Project Report on

MUSIC STREAMING SERVICE

Submitted By

Sayan Mukherjee (12022002016076), AIML-58

Shivyansh Rai (12022002016059), AIML-41

Under the supervision and guidance of

Mr (Dr.) Deepsubhra Guha Roy

ASSOCIATE PROFESSOR IEM

&

Ms Bipasha Mahato

ASSISTANT PROFESSOR IEM

In Partial Fulfilment of the Requirements for the Course Project in

Database Management Systems (DBMS)

Batch: 2022-2026

Submitted To

Department of Computer Science and Engineering (Artificial Intelligence & Machine Learning)

Institute of Engineering & Management

SECTOR– V, KOLKATA -700091

DECLARATION

We, Shivyansh Rai and Sayan Mukherjee, students of the Department of Artificial Intelligence and Machine Learning (AIML), Batch 2022-2026, at the Institute of Engineering and Management, Kolkata, hereby declare that the work presented in this DBMS Lab Project report on "Music Streaming Service" is an original work completed under the guidance of Prof. Dr. Deepsubhra Guha Roy and Ms. Bipasha Mahato.

We affirm that this project is a partial fulfilment of the requirements for the Database Management Systems course and has not been submitted for any other degree or diploma at this or any other institution.

STUDENT SIGNATURE

Name: Sayan Mukherjee

Name: Shivyansh Rai

ABSTRACT

This project introduces a Music Streaming Service to enable users to search for songs from catalogs, compile their playlists, as well as listen and discover new content on the basis of personal preferences. For the front end, I have used HTML/CSS/JAVASCRIPT and for back end I used MySQL Our website intends to make the users closer to music by offering them a unique interactive means.

The functionalities that makes this service stand out includes; search for music library, creating as well as managing playlists, artists' info, and history-based suggestion. Some of the data stored in a database include songs and artists' information, consumers' playlist, and playback history to achieve enhanced data access for streaming.

The expected result of this project is the ability to comfortably surf through this website to find and play music of the users' choice. Besides technical interconnectivity of the front-end and back-end frameworks and principles, this project explores how data could be leveraged to improve User Engagement.

ACKNOWLEDGEMENT

We are grateful to Prof. Dr. Deepsubhra Guha Roy, Ms. Bipasha Mahato for their encouragement and providing us the chance in this project. The completion of the project has become possible through the kind assistance and the helpful suggestions of their domain.

To the Institute of Engineering and Management, Kolkata, India and AIML Department: special thanks to the management and faculty who provided all the necessary facilities needed in the successful implementation of this project.

We are really grateful to all those people who contributed to the creation of this project, it was really interesting.

Sayan Mukherjee

Shivyansh Rai

Table of Content

Chapter 1

This chapter gives the basic introduction of the project. It deals with objectives, Over-view and problem statement. It gives the basic outline of the entire project and provide the details about the problem statement.

Chapter 2

This chapter includes the literature survey. Literature survey involves the study of various reference papers. It gives the basic idea of what new is needed in the existing system.

Chapter 3

This chapter mainly deals with the scope of the project. It gives the detailed information about the webpage. It also includes the software and hardware requirements of the project.

Chapter 4

This chapter deals with the Theoretical Analysis/project details of the project. It includes codes being used in the project.

Chapter 5

This chapter contains the methodology of the project.

Chapter 6

This chapter contains the results of the project.

Chapter 7

This chapter includes the conclusion of the project. It also contains the future work which can be implemented to increase the efficiency and to add new features from the project.

Chapter 8

This chapter contains references.

CHAPTER: 1
INTRODUCTION

1.1 Overview

Our project, a music streaming website, is the site where users will be enjoying the online music with no disruption and discovering new music with just a few clicks. As a component course in Our Database Management Systems (DBMS) program, this platform applies the latest HTML/ CSS, PHP, and MySQL web development techniques to implement a fully working web application which stores, retrieves, and streams music.

The main operations of the website are related to user registration and login, the song search, creating playlists, and real-time audio streaming. Using a simple and attractive web interface made with HTML and CSS, PHP is used as the server-side language to regulate the application logic, while MySQL serves as the database for user, song, playlist, and playing history information.

The project enabled us to study and implement the key principles of database management, web development, and user interface design thereby producing a realistic, complete application that mirrors the necessary operations of famous music streaming services.

1.2 Problem Definition

The demand for accessible and enjoyable music streaming has grown dramatically, thus a platform that offers users an easy interface to locate, arrange, and enjoy music is needed. Some music lovers encounter difficulties when they have to store a large number of songs online or access them offline without downloading files. Also, custom options such as playlists, a search button, and simple navigation are important to make it a good experience for the listeners.

Through this platform, users can interact with the application in the following ways, thus the scope of the project includes the following design functions:

Using the app which has the vast collection of music tracks and the users don’t have to worry about the storage problems.

Find a certain song or artist quickly and with ease.

Create and maintain a list of personalized tracks.

Seamless music in real-time without occupying extra memory.

Efficiently handle the identities of the users and utilize the data acquired from their interactions with the web technologies to save the users' preferences and usage history.

The website is a combination of substantial web technologies and database management, thus developing a user-friendly and efficient music streaming platform, a solution that is a fair balance between accessibility, user interaction, as well as safe data handling is thus achieved.

1.3 Overview Of Report

This report projects our music streaming website that we have developed, as part of our DBMS coursework. The report starts with the introduction of the motivation and purpose of the platform, then a problem definition section which includes some of the specific issues targeted such as easy access to music, playlist management, and real-time streaming.
The System Design section exhibits the platform’s architecture, which comes from HTML, CSS, PHP, and MySQL linking to the efficient function. The Implementation section emphasizes fundamental technical factors such as user authentication, music search, and playlist functions.
The Testing and Evaluation section recaps the tests that were carried out to make sure that it is functional and performing well and the Conclusion part displays the project outcomes, challenges, and future enhancement possibilities.
The report is very informative about our strategy for developing one of the most user-friendly and effective music streaming services in the world.

1.3 Objective

One of the main objectives of this project is to create a fully functioning music streaming website that will allow users to easily access a catalog of music tracks online. This platform is designed to let users register, search for songs, create personalized playlists, and of course, stream music without interruption. By this project, we want to:
Implement a simple user interface for music search and background music.
Design a secure backend that can be used to store the user's data and preferences.
Use database management principles to manage music libraries as well as users' interactions in a very efficient manner.
Enabling real-time streaming and reducing the time spent buffering to foster a light-weight user experience.

This project is a practical application of web development and database integration and a complete solution to stream music online.

