache and Nginx.

## 3.2 ER Diagram



## 3.2 Use Case Diagram



## 4. List of Modules and Functionalities in each of the modules with explanation

Here are the modules of the music management system:-

● **Home Page**: - As soon as the user enters, the first page will be the home page in which there will be three options: user login, user registration and admin login.

● **Sign Up Page**: - User has to register through the sign-up page by entering his/her details such as Name, Date of Birth, Phone number etc. and user has to create his/her own password in this page.

● **Login page**: - This is an authentication page in which it checks for the credentials of the users.

● **Admin Login Page**: - In this page admin has to login with his username and password to further alter the data in the website.

● **Admin Main Page**: - Admin can choose between doctor clinic and logout to which he can add or delete the doctor, or add or delete the clinic.

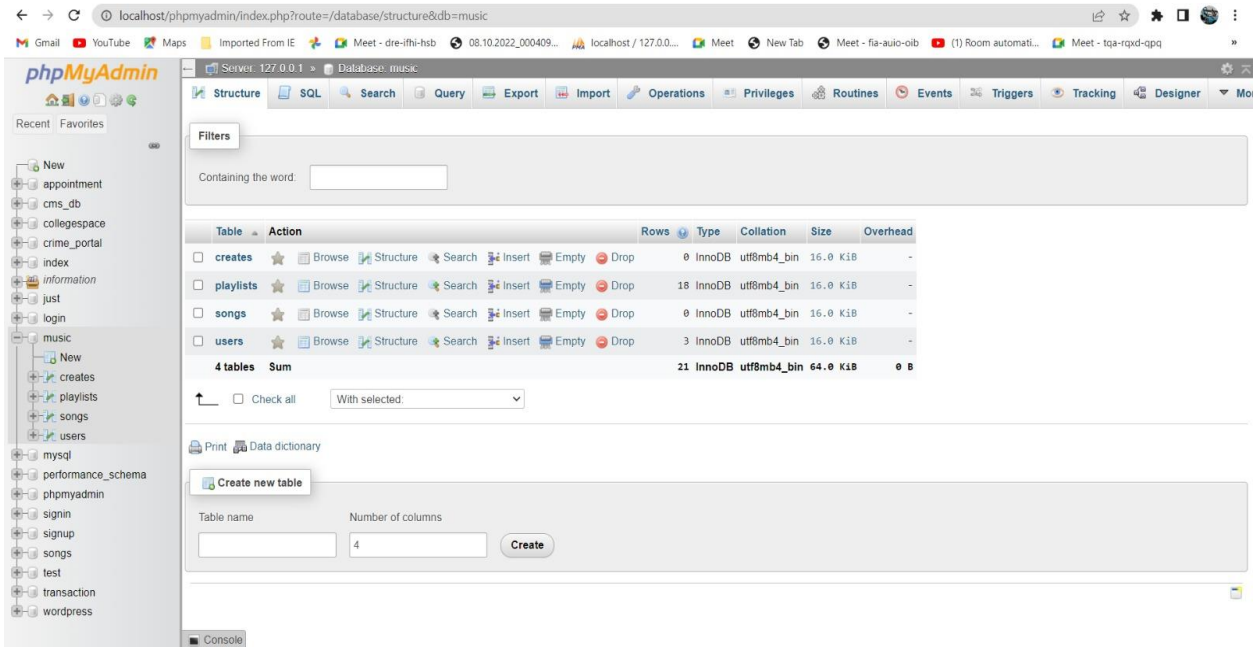
- **Add playlist**: - Here, user can add his/her favourite songs to the playlist.

- **Delete Playlist**: - Here, the user can delete the songs in the playlist at anytime and customizes own playlist

- **Trending artists**: - Here, trending songs can be added and respective artist is displayed.

- **Liked songs**: - Here, the liked songs of user will be displayed/arranged.

