VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI-590018

A DBMS Mini-Project
On

"MUSIC MANAGEMENT SYSTEM"

A Mini project report submitted in partial fulfillment of the requirements for the 5th semester of ***Bachelor of Engineering in Computer Science Engineering*** *of Visvesvaraya Technological University, Belagavi*

Submitted by:
Darshan R.
1ST21CS057

Under the Guidance of:
Prof. Yashaswi R.
Assistant Professor
Department of CSE
SaIT,Bengaluru

SAMBHRAM
INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SAMBHRAM INSTITUTE OF TECHNOLOGY

M.S. PALYA, JALAHALLI, BENGALURU – 5560097

2023-2024

SAMBHRAM
INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the mini project work entitled **"Music Management System"** has been successfully carried out by Darshan R. bearing USN 1ST21CS057, bonafide student of **Sambhram Institute of Technology** in partial fulfillment of the requirements for the **5th semester** of **Bachelor of Engineering** in **Computer Science and Engineering** of **Visvesvaraya Technological University**, Belagavi, during academic year 2023-2024. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the laboratory requirements of 5th semester BE, CSE.

Prof. Yashaswi R.
Assistant Professor
Dept. of CSE

Dr. T John Peter
HOD
Dept. of CSE

External Viva:

Name of the Examiners

1.

2. **Signature with Date**

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped me in carrying out this project work. We would like to take this opportunity to thank them all.

I would like to thank **Dr. H.G. Chandrakanth**, Principal, SaIT, Bangalore, for his moral support towards completing my project. I would like to thank **Dr.T. John Peter**, Prof & Head, Department of Computer Science & Engineering, SaIT, Bangalore, for his valuable suggestions and expert advice. I deeply express my sincere gratitude to my guide to **Prof. Yashaswi R** Assistant Professor, SaIT, Bangalore, for their able guidance, regular source of encouragement and assistance throughout this project.

I would like to thank all the teaching and non-teaching staff of Department of Computer Science & Engineering, SaIT, Bengaluru for their constant support and encouragement.

Date: **Darshan R.**

Place: Bengaluru

ABSTRACT

Melomaniczz is an innovative music player application developed as a mini-project, integrating principles of Database Management System to offer enhanced music organization capabilities.

The application allows users to play their music files seamlessly. Melomaniczz employs a well-structured database backend to store music metadata such as artist names, album titles, and release years. This enables users to easily search and sort music from their music library, enhancing the overall music listening experience.With a user-friendly interface, Melomaniczz offers better navigation and playback controls, making it easy for users to access their favourite songs and albums.

Additionally, the application incorporates feature like playback speed setting.The application also prioritizes security measures to safeguard user information and music files where use create their own password to access his application page.

Melomaniczz represents a significant advancement in music player technology by integrating DBMS principles to provide users with a seamless and personalized music listening experience. Its intuitive interface, robust features, and efficient database management make it a standout choice for music enthusiasts looking to enjoy their music collections effortlessly.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Overview

The Music management system is a computer-based application that provide many numbers of songs and is categorized based on the artists, albums so that the user can search the songs based on their necessity. Melomaniczz represents a significant advancement in music player technology by integrating DBMS principles to provide users with a seamless and personalized music listening experience. Its intuitive interface, robust features, and efficient database management make it a standout choice for music enthusiasts looking to enjoy their music collections effortlessly. Managing and enjoying vast music collections efficiently is a common challenge. Melomaniczz aims to address this challenge by providing users with a feature-rich music player that leverages the power of database technology.

1.2 Problem statement

Design and implement the effective music system where we search, listen and download the music of their choice by categorizing based on song title, artists and albums. The abundance of music files available across various platforms, users encounter difficulties in organizing, accessing, and playing their music seamlessly. This problem statement outlines the specific issues to be addressed. The Music Management System addresses the challenges faced by users in managing and enjoying their music collections effectively.

The primary problem addressed by users is that they have music files scattered across different devices, directories, and formats, leading to a lack of organization and difficulty in finding specific songs or albums. This could be one of the problems that this project may reduce.