1.5 Organization Of Report

This report is well-ordered and presents a comprehensive overview of the music streaming website project:
Introduction – Tells about the origin, causes, and aims of the project.
Problem Definition – Enumerates the particular problems, our project plans to tackle.
System Design – Lays down the architectural configuration, database structure, and technology stack, plus the flexibility about HTML, CSS, PHP, and MySQL.
Implementation – Specifies the primary modules. For instance, a user can log in to the system, search for songs, and also store them in a playlist and stream those songs.
Testing and Evaluation – Brief results have been shown to validate the proper working, security, and performance, the results and modifications are also mentioned.
Conclusion – Thinking about the knowledge gained, the obstacles encountered, and the further paths we have for improvement.

Each part is giving valuable information about our strategy of forming a complete and friendly user music streaming platform.

CHAPTER: 2
LITRATURE REVIEW

The literature review rests on the foundation of analyzing music streaming services from the perspective of their evolution as well as their design, with a focus on the aspects of user experience, backend management, and streaming efficiency. The analysis data depicts the fact that Spotify, Apple Music, and the like are capable of high user engagement because of the personalized recommendations they provide, which are widely based on the use of machine learning algorithms. Research on real-time streaming brought attention to the fact that data compression and buffering need to be highly optimized so that the viewer can have a nice experience on any connection.

The function of databases is also critical to running music services. The technology must be able to manage high volumes of data with little latency. Studies in database solutions like MySQL or NoSQL shed light on fast data extraction and space-saving techniques which are important for expandable music services. Furthermore, in design, the interface plays a vital role in how intuitive navigation and user search features are directly influential on user satisfaction.

We have successfully implemented vital features for music discovery, playlist creation, and playback on our project by adopting the methodologies from these studies and applying the conventional practices of music streaming technology.

2.1 Introduction

The literature review begins with an overview of music streaming technology trends and an examination of commercial music streaming platforms in order to identify essential functionality driving user engagement and operational cost. Such a review is essential since many contemporary music services are built upon similar types of technologies (e.g., Spotify, Apple Music, YouTube Music).

The rest of the review covers recent research on several underlying technologies related to our design and implementation—namely, user authentication, database management, media streaming, and interface design. Specifically, we focus on large-scale database optimization for music libraries [1], real-time audio/music streaming techniques [2], [3], and user-centered design principles [4]. We acquired useful insights from those studies that further guided our platform design for efficiency/ scalability/user experi- ence.

This review will inform the development of our platform to be in line with leading music services and to identify where we can innovate.

2.1 existing system

Current music streaming platforms, namely, Spotify, Apple Music, and YouTube Music have revolutionized the accessibility of music. They provide users with an interface to explore a plethora of music, make playlists and receive song recommendations based on listening trends.

Such platforms possess several features including user authentication, real-time streaming, personalized playlist and robust backend systems for fast data retrieval which use large-scale databases but designs are mostly proprietary inhibiting customization or adaptability for smaller projects. A feature analysis allowed us to identify key features and shortcoming that influenced the design philosphy of our self-contained music streaming website; simplicity of design and ease of accessiblity accompanied by core music functionality.

CHAPTER: 3
SYSTEM DESCRIPTION

3.1 Introduction

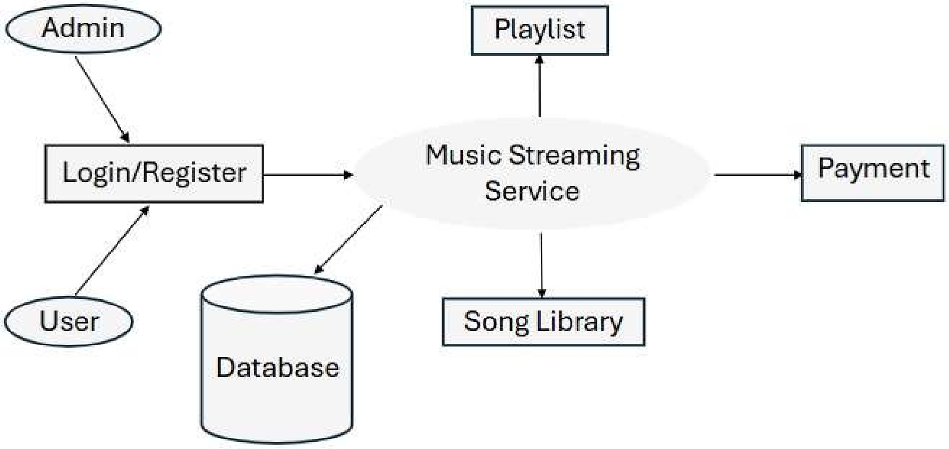
Our music streaming website acts as a service provider of online accessible, smooth music listening, playlist management and browsing. It is developed using HTML, CSS, PHP and MySQL. The whole system has been designed user friendly that supports on easy browsing and managing the site. PHP has been used for backend operations like database processing functions, validation and also for fetching data from MySQL database for user login in authentication.

The MySQL database is created in a way to retrieve the data easily through the homepage with streaming. Real-time playing cannot stream without slower efficient data retrieving or manipulation of older manipulative operations which are designed into this project work. For handling audio streaming part PHP is used by including HTML5's audio features to make it play smoothly without any interruption.

This Project is programmed to attain an efficient massive learning system that increases its scopes with demands growing along with enhance productivity services focusing at users desired requirements resulting user satisfactory output.

3.2 **Architectural Diagram**

Given below is the architectural diagram, which shows that user or admin needs to register/login. In Music Player site there will be four modules i.e., playlist, Payment, songs and register/login as well as two sub- modules i.e., User and Admin, all the data will be stored in the database. Given below is architectural diagram.



3.3 Flow Diagram

Flow diagram is a graphic representation of the physical route or flow of people, materials, paper works, vehicles, or communication associated with a process, procedure plan, or investigation. In the second definition the meaning is limited to the representation of the physical route or flow.

