VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI - 590018

Mini Project Report
On

MUSIC STREAMING DATABASE SYSTEM

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering
In
Computer Science and Engineering

Submitted by
ABHISHEK ANANDWADE
1BG18CS002

Vidyaya Amratham Ashnuthe

B.N.M. Institute of Technology

(Approved by AICTE, Affiliated to VTU, Accredited as grade A Institution by NAAC)
All UG branches – CSE,ECE,ISE & Mech.E. accredited by NBA for academic years 2018-19 to 2020-21 & validupto30.06.2021
Post box no. 7087, 27th cross, 12th Main, Banashankari 2nd Stage, Bengaluru-560070, INDIA
Ph: 91-80- 26711780/81/82 Email: principal@bnmit.inwww. bnmit.org

Department of Computer Science and Engineering

2020-2021

B.N.M. Institute of Technology

(Approved by AICTE, Affiliated to VTU, Accredited as grade A Institution by NAAC)
All UG branches – CSE ,ECE, ISE & Mech. E. accredited by NBA for academic years 2018-19 to 2020-21 & valid upto 30.06.2021
Post box no. 7087, 27th cross, 12th Main, Banashankari 2nd Stage, Bengaluru-560070, INDIA
Ph: 91-80- 26711780/81/82 Email: principal@bnmit.inwww. bnmit.org

Department of Computer Science and Engineering

VidyayaAmruthamAshnuthe

CERTIFICATE

Certified that the Mini Project entitled **Music Streaming Database** carried out by Mr. **Abhishek Anandwade** USN **1BG18CS002** a Bonafede student of V Semester B.E., **B.N.M Institute of Technology** in partial fulfillment for the Bachelor of Engineering in COMPUTER SCIENCEAND ENGINEERING of the **Visvesvaraya Technological University**, Belagavi during the year 2020-21. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the report. The Project report has been approved as it satisfies the academic requirements in respect of Database Management System with Mini Project Laboratory prescribed for the said Degree.

Prof. Asha
Assistant Professor & Lab-In charge
Department of CSE
BNMIT, Bengaluru

Dr. Sahana D. Gowda
Professor and HOD
Department of CSE
BNMIT, Bengaluru

Name & Signature

Examiner 1:

Examiner 2:

ABSTRACT

The main objective of a music streaming system is to create a music streaming platform that allows users to search and play songs online based on its title, song Id or song name. The songs are been uploaded to the music database by the admin by using the admin login details. The user can login to the streaming database to stream through the songs. A table is been created to that consists the details of the song i.e. song id, song name, singer name, year. Further the genre of the song can be added to the table along with the song id that will help the user to sort the songs better. The Administrator will have additional functionalities when compared to the common user in order to make the application more user friendly. The main purpose of this system is to allow the system administrator to store details of a large number of songs, allow for add, search, etc. and to separate the functionalities of the system administrator and the user. Thus, the system should be able to satisfy the user requirement. It should be easy to understand and use and hence have a good user interface.

ACKNOWLEDGEMENT

The completion of this project on “Music Streaming Database” has been a good learning experience and I would like to thank all the teachers who have made it possible.

I would like to thank **Shri. Narayan Rao R Maanay**, Secretary, BNMIT, Bengaluru for providing excellent academic environment in the college.

I would like to sincerely thank **Prof. T J Rama Murthy**, Director, BNMIT, Bengaluru for having extended his support and encouragement during the course of the work.

I would like to express my gratitude to **Prof. Eishwar N Maanay**, Dean, BNMIT, Bengaluru for his relentless support, guidance and assistance.

I would like to thank **Dr. Krishnamurthy G N**, Principal, BNMIT, Bengaluru for his constant encouragement.

I would like to thank, **Dr. Sahana D Gowda**, Professor and Head of the Department of Computer Science and Engineering who has shared her opinions and thoughts which helped me in giving my presentation successfully.

I extend my heartfelt gratitude to **Prof. Asha**, Assistant Professor, Department of Computer Science and Engineering, BNMIT, Bengaluru, for completion of the project.

Finally, I would like to thank all the faculty members of Department of Computer Science and Engineering, BNMIT, Bengaluru, for their support. I would like to thank my family and friends for their unfailing moral support and encouragement.

Abhishek Anandwade
1BG18CS002

Table of Contents

CONTENTS **Page No.**

List of Figures

Chapter 1

INTRODUCTION

1.1 Overview of Database Management System

A Database is a collection of related data organized in a way that data can be easily accessed, managed and updated and various operations can be performed on it. By Data, we mean known facts that can be recorded and that have implicit meaning. A DBMS is a collection of programs that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications. DBMS also provides protection and security to database. It maintains data consistency in case of multiple users. Here are some examples of popular DBMS, MySQL, Oracle, Sybase, Microsoft Access and IBM DB2 etc.

The database system can be divided into four components:

- The database system can be divided into System developer and End users.
- Database application: Database application may be Personal, Departmental, Enterprise and Internal
- DBMS: Software that allow users to define, create and manages database access, Ex: MySQL, Oracle etc.
- Database: Collection of logical data.

Functions of database management system:

- Provides Recovery services
- Provides utility
- Provides data Independence
- Provides a clear and logical view of the process that manipulates data

Advantages of DBMS:

- Segregation of application program
- Minimal data duplicity and Easy retrieval of data

1.2 Problem Statement

The aim of this project is to design a Music Streaming Database where customer see the list of songs. The database system contains Songs in stock along with song name, song id, singer id and other information.

The system provides the user with a choice of different songs available for listening. The customer may even see all the songs in the database.

1.3 Objectives

- Online music streaming database enables the user to search for a particular song and play it.
- A user can browse through multiple songs and can add songs to his favorite's list so that all the songs liked by the user are at one place.
- The system has songs with multiple genres that a user can select from.
- Every song has a singer that a user can sort the songs by. User can get all the songs by a singer at one place so that the user has good experience.
- All the songs belong to their respective albums hence, the system enables the user to surf through multiple albums.
- The Administrator will have additional functionality when compared to the common user. He can add, delete and update the songs in the database, so that the user can have a good experience.

1.4 Dataset Description

Given below are the entities along with its attributes and relations present in the database of this application that are used to retrieve information from the database as per requirement of user.

- An entity type **'SONG'** with attributes where song_id, song_name, singer_id, movie_album_id, Year where song_id is the primary key attribute.

- An entity type **‘SINGER’** with attributes singer_id and singer_name where singer_id is the primary key attribute.
- An entity type **‘MOVIE_ALBUM’** with attributes song_id, movie_album_id, movie_album_name, actor_name, actress_name where movie_album_id is the primary key attribute and song_id is the foreign key attribute referencing from **‘SONG’** table. This entity has a strong relationship with **‘SONG’** entity.
- An entity type **‘GENRE’** with attribute song_id and genre which has song_id as the foreign key attribute.
- An entity type **‘PLAYLIST’** with attribute song_id, singer_id where song_id acts as the foreign key attribute.