Traditional music players offer limited search capabilities, making it challenging for users to locate desired songs or artists quickly. But in our project, we made an efficient searching facility so that the user searches his song of interest based on the artist or the album or song name whenever they access this application.

1.3 Objectives

- The primary object of this project is to play audio files**.** Users can listen to their favorite songs, albums, or playlists.
- To develop a system that allows users to search and manage their music files effectively.
- For storing metadata such as song titles, artist names, album titles, and release years in a structured database.
- To implement robust search and filtering functionalities to enable users to find specific songs, albums, or artists quickly.
- Allows users to search by various criteria, including song title, artist name, album title.
- To implement mechanisms to ensure data integrity and security, including user authentication, access control, and encryption of sensitive information.
- To allow users to adjust playback speed to optimize their listening experience.

2.Literature survey

2.1 Tools and Technologies

2.1.1 XAMPP

Xampp stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P), and Perl (P). It is a free and open-source software that allows developers to create a local web server on their computers. This is useful for testing and developing web-based applications, such as your inventory management system, without the need to access a remote web server.

XAMPP consists of a package of software that includes Apache, a web server; MariaDB (or MySQL), a relational database management system; PHP, a programming language used for server-side scripting; and Perl, a programming language used for web development. All of these components work together to allow you to run a web server on your own computer, which is useful for developing, testing and troubleshoot web-based applications.

XAMPP provides a simple, easy-to-use interface that makes it easy for developers to set up a local web server on their computers, and it is available for Windows, macOS, and Linux. This allows developers to test their web applications on a local server, which can save a lot of time, and effort, since it eliminates the need to upload files to a remote server every time a change is made. Additionally, XAMPP comes with a control panel that makes it easy to start and stop the various components of the software, and to configure settings such as PHP, MySQL, and Apache.

2.1.2 HTML

HTML (Hypertext Mark-up Language), is the standard mark-up language used to create and design web pages. It provides the structure and content of a webpage by using a set of elements or tags, which define different parts of the page such as headings, paragraphs, images, links, forms, and more. HTML documents are made up of a series of elements, which are represented by tags, such as <p> for paragraphs, <img> for images, <a> for anchor tag and many more. These tags are used to define the structure and content of a web page, and the web browser uses them to display the page in a meaningful way.

2.1.3 CSS

CSS (Cascading Style Sheets), is a stylesheet language that is used to describe the presentation of a document written in HTML. It allows developers to separate the presentation of a web page from its content and structure, which makes it easier to maintain and update the appearance of a website. Using CSS, developers can define styles for specific elements on a web page, such as the colour of text or the

background colour of a button. These styles can then be applied to multiple elements throughout the website, which saves a lot of time and effort compared to styling each element individually.

2.1.4 JavaScript

JavaScript is a programming language that is primarily used to create interactive front-end web applications. It allows developers to add dynamic behaviour to web pages, such as changing content, responding to user input, and validating forms. JavaScript code can be run directly within web browsers, allowing developers to create interactive, responsive web pages that can update in Realtime without the need for a page refresh. JavaScript can also be used to manipulate the Document Object Model (DOM), which represents the structure of a web page, allowing developers to dynamically change the layout, content, and appearance of a web page.

3.Requirement Specification

3.1 Functional Requirements

3.1.1 Admin-level functional requirements

The system should allow the admin to access the user data from the user sign-up module whenever the user sign-up. It also allows the admin to add, delete or edit the songs to the database. It allows to add artist, albums that are not present in the database. Admin has the access to remove the user from the database.

3.1.2 User-level functional requirements

The system should allow the user to create their own sign-in id through the sign-up page by providing the username, email and password for the first time and later can sign -in through the username and their password through Sign-in page.

3.2 Non-Functional Requirements

3.2.1 Software requirements

- Operating system: The system is compatible with Windows, macOS and Linux operating systems.
- Web server: The system is able to run on Apache web server, which is included in the XAMPP package
- Database: The system uses a relational database management system, such as MySQL for storing and retrieving data.
- Programming languages: The system should be developed using PHP as the primary programming language.
- User interface: The system has a user-friendly and intuitive interface, making it easy for users to search and listen to their Choice of music.
- Performance: The system is able to handle a large amount of data and be able to generate the searched output.
- Security: The system has appropriate security features to protect sensitive user data like password to ensure the confidentiality.