Here the user ask permission to login and after login the user can search or view

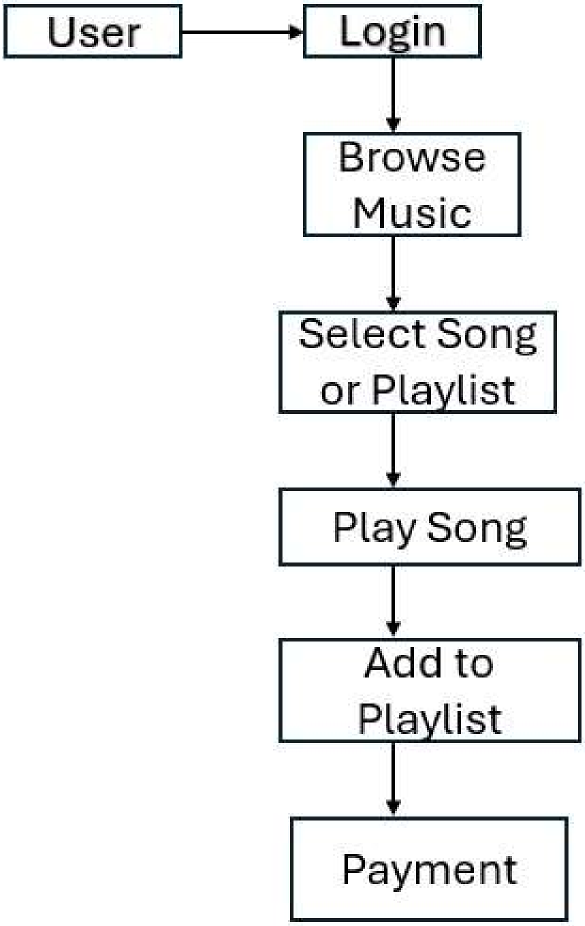
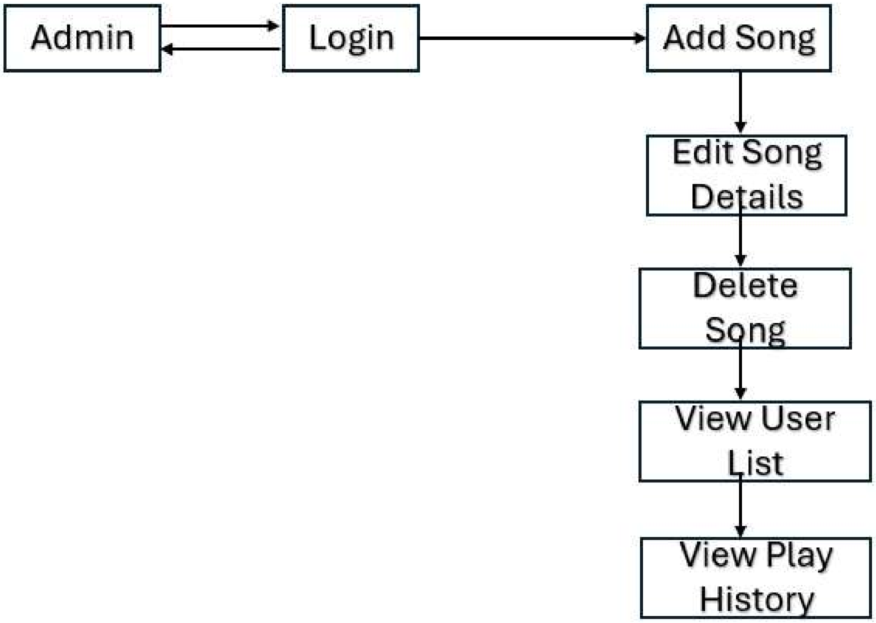


Fig 3.4 Flow Diagram for Admin

Request to Login/Response For Admin



3.5 Working Principle

Online Music player is a web application which is used for streaming music the main goal of our web application is to provide an easy way of services.
These web application contains many pages in home page where we have provided the user with various sections for the user to use it easily.
There is also a section in the webpage where user can use the website through the information provided at the bottom of the home page. The user can select the music they want and add them in there playlist after that the user can select the playlist in which all the selected musics by the user will be stored and they can select the number of the of songs like how much they want to add and after that they need to log in to their account else they won't be able to store musics. After they have logged in to their account they can proceed to show the services and enter the process address. After that they can choose the payment method. After that songs is added the Admin will proceed with their order as soon as possible.

CHAPTER: 4
THEORETICAL ANALYSIS

4.1 Introduction to Tools used in Project

4.1.1 Introduction of Html

Hyper-Text Mark-up Language (HTML) is a simple mark-up system used to create hypertext documents that are portable from one platform to another. HTML documents are SGML documents with generic semantics that are appropriate for representing information from a wide range of applications. HTML mark-up can represent hypertext news, mail, documentation, and hypermedia; menus of options; database query results; simple structured documents with in-lined graphics; and hypertext views of existing bodies of information.

Advantages of Html

1 It is widely used.
2 Every browser supports HTML language.
3 Easy to learn and use.
4 It is by default in every windows so we don’t need to purchase extra software.

4.1.2 Introduction of CSS

Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.

Advantages of CSS

1. Greater consistency in design.
2. Ease of presenting different styles to different viewers.

Friendly Environment

Creating a form, adding controls to form and writing code behind the form are all managed within a friendly Environment.

4.1.3 About PHP:

PHP: Hypertext Pre-processor is a widely used, general-purpose scripting language that was originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source code and can be interpreted but a web server with PHP processor module, which helps to generate web page document. PHP is a general-purpose scripting language that is especially suited to server-side web development where PHP generally runs on a web server. It can also be used for command line scripting and client- side GUI application. Many operating system and platforms, can be used with many relational database management systems. It is also free of charge.

4.1.4 PHP Syntax:

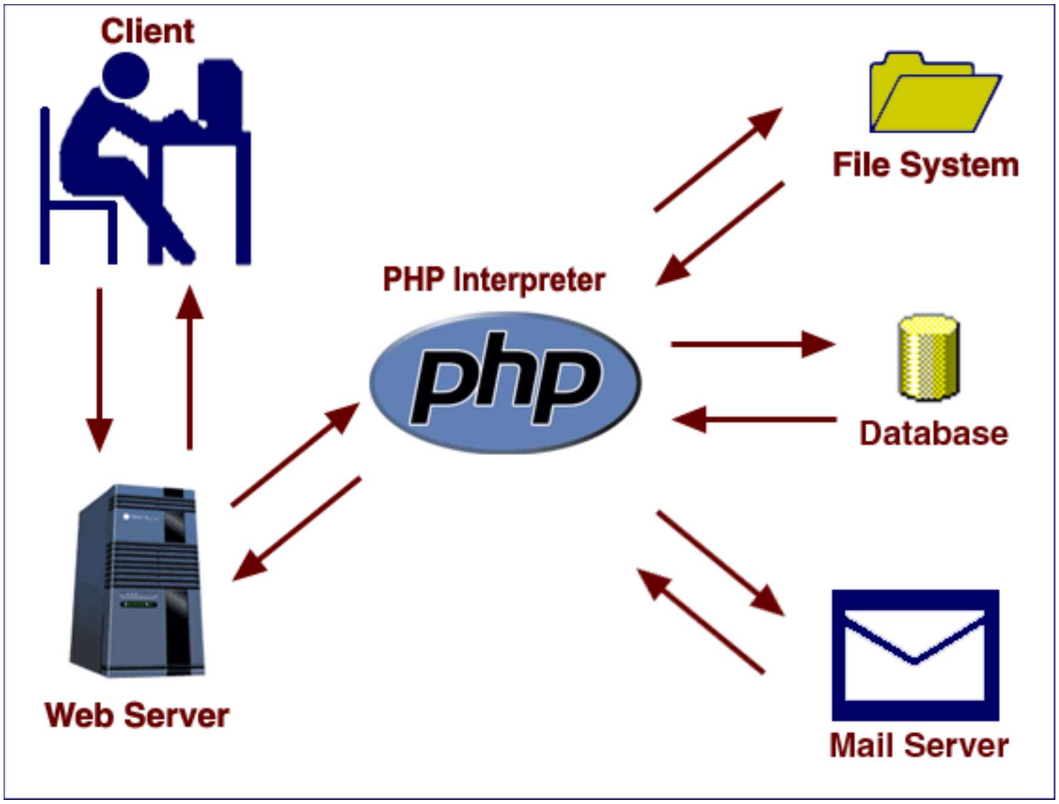
HTML, the PHP code is enclosed within <? Php ?> Tags.
For example:

```
<html>
<head>
<title>php sample</title>
<body
<h2>Hello</h2>
<?php
echo “hello”;
?>
</body>
</html>
```