Chapter 2

SYSTEM REQUIREMENTS

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as (computer) system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time.

2.1 Software Requirements

• Operating System	:	Windows 7 or higher
• Front End	:	HTML5/CSS/JavaScript
• Back end	:	MySQL/PHP
• Server	:	XAMPP
• Browser	:	Mozilla Firefox

2.2 Hardware Requirements

• Processor	:	Pentium III or higher
• RAM	:	2GB or above
• Hard Disk	:	20 GB or More

Chapter 3

SYSTEM DESIGN

3.1 Entity Relationship Diagram

An Entity Relationship Diagram describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between instances of those entity types. In software engineering, an ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure which can be implemented in a database, typically a relational database.

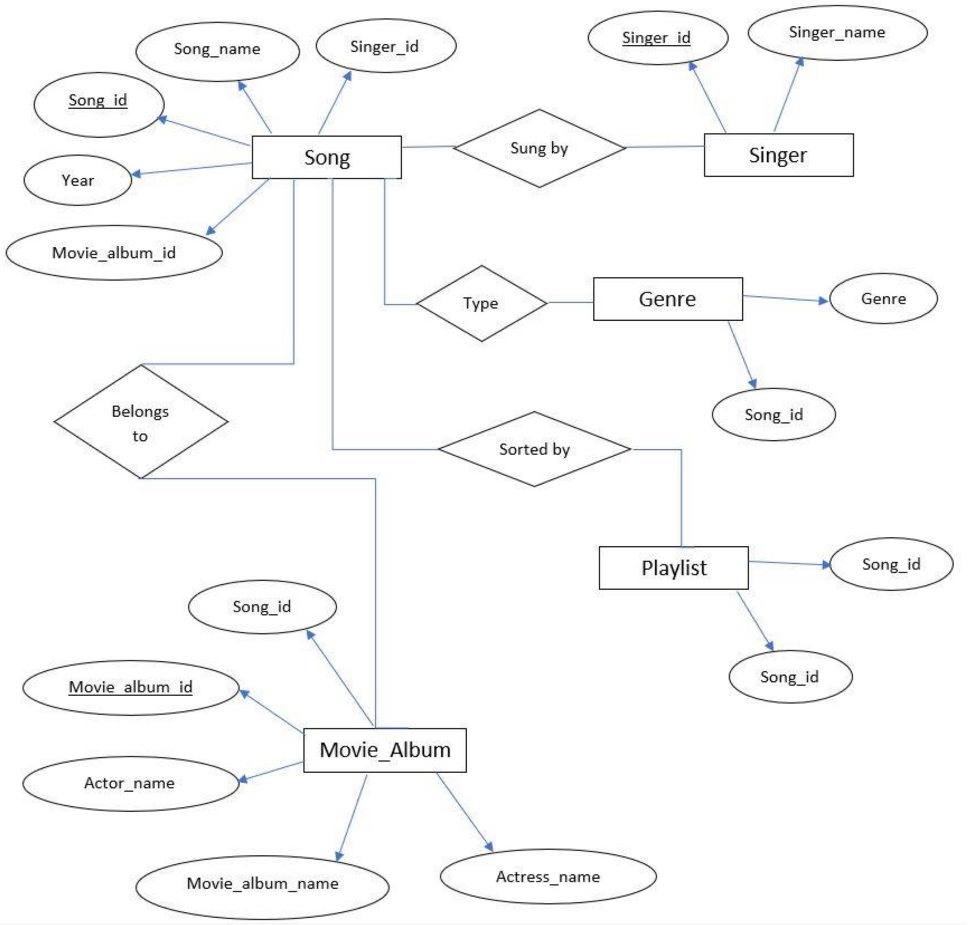


Figure 3.1: E R diagram of Music Streaming Database System.

3.2 Schema Diagram

A database schema is the skeleton structure that represents the logical view of the entire database. It formulates all the constraints that are to be applied on the data. A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagram.

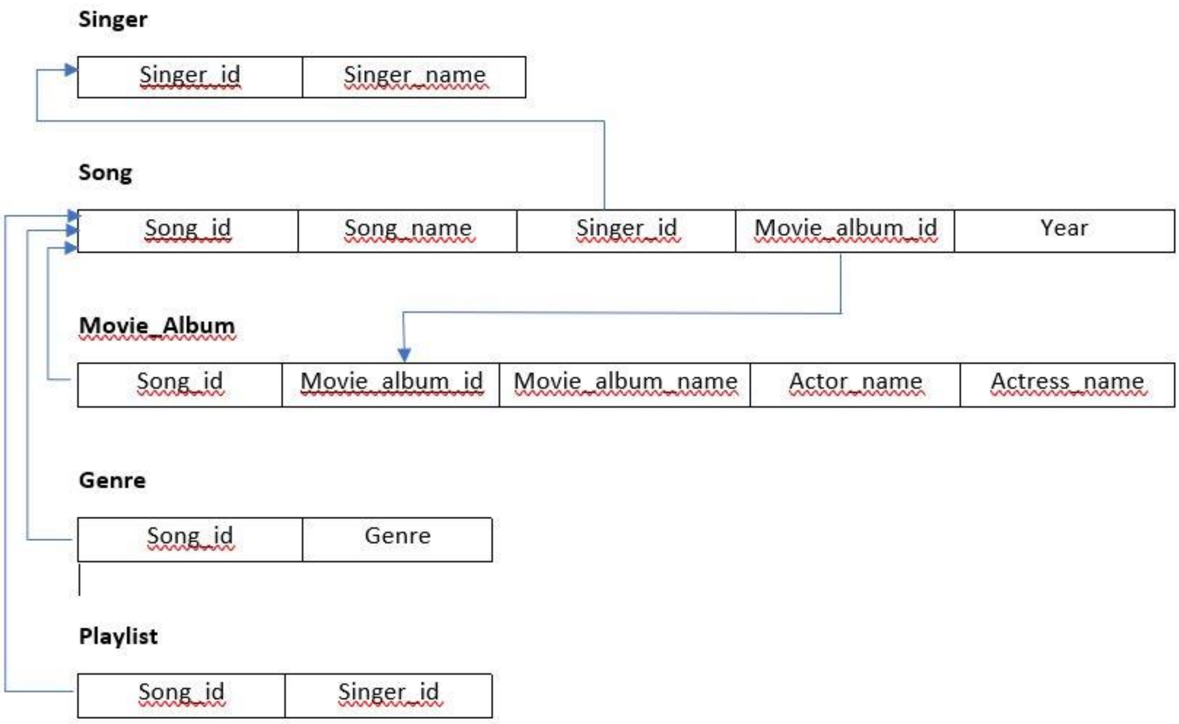


Figure 3.2: Schema diagram of Music Streaming Database System

3.3 Overview of Graphical User Interface

GUI is program interface that takes advantages of the computer's graphics capabilities to make the program easier to user. Well-designed graphical user interfaces can free the user from learning complex command language. On the other hand, many users find that they work more efficiently with a command-driven interface, especially if they already know the command language.