3.2.2 Hardware requirements

- Processor: The system should require a minimum of 2GHz processor with at least 4GB of RAM to ensure smooth performance.
- Storage: The system should require a minimum of 30GB of storage space for the system files, software installation, and data storage.
- Network: The system should have a stable internet connection, either wired or wireless, for remote access and data transfer.
- Display: The system should be compatible with a monitor resolution of at least 1024x768.
- Peripherals: The system should be compatible with standard input devices such as keyboard, mouse, and scanner, if necessary.

3.3 Database Requirements

- **User database:** This stores the information for user sign-in like user id, email and user password where user id is the primary key attribute. Here many users manage the same music system hence there is N:1 cardinality ratio between the user and music.
- **Admin database:** This stores the sign-in information of the admin that is username and their password by ensuring secure login. The username will be the primary key and there is N:1 cardinality ratio where many admin can manage, add, delete and edit the different attribute of the music system.
- **Artist database:** The attribute like Artist name, no_of_artist songs, artist cover is all the part of it. The artist name being the primary key, it has 1: N cardinality ratio where one artist can write many songs. The no_of_artist songs contains the count of songs that has the particular artist name.
- **Album database:** This has the attributes like Album name that is primary key, album Release year, no_of_songs that has the count of songs that come under this album and Album cover that has the album poster. The cardinality ratio is 1: N where one album has many songs.
- **Songs database:** It has the attributes like title , artist name, album name and the audio where title is the primary key, artist name and album_name is referenced by the artist and album data table respectively and audio has the music file that is playable and downloadable. It has N:1 cardinality ratio where many songs belong to a album and any songs are sung by a artist.