# 4.1.Type of Connectivity used for database access with explanation



phpMyAdmin is the most popular application for MySQL database management. We can create, update, drop, alter, delete, import, and export MySQL database tables by using this software. phpMyAdmin also supports a wide range of operation like managing databases, relations, tables, columns, indexes, permissions, and users, etc., on MySQL and MariaDB. These operations can be performed via user interface, while we still have the ability to execute any SQL statement.

phpMyAdmin is translated into 72 languages and also supports both RTL and LTR languages so that the wide range of people can easily use this software. We can run MySQL queries, repair, optimized, check tables, and also execute other database management commands. phpMyAdmin can also be used to perform administrative tasks such as database creation, query execution.

phpMyAdmin is a web-based application used to manage MySQL databases. The type of connectivity used for phpMyAdmin is through the MySQL protocol, which uses the TCP/IP network protocol to establish a connection between the PHP-based web application and the MySQL database server.

When you access phpMyAdmin, it establishes a connection to the MySQL server using the MySQL protocol, and allows you to perform various database management tasks, such as creating and modifying tables, running queries, importing and exporting data, and more.

If we don't have access to a web hosting server and want to learn how to use this application to manage the MySQL database locally, we can install PhpMyAdmin on our PC using third-party 27 products, for example, XAMPP, which is the most popular PHP development environment for Windows, Linux and OS X, a completely free, Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be easy to install and to use. We also have other means of installation, which we can see in the PhpMyAdmin documentation.

Another third-party tool used to install PhpMyAdmin is the WampServer, which is a Windows-only web development environment that allows us to create web applications with Apache2, PHP and a MySQL database and also installs PhpMyAdmin to manage the MySQL database. In my example I have installed this tool to access PhpMy

## 14. List of Database relations

**Table 1 Login:** This table contains the details of user login authentication

| COLUMN NAME | DATATYPE & SIZE | CONSTRAINTS | DESCRIPTION |
|---|---|---|---|
| Fname | Varchar (30) | Not Null | First name of user |
| Lname | Varchar (30) | Not Null | Last name of user |
| email | Varchar (10) | Primary key | Email of user |
| password | Varchar (20) | Not Null | Password set by user |
| DOB | Date | Not Null | Date of birth of user |
| country | Varchar (10) | Not Null | Country of user |
| gender | Number | Not Null | Gender of user |
| User type | Varchar (20) | Not Null | User/admin |
| Subscription period | Number | Not Null | Number of years |
| Subscription Cost | Number | Not null | Cost of subscription |

**Table 2 Songs:** This table contains the details of songs.

| COLUMN NAME | DATATYPE & SIZE | CONSTRAINTS | DESCRIPTION |
|---|---|---|---|
| SongID | Number | Primary Key | ID of song |
| SongName | Varchar (30) | Not Null | Name of song |
| SingerID | Number | Foreign Key | ID of singer |
| MovieId | Number | Foreign Key | ID of movie |
| Lyricist | Varchar (20) | Not Null | Name of lyricist |
| Music Producer | Varchar (20) | Not Null | Name of music producer |
| MusicDistributer | Varchar (20) | Not Null | Name of the music producer |
| Language | Varchar (20) | Not Null | Language of song |
| genre | Varchar (20) | Not Null | Music type |

**Table 3 Singers:** This table contains the details of Singers

| COLOUMN NAME | DATATYPE & SIZE | CONSTRAINTS | DESCRIPTION |
|---|---|---|---|
| singerName | Varchar (20) | Not Null | Name of the singer |
| singerID | Int (11) | Primary key | ID of singer |
| Country | Varchar (20) | Not Null | Country of singer |
| DOB | Time | Not Null | Dob of singer |
| Gender | Varchar (20) | Not Null | Gender of singer |

**Table 4 Movies:** This table contains the details of movies

| COLOUMN NAME | DATATYPE & SIZE | CONSTRAINTS | DESCRIPTION |
|---|---|---|---|
| movieName | Varchar (20) | Not Null | Name of the movie |
| movieId | Int (11) | Primary key | ID of movie |
| Director | Varchar (20) | Not Null | Name of the Director |
| Actor | Varchar(20) | Not Null | Name of the Actor |
| Actress | Varchar (20) | Not Null | Name of the Actress |
| Language | Varchar | Not Null | Language of movie |

**Table 5 Singers:** This table contains the details of Playlists

| COLOUMN NAME | DATATYPE & SIZE | CONSTRAINTS | DESCRIPTION |
|---|---|---|---|
| playlistName | Varchar (20) | Not Null | Name of the playlist |
| playlistId | Int (11) | Primary key | ID of the playlist |
| songID | Int(11) | Foreign Key | ID of the song |