Hypertext Markup Language (HTML) is the standard markup 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.

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup 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 and is applicable to rendering in speech, or on other media. 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.

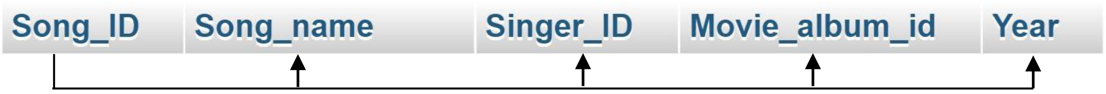
Hypertext Preprocessor (PHP) is a general-purpose programming language originally designed for web development. PHP code may be executed with a command line interface (CLI), embedded into HTML code, or used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in a web server or as a Common Gateway Interface (CGI) executable. The web server outputs the results of the interpreted and executed PHP code, which may be any type of data, such as generated HTML code or binary image data. PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic drone control.

3.4 Normalization

Normalization rule are divided into following normal form.

1. First Normal Form
2. Second Normal Form
3. Third Normal Form

SONG table



It is in 1NF as it does not have any composite or multivalued attributes.

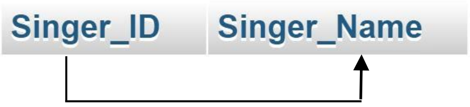
Customer’s cart is multivalued; hence it is decomposed to a separate relation.

It is in 2NF as it is fully functionally dependent.

(song_id) → (song_name, singer_id, movie_album_id, Year)

As the relation does not contain a non-key attribute functionally determining other non-key attributes, it is in 3NF (there is no transitivity in attributes).

SINGER table



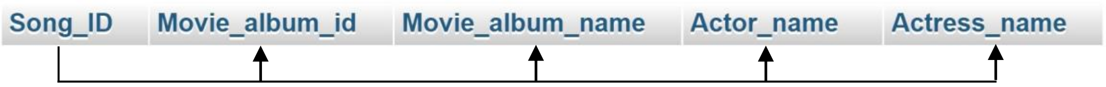
It is in 1NF as it does not have any composite or multivalued attributes.

It is in 2NF as it is fully functionally dependent.

(singer_id) → (singer_name)

As the relation does not contain a non-key attribute functionally determining other non-key attributes, it is in 3NF(there is no transitivity in attributes).

MOVIE_ALBUM table



It is in 1NF as it does not have any composite or multivalued attributes.

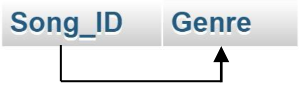
It is in 2NF as it is fully functionally dependent.

(song_id) → (movie_album_id, movie_album_name, actor_name, actress_name)

(movie_album_id) → (movie_album_name, actor_name, actress_name)

As the relation does not contain a non-key attribute functionally determining other non-key attributes, it is in 3NF(there is no transitivity in attributes).

GENRE table



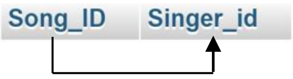
It is in 1NF as it does not have any composite or multivalued attributes.

It is in 2NF as it is fully functionally dependent.

(song_id) → (Genre)

As the relation does not contain a non-key attribute functionally determining other non-key attributes, it is in 3NF(there is no transitivity in attributes).

PLAYLIST table



It is in 1NF as it does not have any composite or multivalued attributes.

It is in 2NF as it is fully functionally dependent.

(song_id) → (singer_id)

As the relation does not contain a non-key attribute functionally determining other non-key attributes, it is in 3NF (there is no transitivity in attributes).

Chapter 4

IMPLEMENTATION

4.1 Table Creation

ADMIN

```
create table admin
(
user varchar(10) primary key,
pass varchar(10) not null
);
```

USER

```
create table user
(
user varchar(10) primary key,
pass varchar(10) not null
);
```

SONG

```
create table song
(
Song_id varchar(20) primary key,
Song_name varchar(20) not null,
Singer_id varchar(20) not null,
Movie_album_id varchar(20) not null,
Year int not null
);
```

MOVIE ALBUM

```
create table movie_album
(
movie_album_id varchar(20) primary key,
movie_album_name varchar(20) not null,
actor_name varchar(20) not null,
actress_name varchar(20) not null
);
```

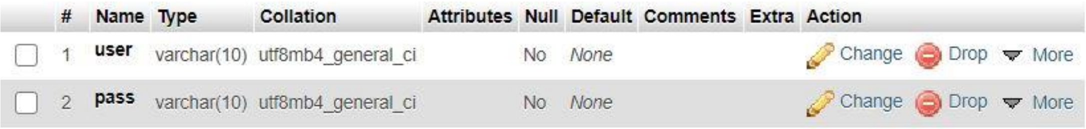
GENRE

```
create table genre
(
Song_id varchar(20) on delete cascade,
Genre varchar(20) not null
);
```

PLAYLIST

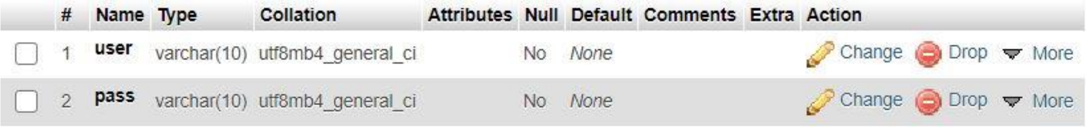
```
create table playlist
(
Song_id varchar(20) on delete cascade,
Singer_id varchar(20) not null
);
```

4.2 Description of Tables



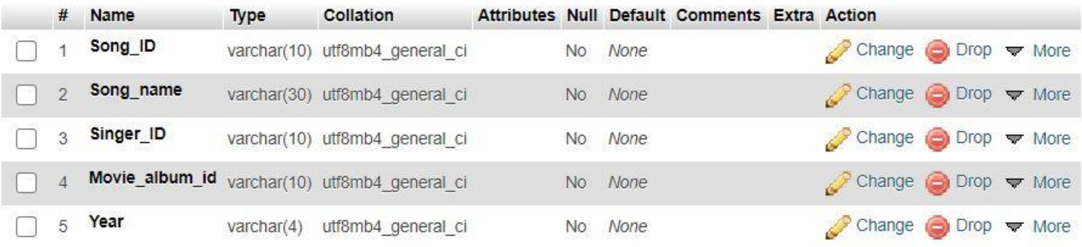
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	user	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
□	2	pass	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More