4.System Design

4.1 ER Diagram

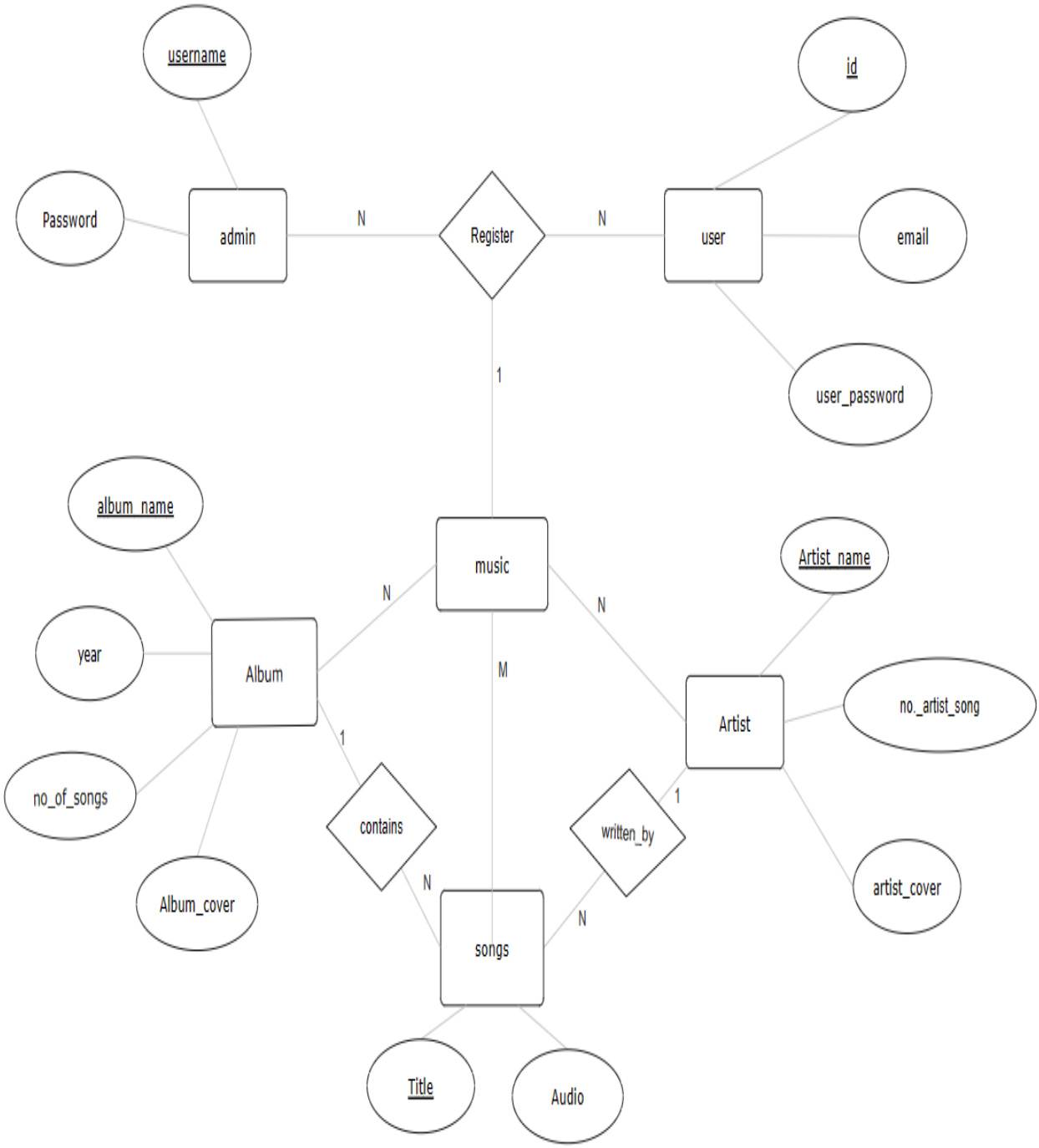

Figure 4.1 ER diagram for music management system

4.2 Relation Schema

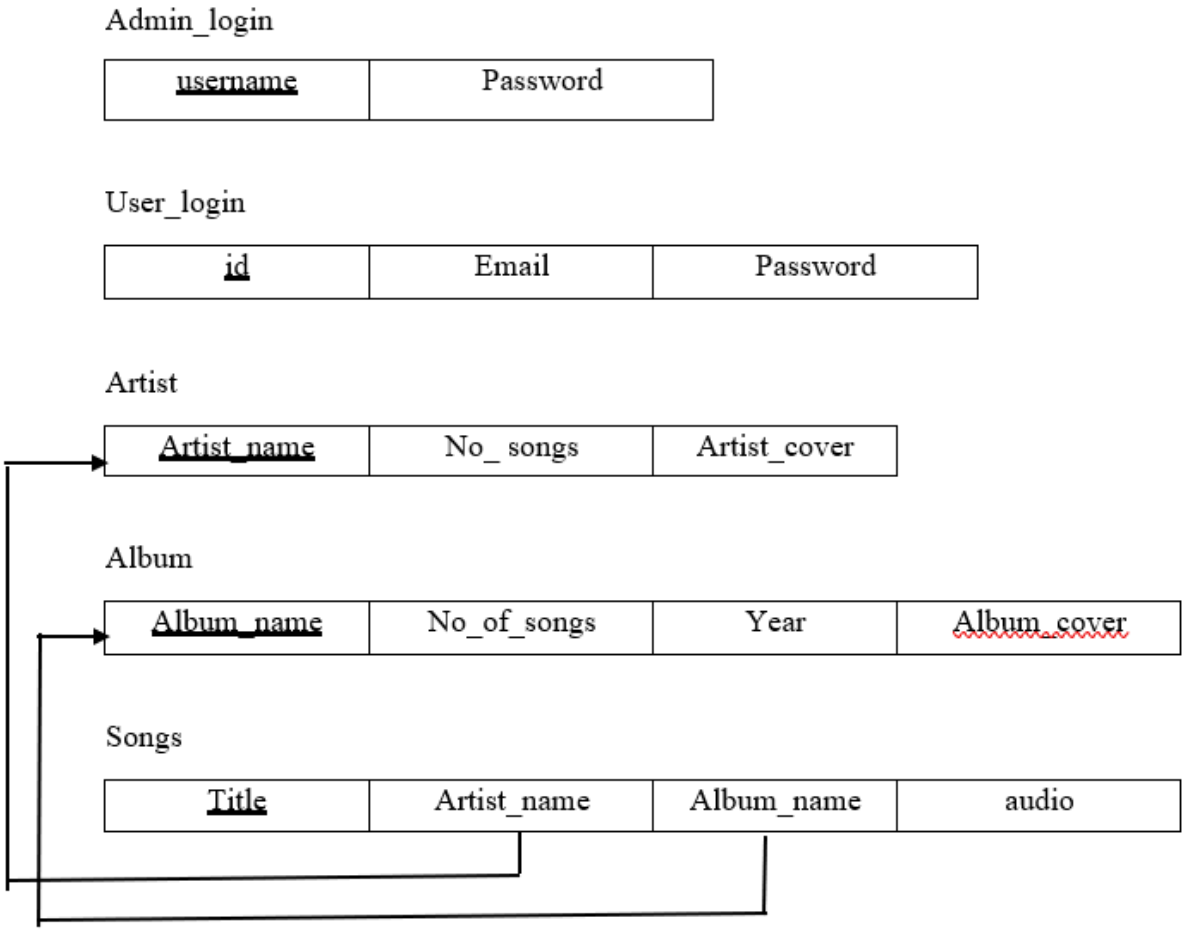

Figure 4.2 Relation Schema of music management system

5.Implementation

5.1 Frontend

Index

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Welcome Page</title>
<style>
body {
margin: 0;
padding: 0;
height: 100vh;
display: flex;
justify-content: center;
align-items: center;
background: linear-gradient(135deg, #667eea, #764ba2);
}
.container {
text-align: center;
background: rgba(255, 255, 255, 0.8);
padding: 20px;
border-radius: 10px;
box-shadow: 0 10px 20px rgba(0, 0, 0, 0.1);
}button {
padding: 50px 50px;
margin: 10px;
background: #667eea;
color: white;
border: none;
border-radius: 5px;
cursor: pointer;
transition: background 0.3s;
}button:hover {
background: #764ba2;
}
</style>
</head>
<body>
<div class="container">
<h1>Register</h1>
```

```
<button onclick="window.location.href = 'user_signin.php'">User</button>
<button onclick="window.location.href = 'admin_signin.php'">Admin</button>
</div>
</body>
</html>
```

User-signin

```
<?php
require 'connection.php';
if ($_SERVER["REQUEST_METHOD"]=="POST") {
$id=$_POST["user"];
$pass=$_POST["pass"];
$sql="SELECT * FROM user_login WHERE id = '$id';";
$result=mysqli_query($conn,$sql);
$num=mysqli_num_rows($result);
if ($num==1) {