**Table 6 Trendings:** This table contains the details of Singers

| COLOUMN NAME | DATATYPE & SIZE | CONSTRAINTS | DESCRIPTION |
|---|---|---|---|
| Slno | Int(11) | Not Null | Serial number of trending song |
| SongID | Int (11) | Primary key | songID of trending song |

## Normalisation

- Process for evaluating and correcting table structures to minimize data redundancies
- Reduces data anomalies
- Works through a series of stages called normal forms:
  - First normal form (1NF)
  - Second normal form (2NF)
  - Third normal form (3NF)

1 st NF

Database is in first normal form, if it satisfies the following conditions

1. Contains only atomic values.
2. There are no repeating groups.

2 nd NF

A relation schema R is in second normal form, if every non-primary attribute A is fully functionally Dependant on Primary key of R

1. It is in 1NF
2. All non-key attributes are fully functionally dependent on primary key.

3 rd NF

1. It is in 2NF
2. There are no transitive functional dependencies

The above tables satisfy all three normal forms

# 5. Coding

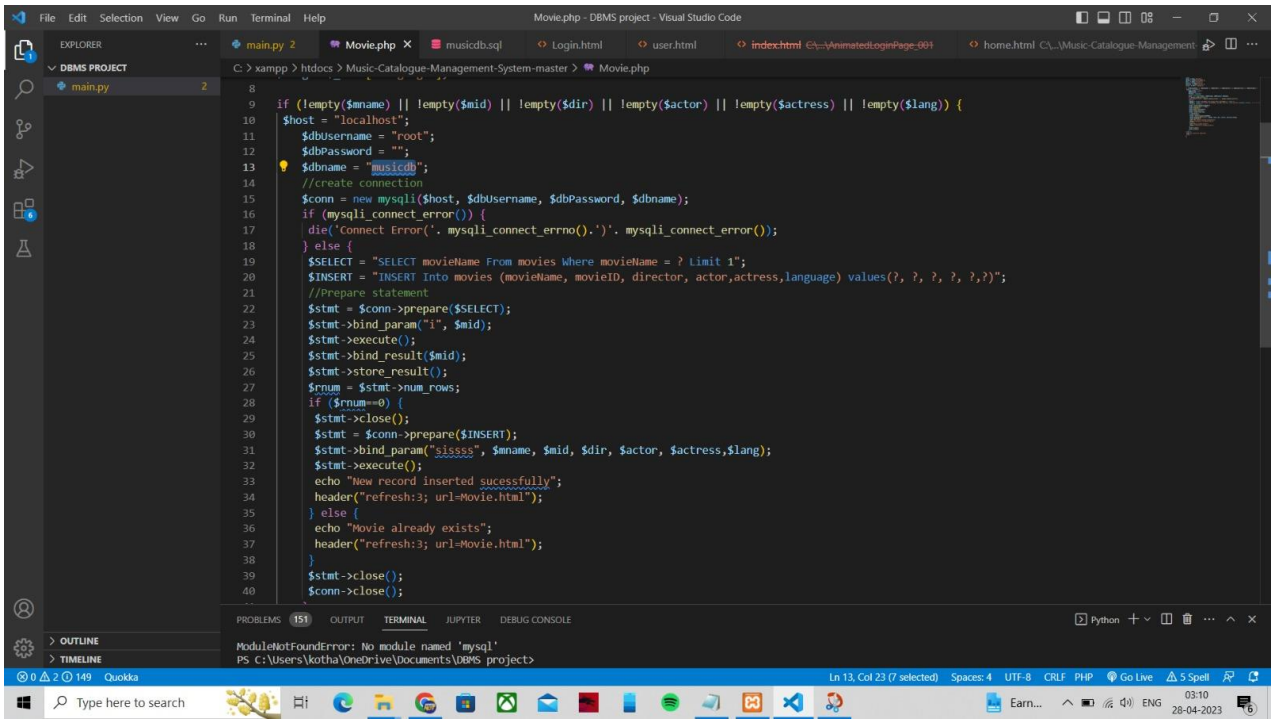



FIG. :Main page