Fig 4.1 – Admin Description



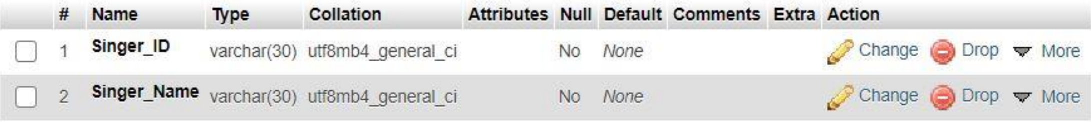
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	user	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
□	2	pass	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More

Fig 4.2 – User Description



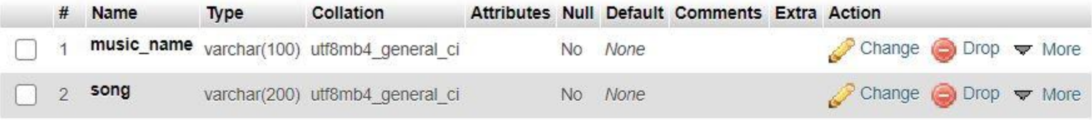
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	Song_ID	varchar(10)	utf8mb4_general_ci		No	*None*			Change Drop More
□	2	Song_name	varchar(30)	utf8mb4_general_ci		No	*None*			Change Drop More
□	3	Singer_ID	varchar(10)	utf8mb4_general_ci		No	*None*			Change Drop More
□	4	Movie_album_id	varchar(10)	utf8mb4_general_ci		No	*None*			Change Drop More
□	5	Year	varchar(4)	utf8mb4_general_ci		No	*None*			Change Drop More

Fig 4.3 – Song Description



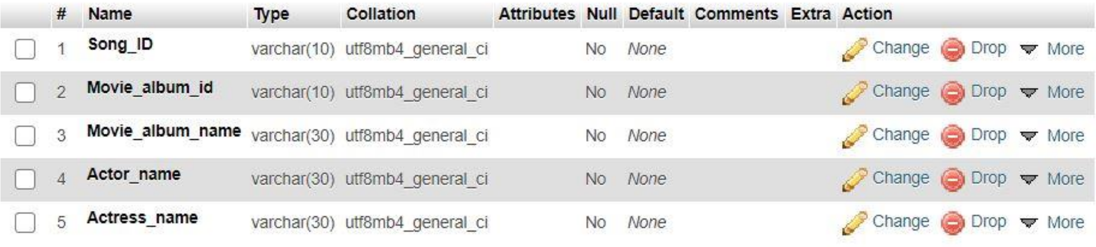
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	Singer_ID	varchar(30)	utf8mb4_general_ci		No	*None*			Change Drop More
□	2	Singer_Name	varchar(30)	utf8mb4_general_ci		No	*None*			Change Drop More

Fig 4.4 – Singer Description



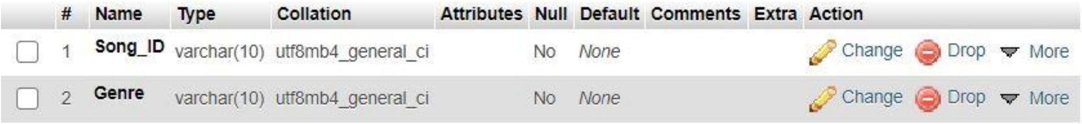
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	music_name	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop More
□	2	song	varchar(200)	utf8mb4_general_ci		No	*None*			Change Drop More

Fig 4.5 – Music Description



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	Song_ID	varchar(10)	utf8mb4_general_ci		No	*None*			Change Drop More
□	2	Movie_album_id	varchar(10)	utf8mb4_general_ci		No	*None*			Change Drop More
□	3	Movie_album_name	varchar(30)	utf8mb4_general_ci		No	*None*			Change Drop More
□	4	Actor_name	varchar(30)	utf8mb4_general_ci		No	*None*			Change Drop More
□	5	Actress_name	varchar(30)	utf8mb4_general_ci		No	*None*			Change Drop More

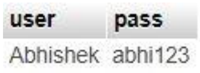
Fig 4.6 – Album Description



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	Song_ID	varchar(10)	utf8mb4_general_ci		No	*None*			Change Drop More
□	2	Genre	varchar(10)	utf8mb4_general_ci		No	*None*			Change Drop More

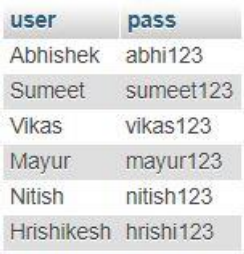
Fig 4.7 – Genre

4.3 Populated Tables



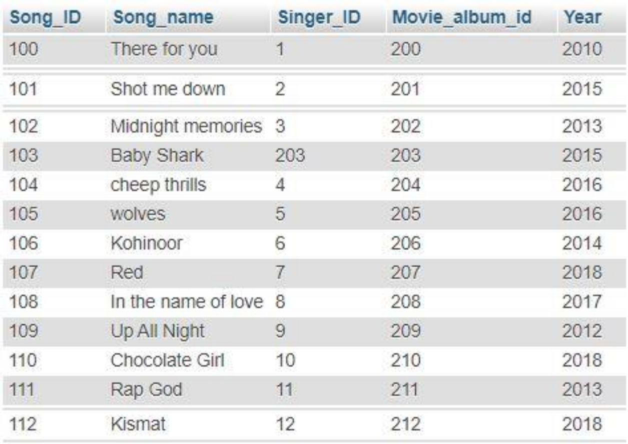
user	pass
Abhishek	abhi123

Fig 4.8 – Admin Table



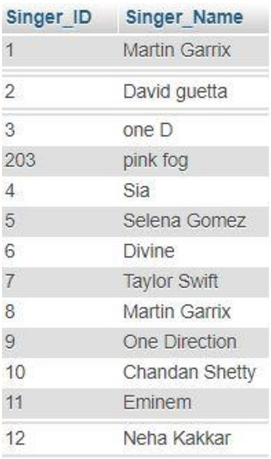
user	pass
Abhishek	abhi123
Sumeet	sumeet123
Vikas	vikas123
Mayur	mayur123
Nitish	nitish123
Hrishikesh	hrishi123