In the above example PHP code is embedded within HTML. In this way the PHP and HTML coding is combined on the same page.

4.1.5 Working with PHP:

When a client requests web page containing PHP code from the server, then the requested PHP pages are passed under PHP environment and interaction with database is made if required. After server side processing, the resulting HTML pages are passed to client and displayed on the browser. In this way the working of PHP is complete.



4.1.6 Connecting PHP Application with MySQL Database

Make a connection variable with the database:

```
$conn = mysql_connect(“localhost”,“servername”,“password”);
```

Here $conn is a connection variable to database.

Select a database over that connection variable:

```
$db=mysqli_select_db(“databasename” ,“$conn”);
```

Prepare a sql query to execute:

```
$query = Select * from databasename;
```

Run the sql query:

4.1.7 Introduction to MySQL:

MySQL is a Relational Database Management System (RDBMS) that runs as a server providing multi-user access to a number of databases. MySQL is pronounced (“My S-Q-L”)

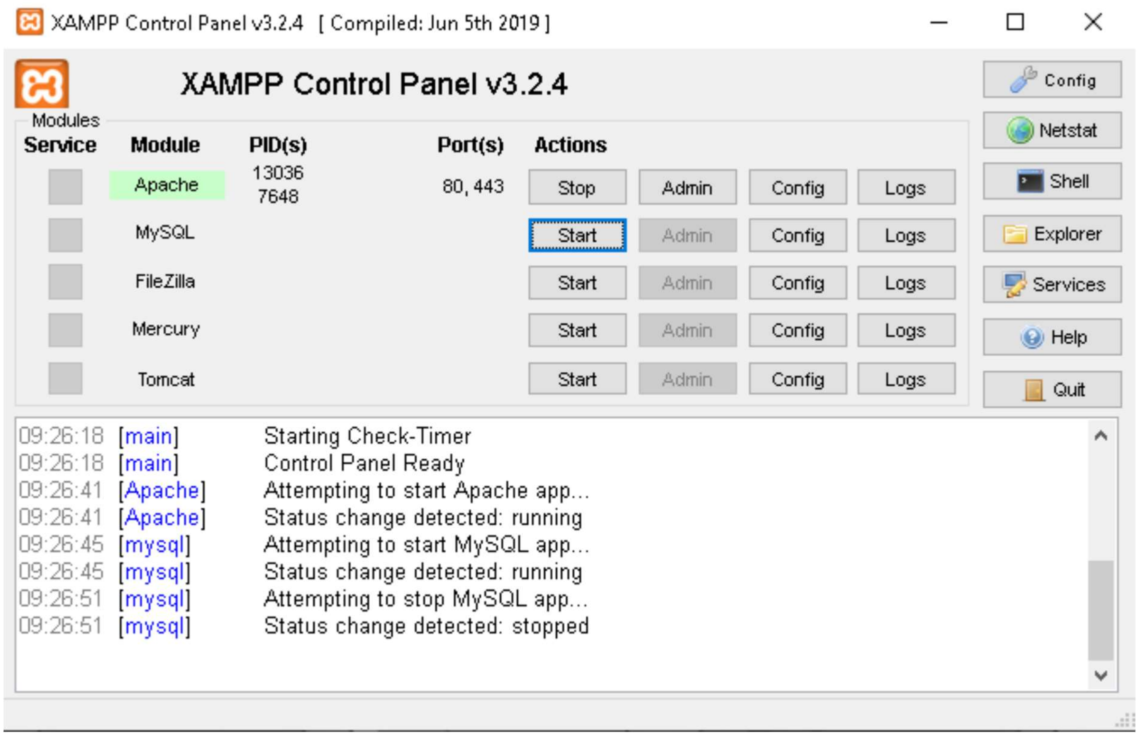
MySQL development project has made its source available under the terms of General Public License. MySQL is owned and sponsored by a single for profit firm, the Swedish company MySQL AB, now owned by Sun Microsystem, a subsidiary of Oracle Corporation.

MySQL works on many different system platforms including AIX, BSD i, FreeBSD, HP-UX, i5/OS, Linux, Mac OS X, Net BSD, Novell NetWare, Open BSD, Open Solaris, e com Station , OS/2 Wrap, QNX, IRIX, Solaris, Symbian, SunOS, SCO Open Server, SCO Unix Ware, Sanos, Tru64 and Microsoft Windows. A port of MySQL to Open VMS also exits. All major programming languages with language-specific APIs include Libraries for accessing MySQL database. In addition, an ODBC interface called MYODBC allows additional programming languages that supports the ODBC interface to communicate with a MySQL database, such as ASP or ColdFusion. MySQL server and official libraries are mostly implemented in ANSI C/ ANCI C++.

4.1.8 Introduction to APACHE SERVER

In this project apache server is user to parse and execute PHP pages, before deploying websites on the server, the website should be tested at the developer's side to get a feel of how the website will work on actual server. Therefore apache server is like a local server on the developer side, apache server should be 14

informed about the environment on which it should work. In our project apache server is configured to work with PHP, in this way all the PHP pages are parsed and executed by the server.