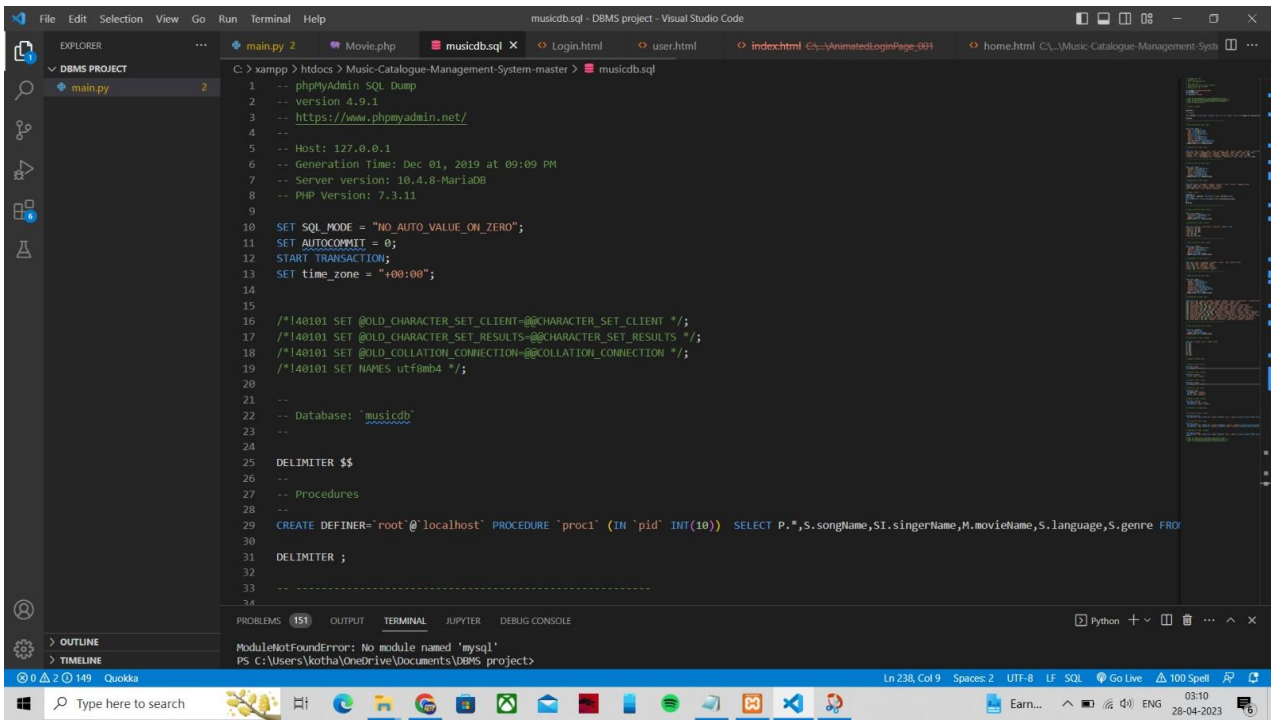



FIG. : Mysql code

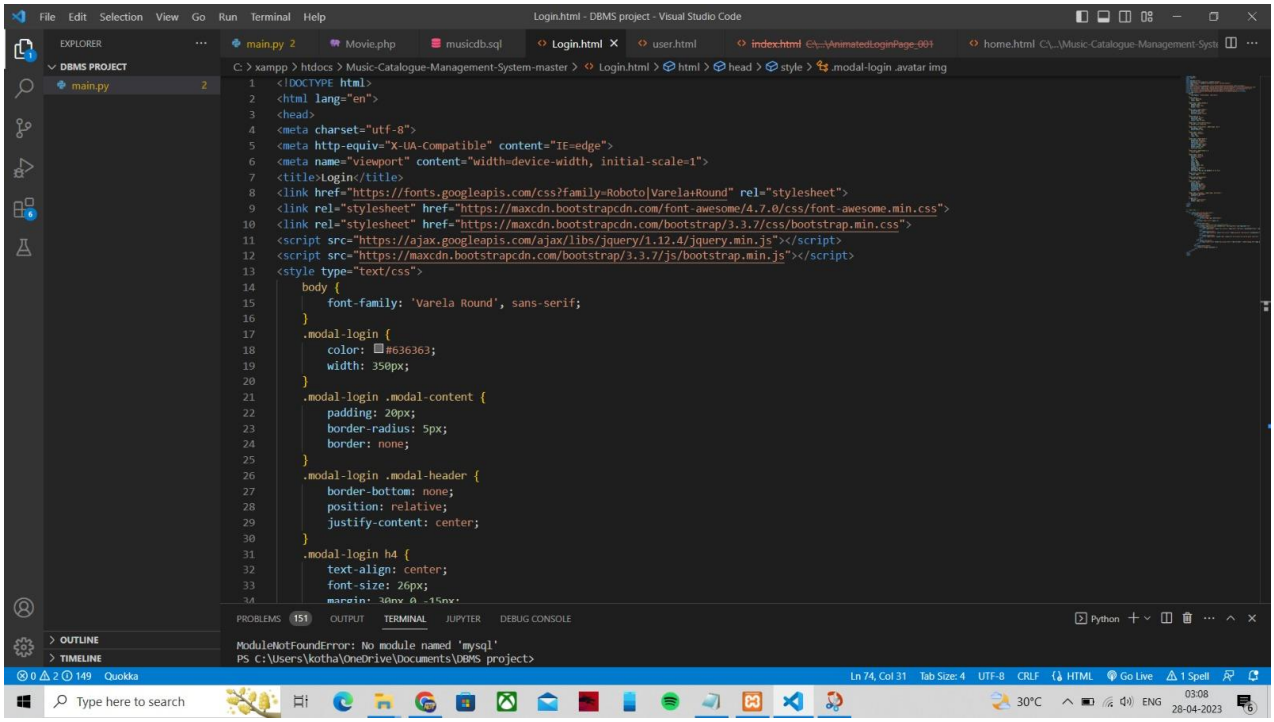



FIG. : HTML code

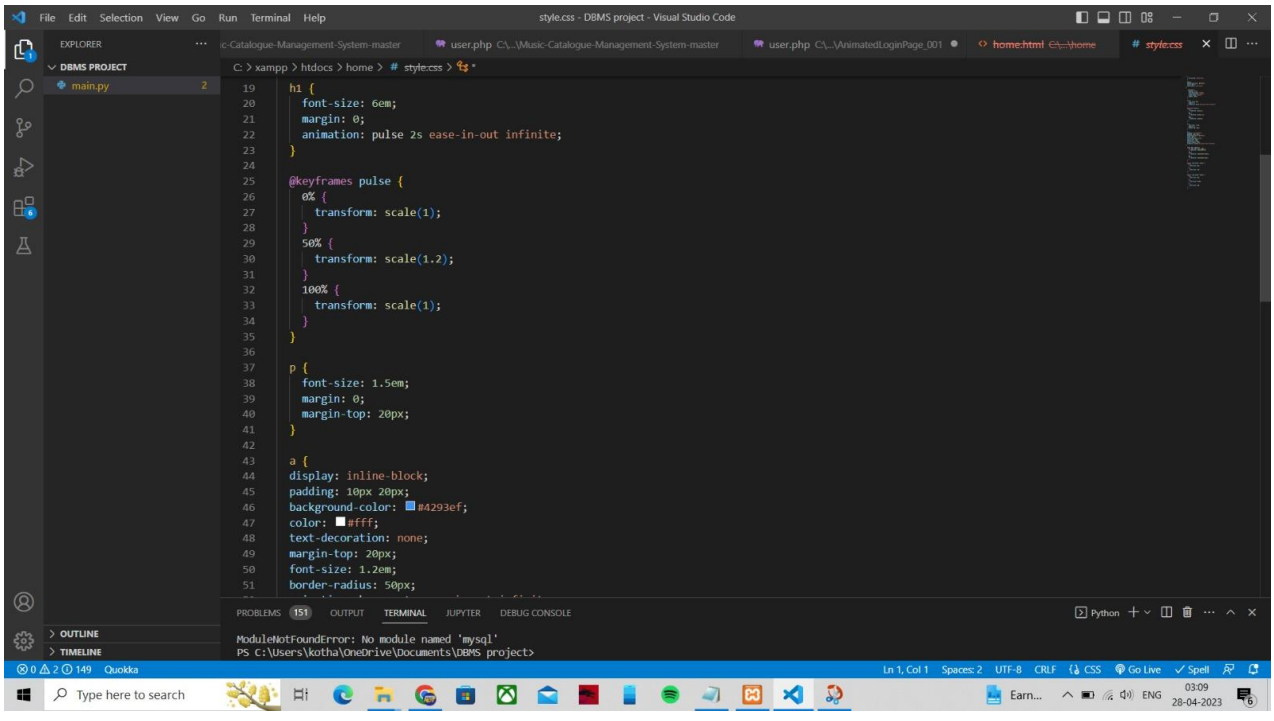



FIG. ; CSS code

# 6.Result and Discussion

The result of a music management system is a well-organized and efficient way of managing music data. With features such as data entry, editing, search, sorting, backup, and unique services such as music recommendation, analysis, playlist management, licensing, and event management, users can easily manage their music data and enhance their music experience.