Fig 4.9 – User Table



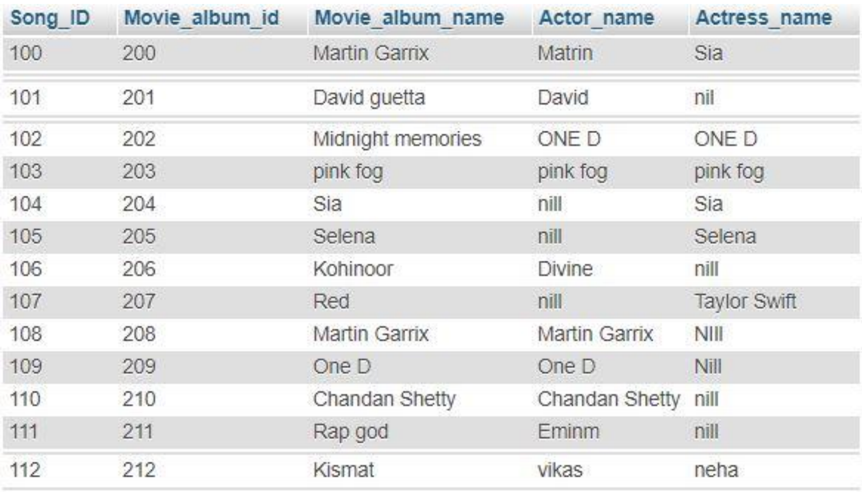
Song_ID	Song_name	Singer_ID	Movie_album_id	Year
100	There for you	1	200	2010
101	Shot me down	2	201	2015
102	Midnight memories	3	202	2013
103	Baby Shark	203	203	2015
104	cheep thrills	4	204	2016
105	wolves	5	205	2016
106	Kohinoor	6	206	2014
107	Red	7	207	2018
108	In the name of love	8	208	2017
109	Up All Night	9	209	2012
110	Chocolate Girl	10	210	2018
111	Rap God	11	211	2013
112	Kismat	12	212	2018

Fig 4.10 – Song Table



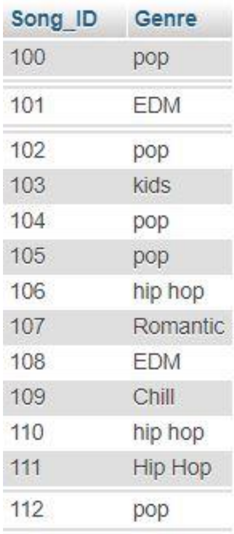
Singer_ID	Singer_Name
1	Martin Garrix
2	David guetta
3	one D
203	pink fog
4	Sia
5	Selena Gomez
6	Divine
7	Taylor Swift
8	Martin Garrix
9	One Direction
10	Chandan Shetty
11	Eminem
12	Neha Kakkar

Fig 4.11 – Singer Table



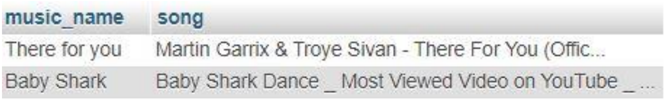
Song_ID	Movie_album_id	Movie_album_name	Actor_name	Actress_name
100	200	Martin Garrix	Matrin	Sia
101	201	David guetta	David	nil
102	202	Midnight memories	ONE D	ONE D
103	203	pink fog	pink fog	pink fog
104	204	Sia	nill	Sia
105	205	Selena	nill	Selena
106	206	Kohinoor	Divine	nill
107	207	Red	nill	Taylor Swift
108	208	Martin Garrix	Martin Garrix	NIll
109	209	One D	One D	Nill
110	210	Chandan Shetty	Chandan Shetty	nill
111	211	Rap god	Eminm	nill
112	212	Kismat	vikas	neha

Fig 4.12 – Album Table



Song_ID	Genre
100	pop
101	EDM
102	pop
103	kids
104	pop
105	pop
106	hip hop
107	Romantic
108	EDM
109	Chill
110	hip hop
111	Hip Hop
112	pop

Fig 4.13 – Genre Table



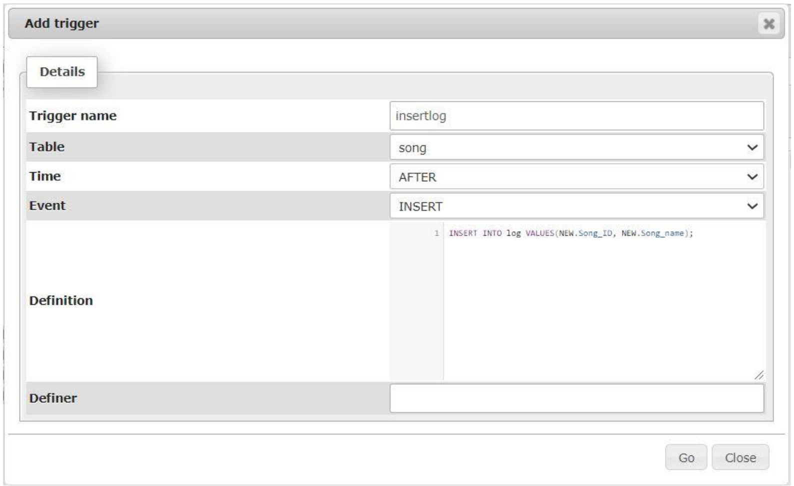
music_name	song
There for you	Martin Garrix & Troye Sivan - There For You (Offic...
Baby Shark	Baby Shark Dance _ Most Viewed Video on YouTube _ ...

Fig 4.14 – Music Table mp3

4.4 SQL Triggers & Stored Procedures

Triggers

A database trigger is procedural code that is automatically executed in response to certain events on a table or view in a database. The trigger is mostly used for maintaining the integrity of the information on the database.



Add trigger

Details

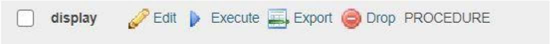
Trigger name	insertlog
Table	song
Time	AFTER
Event	INSERT
Definition	`INSERT INTO log VALUES(NEW.Song_ID, NEW.Song_name);`
Definer	

Fig 4.15 – Trigger

Stored Procedure

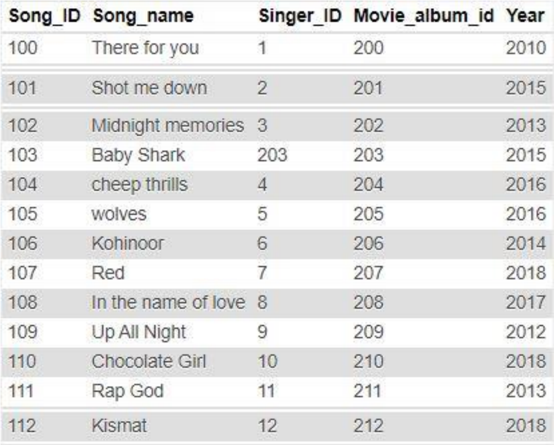
A stored procedure is a set of Structured Query Language (SQL) statements with an assigned name, which are stored in a relational database management system as a group, so it can be reused and shared by multiple programs.