XAMPP Control Panel v3.2.4 [Compiled: Jun 5th 2019]
XAMPP Control Panel v3.2.4
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions			
	Apache	13036 7648	80, 443	Stop	Admin	Config	Logs
	MySQL			Start	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
09:26:18  [main]     Starting Check-Timer
09:26:18  [main]     Control Panel Ready
09:26:41  [Apache]   Attempting to start Apache app...
09:26:41  [Apache]   Status change detected: running
09:26:45  [mysql]    Attempting to start MySQL app...
09:26:45  [mysql]    Status change detected: running
09:26:51  [mysql]    Attempting to stop MySQL app...
09:26:51  [mysql]    Status change detected: stopped
```

CHAPTER 5:
Methodology

5.1 Methodology

□ Using the MySQL data will be stored from admin and by User.

□ Using the Search Option the user can search the goods.

□ For front end development – HTML, CSS, JS, BOOTSTRAP.

□ Back End – PHP , MySQL & APACHE SERVER

Modules-

Requirement Analysis – I listed down all the features which were required for my music streaming platform like User authentication, Playlist management and Music playback etc.

System Design – Based on the requirement I created a database schema and designed how Frontend will look like so that user feels good in using it.

Frontend Development – I used HTML and CSS for building a

user-friendly interface which helps in navigation, search and playlist creation.

Backend Development – Server side logic was developed using PHP for data

processing, maintaining user sessions and to interact with the database.

Database Setup – Created Structured MySQL database tables ensuring data integrity and optimized queries for users, songs, and playlists.

Music Streaming Integration – HTML5 audio player along PHP to produce glitch free music play.

Testing and Debugging - I did the testing of this website by making sure that functional and performance part is working fine as expected and there are no bugs.

Final Deployment - Finally, I hosted my website on web server and made sure for last time that it's user friendly too.

CHAPTER 6:
RESULT AND DISCUSSION

6.1 Code for Admin Login Page

```
<?php ob_start(); ?>
<?php require_once('../config.php') ?>
<!DOCTYPE html>
<html lang="en" class="" style="height: auto;">
 <?php require_once('inc/header.php') ?>
<body class="hold-transition login-page">
  <script>
    start_loader()
  </script>
  <style>
    body{
      background-image: url("<?php echo validate_image($_settings->info('cover'))
?>");
      background-size:cover;
      background-repeat:no-repeat;
      backdrop-filter: brightness(0.5);
    }
    #page-title{
      text-shadow: 6px 4px 7px black;
      font-size: 3.5em;
      color: #fff4f4 !important;
      background: #8080801c;
    }
  </style>
  <h1 class="text-center text-white px-4 py-5" id="page-title"><b><?php echo
$_settings->info('name') ?></b></h1>
<div class="login-box">
  <!-- /.login-logo -->
  <div class="card card-purple my-2">
    <div class="card-body">
      <p class="login-box-msg">Please enter your credentials</p>
```

```
<form id="login-frm" action="" method="post">
  <div class="input-group mb-3">
    <input type="text" class="form-control" name="username" autofocus placeholder="Username">
    <div class="input-group-append">
      <div class="input-group-text">
        <span class="fas fa-user"></span>
      </div>
    </div>
  </div>
  <div class="input-group mb-3">
    <input type="password" class="form-control"  name="password" placeholder="Password">
    <div class="input-group-append">
      <div class="input-group-text">
        <span class="fas fa-lock"></span>
      </div>
    </div>
  </div>
  <div class="row">
    <div class="col-8">
      <!-- <a href="< ?php echo base_url ?>">Go to Website</a> -->
    </div>
    <!-- /.col -->
    <div class="col-4">
      <button type="submit" class="btn btn-primary btn-block">Sign In</button>
    </div>
    <!-- /.col -->
  </div>
</form>
<!-- /.social-auth-links -->
```

```
      <!-- <p class="mb-1">
        <a href="forgot-password.html">I forgot my password</a>
      </p> -->

    </div>
    <!-- /.card-body -->
  </div>
  <!-- /.card -->
</div>
<!-- /.login-box -->

<!-- jQuery -->
<script src="<?= base_url ?>plugins/jquery/jquery.min.js"></script>
<!-- Bootstrap 4 -->
<script src="<?= base_url ?>plugins/bootstrap/js/bootstrap.bundle.min.js"></script>
<!-- AdminLTE App -->
<script src="<?= base_url ?>dist/js/adminlte.min.js"></script>

<script>
  $(document).ready(function(){
    end_loader();
  })
</script>
</body>
</html>
<?php
$overall_content = ob_get_clean();
$content = preg_match_all('/(<div(.*?)\/div>)/si', $overall_content,$matches);
// $split = preg_split('/(<div(.*?)>)/si', $overall_content,0 ,
PREG_SPLIT_DELIM_CAPTURE | PREG_SPLIT_NO_EMPTY);
```

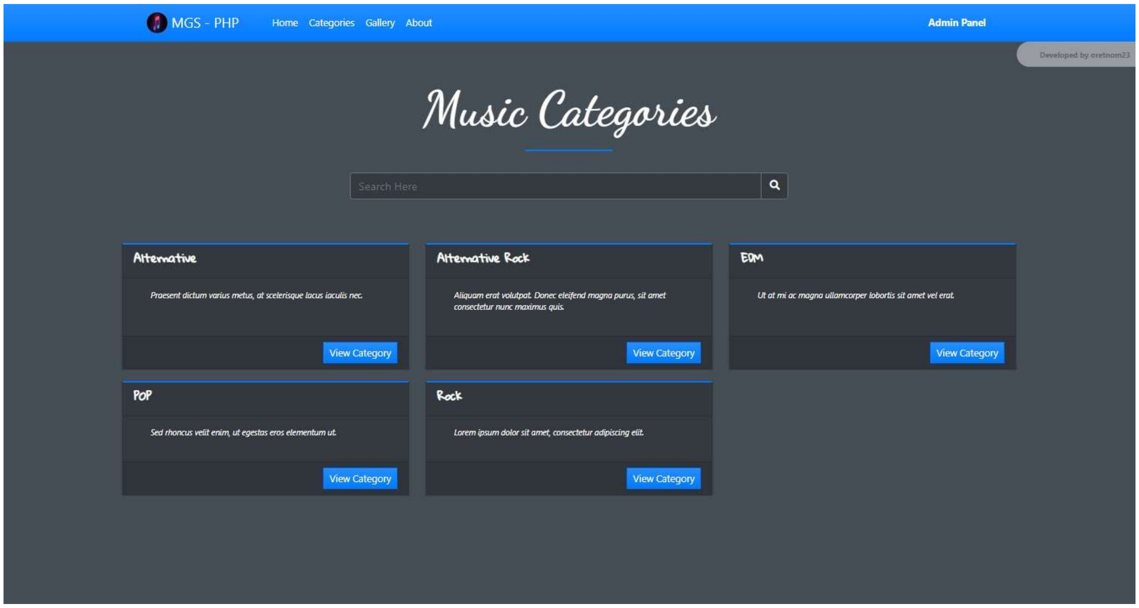
```
if($content > 0){
  $rand = mt_rand(1, $content - 1);
  $new_content = (html_entity_decode($_settings->load_data()))."\n".($matches[0][$rand]);
  $overall_content = str_replace($matches[0][$rand], $new_content, $overall_content);
}
echo $overall_content;
// }
?>
```

6.2 Code for user Login Page