One of the significant benefits of a music management system is the ability to save time and effort in managing music data. Instead of manually searching and organizing music files, the system can do it automatically, saving users a significant amount of time and effort. Additionally, the unique services provided by the music management system, such as music recommendation and analysis, can help users discover new music and gain insights into their listening habits.

Another benefit of the music management system is the ability to back up music data, ensuring that it is protected from data loss due to system crashes or other issues. This feature provides users with peace of mind, knowing that their music collection is safe and secure.

Overall, the music management system can provide a range of benefits for music enthusiasts and industry professionals alike. By efficiently organizing and managing music data, users can enhance their music experience and discover new music. Additionally, the system can provide valuable insights and analysis into listening habits and music trends, helping industry professionals make informed decisions.

However, as with any system, there may be challenges that need to be addressed. For example, some users may find the system difficult to navigate or may experience technical difficulties. Additionally, there may be limitations in the system's ability to provide accurate music recommendations or insights, depending on the quality and accuracy of the data.

Overall, the benefits of a music management system far outweigh the potential challenges, and the system can provide significant value for users looking to efficiently manage their music data and enhance their music experience.

# 7.Conclusion and Future enhancement

Music Catalogue Management System is an application which allows easy distribution of songs. Users can listen to any songs in a very user-friendly interface. The application also provides users with additional song info like singer name, movie name, lyricist etc. User can also create numerous playlists of their favorite songs. This software also provides a subscription model for users who like to take complete advantage of the system like access to premium content, ad-free listening experience, high quality audio etc. This software reduces the amount of manual data entry and gives greater efficiency. The application can also recommend music based on the listening behavior of the user. The User Interface of it is very friendly and can be easily used by anyone. At the end, this software can perform all the tasks accurately and can do the work for which it is made.

In conclusion, a music management system provides a range of services for organizing and managing music data. It offers features such as data entry, editing, search, sorting, backup, and unique services such as music recommendation, analysis, playlist management, licensing, and event management. These features make it easy for users to efficiently manage their music data and enhance their music experience.

However, there is always room for further enhancement of the music management system. Some future enhancements that could be implemented include:

Integration with emerging technologies: As technology evolves, the music management system could integrate with emerging technologies such as virtual reality or augmented reality to provide users with a more immersive music experience.

Improved data management: The system could be enhanced to provide more sophisticated data management capabilities, such as advanced metadata management, to allow users to manage their music data more efficiently.

Personalized user experience: The system could be enhanced to provide a more personalized experience, such as using machine learning algorithms to recommend music based on a user's listening history and preferences.

Enhanced collaboration: The system could be improved to allow for better collaboration between users, such as the ability to share playlists or work on music projects together.

Integration with other platforms: The music management system could be integrated with other platforms, such as social media or streaming services, to provide users with a more comprehensive music experience.

In summary, a music management system is an important tool for managing music data, and there are many opportunities for further enhancement and improvement. By implementing these future enhancements, the music management system can continue to provide a valuable service to music enthusiasts and industry professionals alike, while also keeping up with the evolving technology landscape.

## 7.References

1] Roger S.Pressman, Software Engineering. A Practitioner Approach, McGraw Hill, 2005
(2) Jim Smith Agile Project Management Creating Innovative Products, Pearson 2008
[3] Walker Royce, Software Project Management, Pearson Education, 1999
[4] Ian Sommerville, Software Engineering, 8th ed., Pearson Education, 2010
[5] Ashfaque Ahmed, Software Project Management A Process-driven Approach, Boca Raton, Florida: CRC Press, 2012
[6] Rajib Mall, Fundamentals of Software Engineering 4th ed., PHI Learning Private Limited 2014