Code for Stored Procedure is as follows:



□ display Edit Execute Export Drop PROCEDURE

Fig 4.16 – Stored Procedure (Display Songs)



Song_ID	Song_name	Singer_ID	Movie_album_id	Year
100	There for you	1	200	2010
101	Shot me down	2	201	2015
102	Midnight memories	3	202	2013
103	Baby Shark	203	203	2015
104	cheep thrills	4	204	2016
105	wolves	5	205	2016
106	Kohinoor	6	206	2014
107	Red	7	207	2018
108	In the name of love	8	208	2017
109	Up All Night	9	209	2012
110	Chocolate Girl	10	210	2018
111	Rap God	11	211	2013
112	Kismat	12	212	2018

Fig 4.17 – Stored Procedure (Result)

4.5 Database Connectivity

In computer science, a database connection is the means by which a database server and its client software communicate with each other. The term is used whether the client and the server are on different machines. The client uses a database connection to send commands to and receive replies from the server. A database is stored as a file or a set of files on magnetic disk or tape, optical disk, or some other secondary storage device. The information in these files may be broken down into records, each of which consists of one or more fields. The connection to XAMPP database is obtained by the following code snippet:

```
<?php

 $servername = "localhost";

 $username = "root";
 $password = "";
 $dbname = "music_libraray";
$conn = mysqli_connect($servername,$username,$password,$dbname);

?>
```

Chapter 5

RESULTS

Login page for user

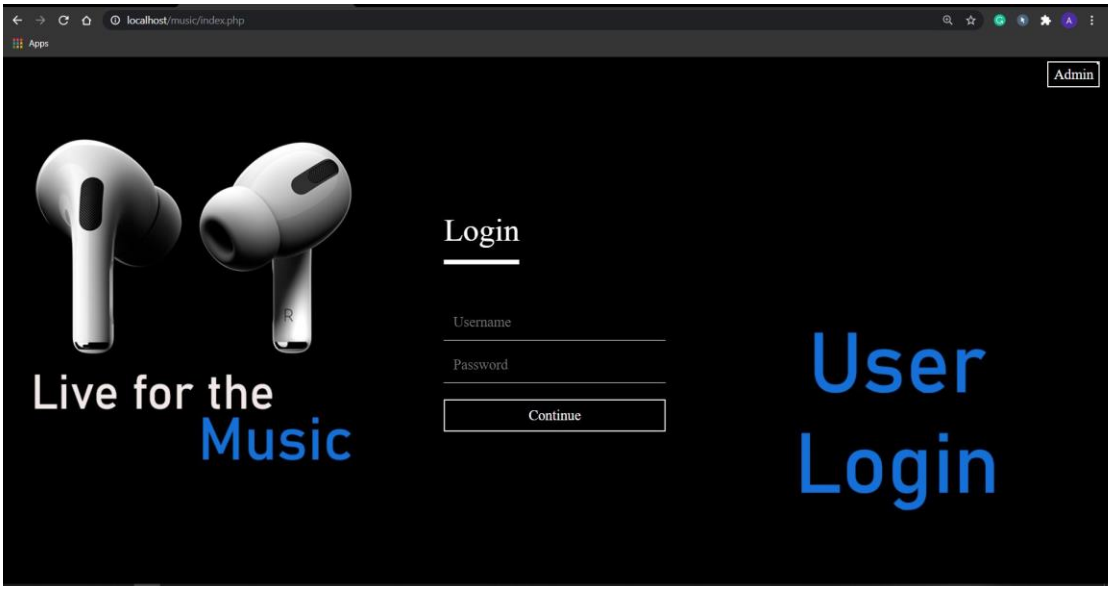


Figure 5.1: User Login page.

Login Page for Admin

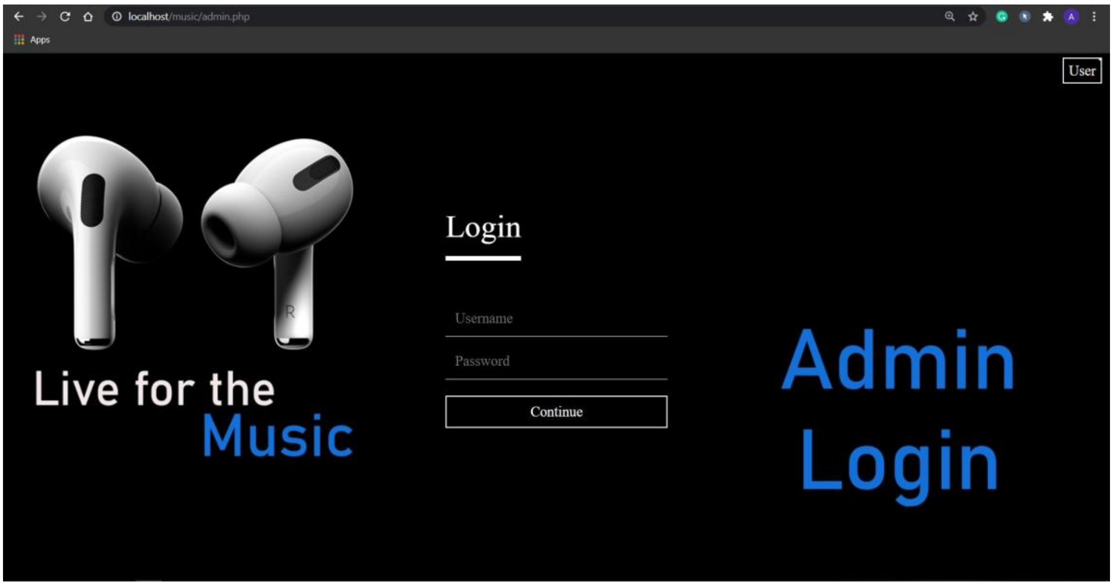


Figure 5.2: Admin Login Page.

User dashboard page to select songs

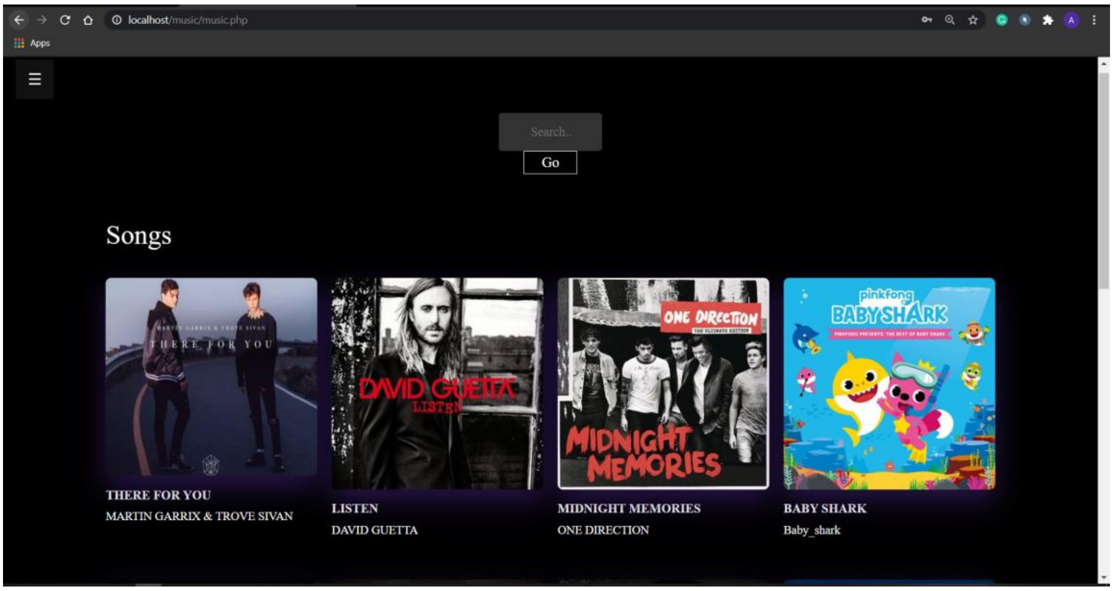


Figure 5.3: User Dashboard Page.