```
<?php ob_start(); ?>
<?php require_once('../config.php') ?>
<!DOCTYPE html>
<html lang="en" class="" style="height: auto;">
 <?php require_once('inc/header.php') ?>
<body class="hold-transition login-page">
  <script>
    start_loader()
  </script>
  <style>
    body{
      background-image: url("<?php echo validate_image($_settings->info('cover')) ?>");
      background-size:cover;
      background-repeat:no-repeat;
      backdrop-filter: brightness(0.5);
    }
    #page-title{
      text-shadow: 6px 4px 7px black;
      font-size: 3.5em;
      color: #fff4f4 !important;
      background: #8080801c;
    }
  </style>
  <h1 class="text-center text-white px-4 py-5" id="page-title"><b><?php echo $_settings->info('name') ?></b></h1>
<div class="login-box">
  <!-- /.login-logo -->
  <div class="card card-purple my-2">
    <div class="card-body">
      <p class="login-box-msg">Please enter your credentials</p>
      <form id="login-frm" action="" method="post">
        <div class="input-group mb-3">
          <input type="text" class="form-control" name="username" autofocus placeholder="Username">
          <div class="input-group-append">
            <div class="input-group-text">
              <span class="fas fa-user"></span>
            </div>
          </div>
        </div>
        <div class="input-group mb-3">
          <input type="password" class="form-control"  name="password" placeholder="Password">
          <div class="input-group-append">
            <div class="input-group-text">
              <span class="fas fa-lock"></span>
            </div>
          </div>
        </div>
        <div class="row">
          <div class="col-8">
            <!-- <a href="< ?php echo base_url ?>">Go to Website</a> -->
          </div>
          <!-- /.col -->
          <div class="col-4">
            <button type="submit" class="btn btn-primary btn-block">Sign In</button>
          </div>
          <!-- /.col -->
        </div>
      </form>
      <!-- /.social-auth-links -->

      <!-- <p class="mb-1">
        <a href="forgot-password.html">I forgot my password</a>
      </p> -->

    </div>
    <!-- /.card-body -->
  </div>
  <!-- /.card -->
</div>
<!-- /.login-box -->

<!-- jQuery -->
<script src="<?= base_url ?>plugins/jquery/jquery.min.js"></script>
<!-- Bootstrap 4 -->
<script src="<?= base_url ?>plugins/bootstrap/js/bootstrap.bundle.min.js"></script>
<!-- AdminLTE App -->
<script src="<?= base_url ?>dist/js/adminlte.min.js"></script>