Menu options for user

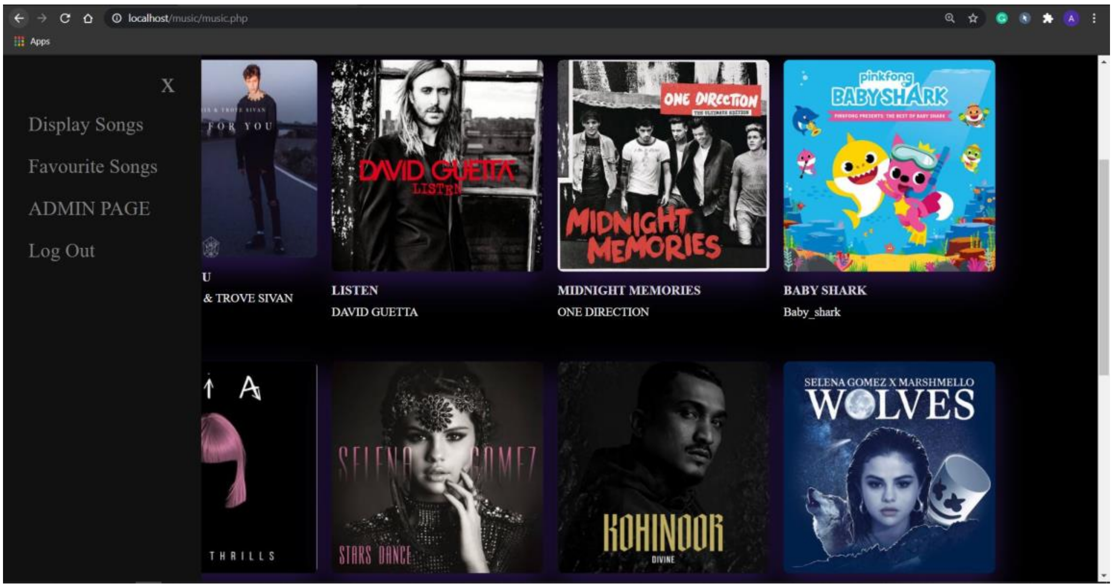


Figure 5.4: Multiple options.

Searching a song by its Title

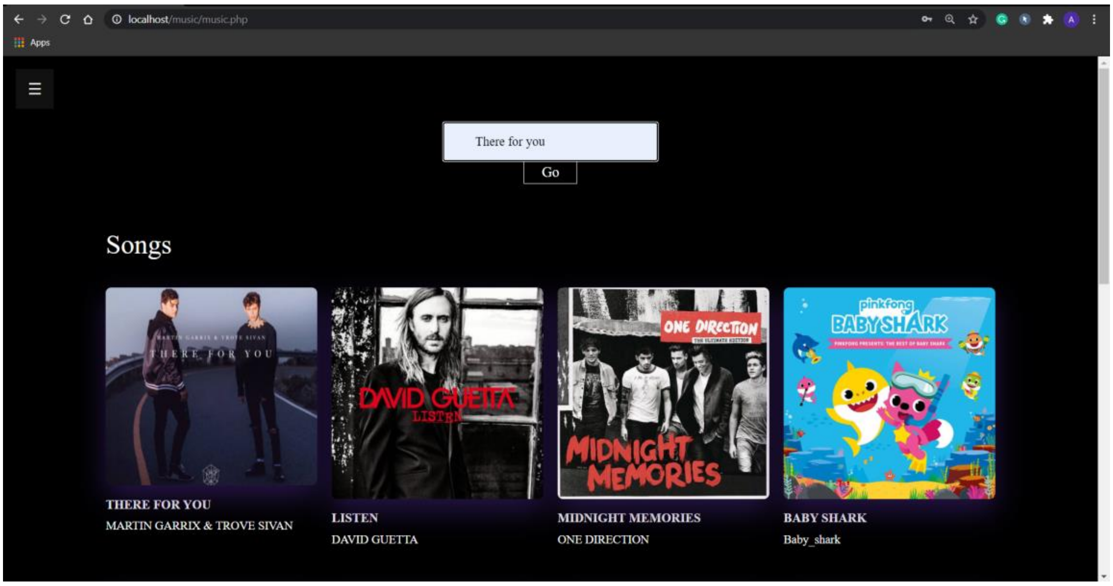


Figure 5.5: Searching a song.

Results of search

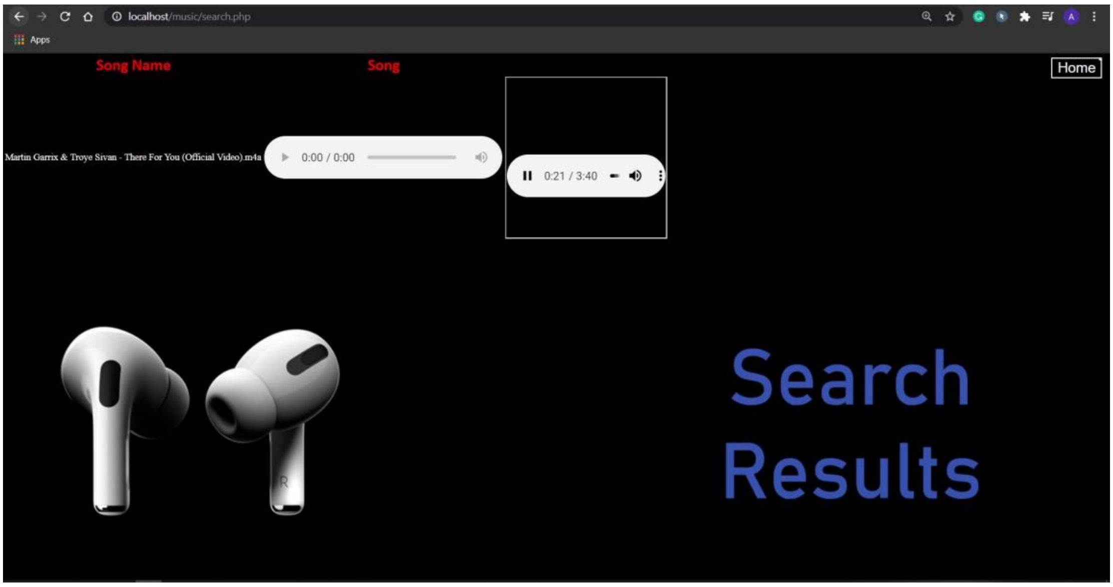
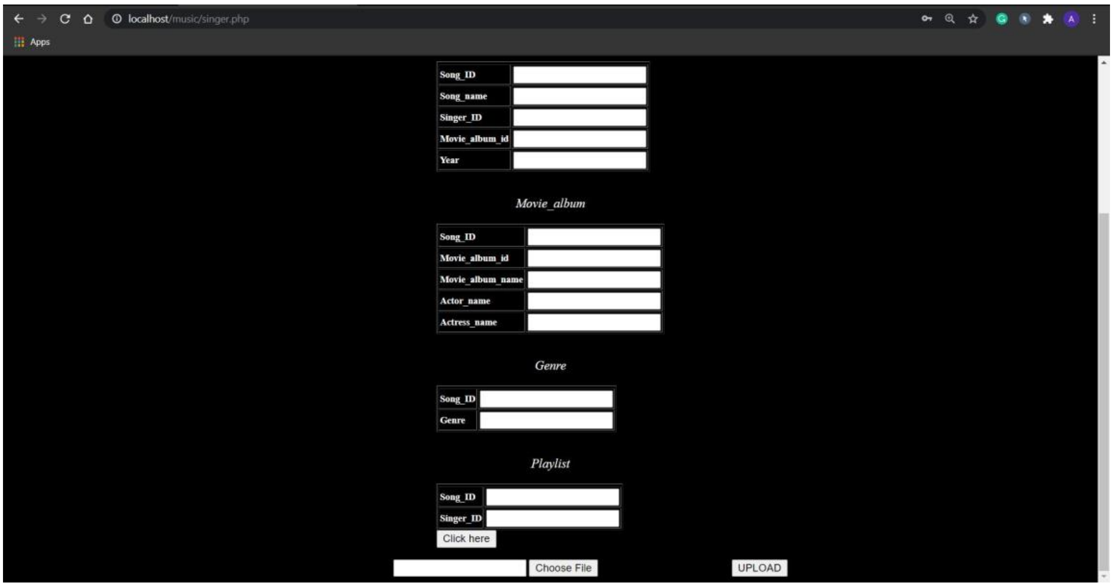

localhost/music/search.php
Apps
Song Name
Song
Home
Martin Garrix & Troye Sivan - There For You (Official Video).m4a
0:00 / 0:00
0:21 / 3:40
Search Results

Figure 5.6: Search Results

Data input by Admin.



Song_ID
Song_name
Singer_ID
Movie_album_id
Year

Movie_album

Song_ID
Movie_album_id
Movie_album_name
Actor_name
Actress_name

Genre

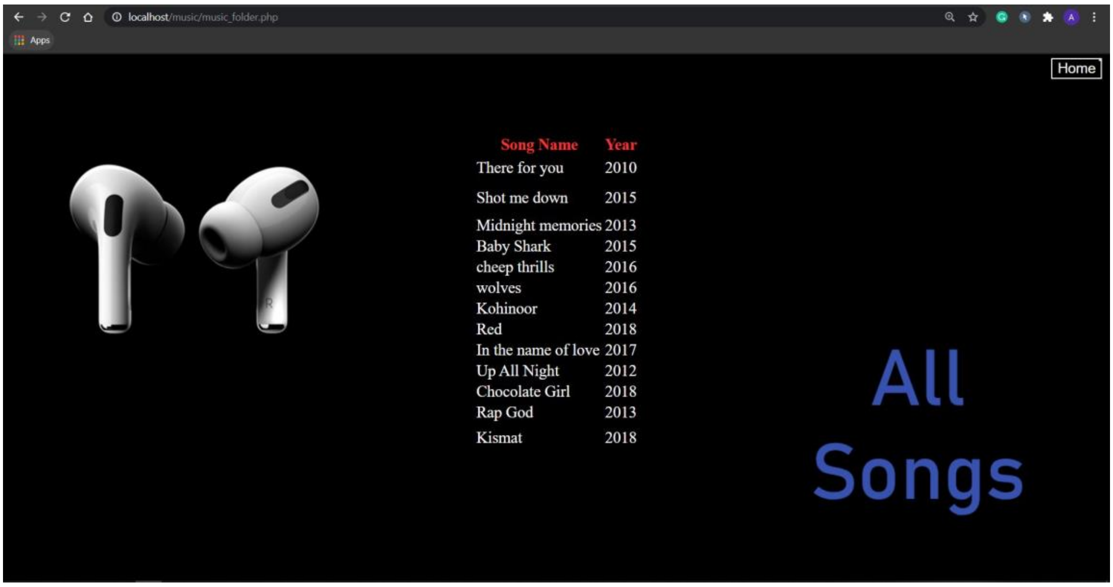
Song_ID
Genre

Playlist

Song_ID
Singer_ID

Figure 5.7: Admin Portal.

Display of all songs in the database



Song Name	Year
There for you	2010
Shot me down	2015
Midnight memories	2013
Baby Shark	2015
cheep thrills	2016
wolves	2016
Kohinoor	2014
Red	2018
In the name of love	2017
Up All Night	2012
Chocolate Girl	2018
Rap God	2013
Kismat	2018

Figure 5.8: All Songs Page.

CONCLUSION

The Music Streaming Database project is designed to meet the requirements of a listening music online. The main aim of maintaining the, music data has been achieved in this system. It has been developed in PHP, HTML, CSS keeping in mind the specification of the system.

For designing the system, PHP is being used as the front end in phpMyAdmin Environment. Overall the project teaches us the essential skills like:

- Understanding the database handling and query processing
- Implement, analyze and evaluate the project developed for an application
- Demonstrate the working of different concepts of DBMS

FUTURE ENHANCEMENT

This project can be used in the real-world music streaming websites after adding some more useful modules in the project for which the respected company is providing services.

Nowadays music is used on many occasions, this project can also be improved with the improvement in those domains.

Additional features that can be added to this project in future are:

- Different online payment gateways can be added to felicitate online transaction.
- Songs can be sorted based on the mood and occasion.
- An option can be added where user can upload their songs in public domain.
- Ever user can have personalized experience with AI based song recommendation.

Utmost care and back-up procedures must be established to ensure 100% successful implementation of the Music Streaming Database system. In case of system failure, the organization should be able to process the transaction with another organization and in case of failure; it should be able to complete the process manually.