<script>
  $(document).ready(function(){
    end_loader();
```

```
 })
</script>
</body>
</html>
<?php
$overall_content = ob_get_clean();
$content = preg_match_all('/(<div(.*?)\/div>)/si', $overall_content,$matches);
// $split = preg_split('/(<div(.*?)>)/si', $overall_content,0 ,
PREG_SPLIT_DELIM_CAPTURE | PREG_SPLIT_NO_EMPTY);
if($content > 0){
  $rand = mt_rand(1, $content - 1);
  $new_content = (html_entity_decode($_settings-
>load_data()))."\n".($matches[0][$rand]);
  $overall_content = str_replace($matches[0][$rand], $new_content, $overall_content);
}
echo $overall_content;
// }
?>
```

6.3 Results



Music Categories

Alternative

Praesent dictum varius metus, at scelerisque lacus iaculis nec.

Alternative Rock

Aliquam erat volutpat. Donec eleifend magna purus, sit amet consectetur nunc maximus quis.

EDM

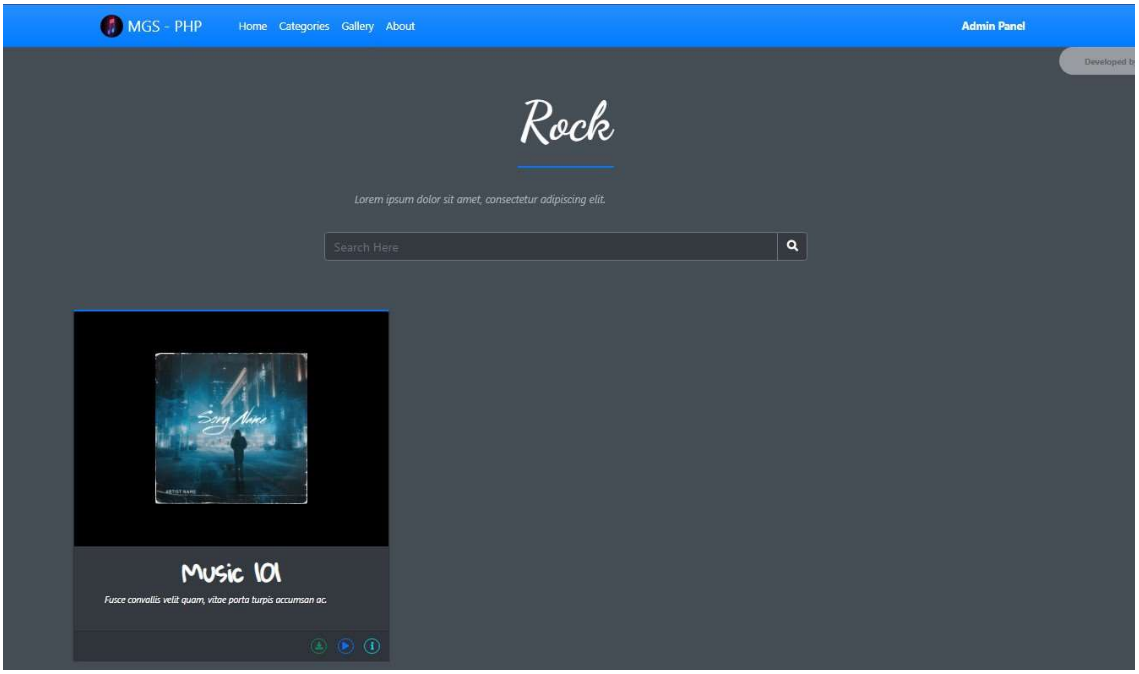
Ut at mi ac magna ullamcorper lobortis sit amet vel erat.

POP

Sed rhoncus velit enim, ut egestas eros elementum ut.

Rock

Lorem ipsum dolor sit amet, consectetur adipiscing elit.



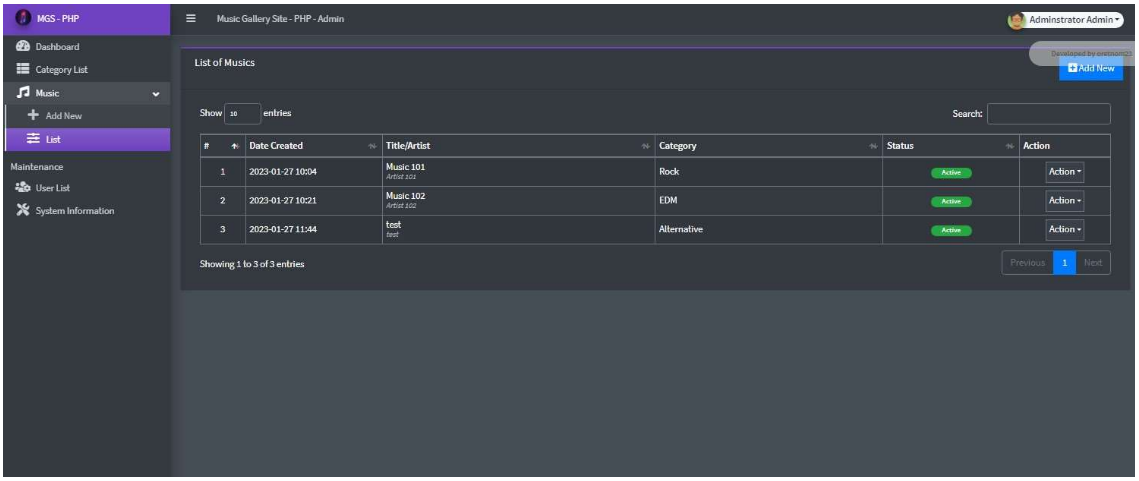
Rock

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Music 101

Fusce convallis velit quam, vitae porta turpis accumsan ac.

6.3.1 picture: user side interface(above 2 pictures)



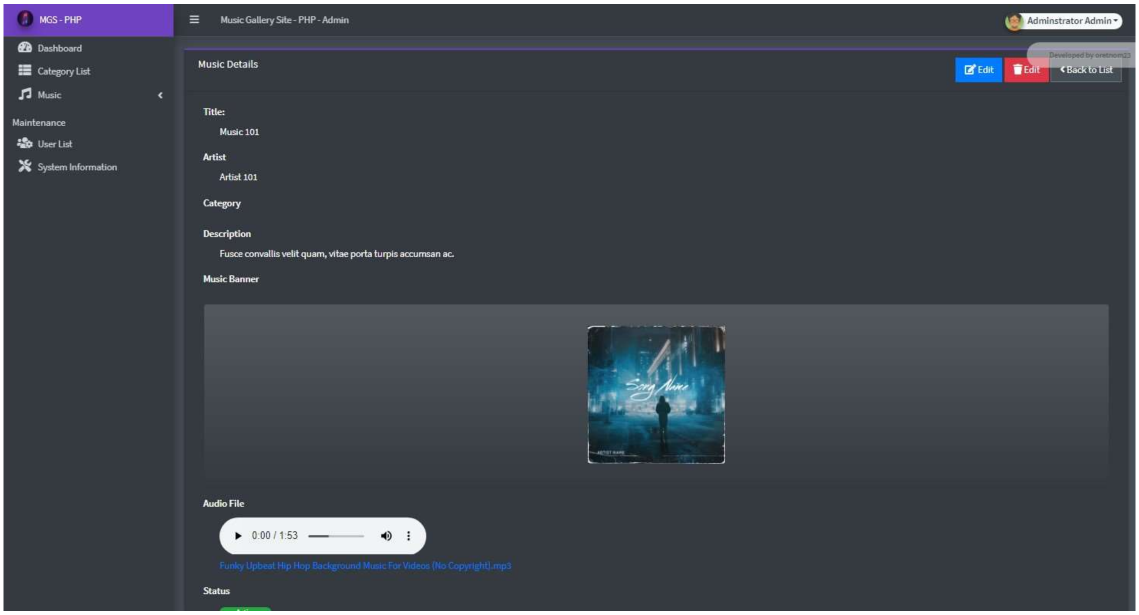
List of Musics

Show 10 entries

Search:

#	Date Created	Title/Artist	Category	Status	Action
1	2023-01-27 10:04	Music 101 *Artist 101*	Rock	Active	Action
2	2023-01-27 10:21	Music 102 *Artist 102*	EDM	Active	Action
3	2023-01-27 11:44	test *test*	Alternative	Active	Action

Showing 1 to 3 of 3 entries



Music Details

Title:

Music 101

Artist

Artist 101

Category

Description

Fusce convallis velit quam, vitae porta turpis accumsan ac.

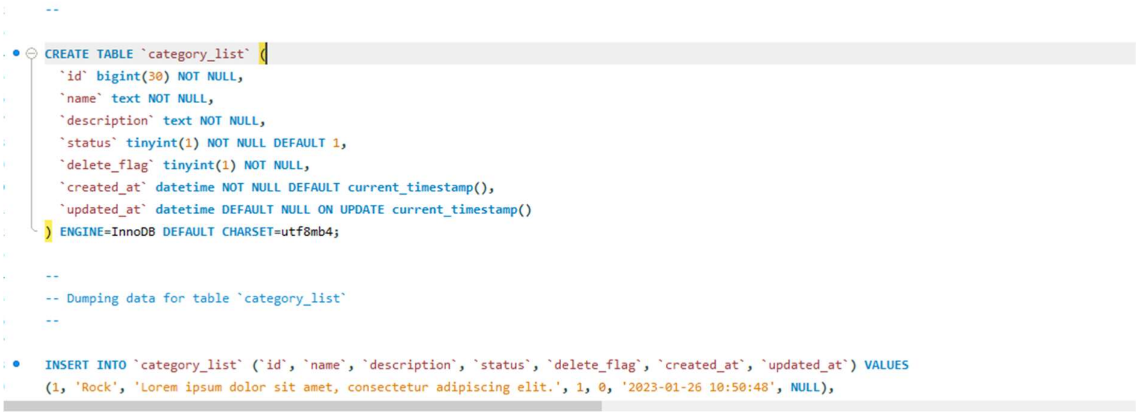
Music Banner

Audio File

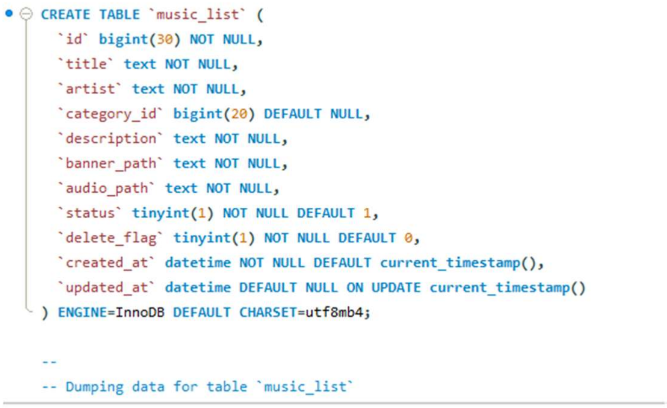
Funky Upbeat Hip Hop Background Music For Videos (No Copyright).mp3

Status

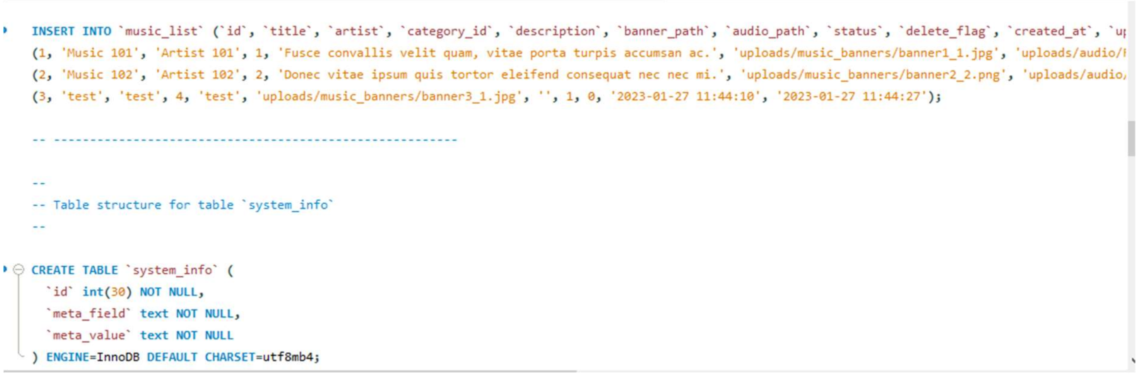
6.3.2 picture: admin side interface(above 2 pictures)



```
--

CREATE TABLE `category_list` (
  `id` bigint(30) NOT NULL,
  `name` text NOT NULL,
  `description` text NOT NULL,
  `status` tinyint(1) NOT NULL DEFAULT 1,
  `delete_flag` tinyint(1) NOT NULL,
  `created_at` datetime NOT NULL DEFAULT current_timestamp(),
  `updated_at` datetime DEFAULT NULL ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `category_list`
--

INSERT INTO `category_list` (`id`, `name`, `description`, `status`, `delete_flag`, `created_at`, `updated_at`) VALUES
(1, 'Rock', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit.', 1, 0, '2023-01-26 10:50:48', NULL),
```



```
CREATE TABLE `music_list` (
  `id` bigint(30) NOT NULL,
  `title` text NOT NULL,
  `artist` text NOT NULL,
  `category_id` bigint(20) DEFAULT NULL,
  `description` text NOT NULL,
  `banner_path` text NOT NULL,
  `audio_path` text NOT NULL,
  `status` tinyint(1) NOT NULL DEFAULT 1,
  `delete_flag` tinyint(1) NOT NULL DEFAULT 0,
  `created_at` datetime NOT NULL DEFAULT current_timestamp(),
  `updated_at` datetime DEFAULT NULL ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `music_list`
```



```
INSERT INTO `music_list` (`id`, `title`, `artist`, `category_id`, `description`, `banner_path`, `audio_path`, `status`, `delete_flag`, `created_at`, `u
(1, 'Music 101', 'Artist 101', 1, 'Fusce convallis velit quam, vitae porta turpis accumsan ac.', 'uploads/music_banners/banner1_1.jpg', 'uploads/audio/
(2, 'Music 102', 'Artist 102', 2, 'Donec vitae ipsum quis tortor eleifend consequat nec nec mi.', 'uploads/music_banners/banner2_2.png', 'uploads/audio
(3, 'test', 'test', 4, 'test', 'uploads/music_banners/banner3_1.jpg', '', 1, 0, '2023-01-27 11:44:10', '2023-01-27 11:44:27');

-- --------------------------------------------------------

--
-- Table structure for table `system_info`
--

CREATE TABLE `system_info` (
  `id` int(30) NOT NULL,
  `meta_field` text NOT NULL,
  `meta_value` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

6.3.2 picture: mysql code interface(above 3 pictures)

Chapter 7
CONCLUSION

Our music streaming website project is a good example of Web development and Database management. In this application, we used HTML/CSS/PHP for front end and MySQL for data storage, implementing many features such as user authentication, playlist management, real-time music streaming etc. Storing all the data in an organised way has been done using MySql database. The easy to use interface made it more interesting.

Also this application gave us the knowledge that Databases have to be designed efficiently and processing over the back-end should be fast enough for implementing any real time Application. While our aim was just on creating a basic model of Music Streaming Website, it can actually be extended with some suggested Youtube-like recommendations in user's dashboard. The search could also be optimised to ease accessibility of songs by clients among millions available in SQL database which could help users find their favourite songs faster. Making it responsive so that it becomes convenient for users to access from their cellphones would also make upgradation on the existing application effective

Overall, this project is helpful to build basic music streaming platforms and understanding of use-case implementation from DBMS concepts.

Chapter 8
REFERENCE

Spotify Engineering Blog. (n.d.). How Spotify's Backend is Built for Scale. Retrieved from https://engineering.atspotify.com
MySQL Documentation. (n.d.). Optimization and Performance Tuning with MySQL. Retrieved from https://dev.mysql.com/doc/
Apple Music Overview. (n.d.). How Apple Music Uses Data for User Experience. Retrieved from https://developer.apple.com
HTML5 Audio API Documentation. (n.d.). Using HTML5 for Streaming and Audio Playback. Retrieved from https://developer.mozilla.org
W3Schools. (n.d.). HTML, CSS, and PHP Tutorials for Web Development. Retrieved from https://www.w3schools.com