$row=mysqli_fetch_assoc($result);
 if ($pass==$row['pass']) {
session_start();
$_SESSION['loggedin']=true;
$_SESSION['uid']=$id;
header("location: home1.php");
}
else{
$passerror="Password incorrect";
}
}
else{
$iderr="Error no such id exists, Create one first";
}
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Login Page</title>
<style>
  body {
 margin: 0;
 padding: 0;
 height: 100vh;
 display: flex;
```

```
justify-content: center;
align-items: center;
background: linear-gradient(135deg, #667eea, #764ba2);
overflow: hidden;
}
.container {
position: relative;
width: 300px;
height: 350px;
background: rgba(255, 255, 255, 0.8);
border-radius: 10px;
box-shadow: 0 10px 20px rgba(0, 0, 0, 0.1);
overflow: hidden;
text-align: center;
animation: animateContainer 5s ease-in-out infinite alternate;
}
@keyframes animateContainer {
0% {
transform: scale(1);
}
100% {
transform: scale(1.05);
}
}
form {
padding: 20px;
}
input[type="text"], input[type="password"] {
 width: 100%;
 padding: 10px;
 margin-bottom: 20px;
 border: 1px solid #ccc;
 border-radius: 5px;
 box-sizing: border-box;
 }
 button {
 width: 100%;
 padding: 10px;
 background: #667eea;
 color: white;
 border: none;
 border-radius: 5px;
 cursor: pointer;
 transition: background 0.3s;
```

```
  }

 button:hover {
 background: #764ba2;
 }
</style>
</head>
<body>
<div class="container">
<h1>Sign-in User</h1>
<form action="user_signin.php" method="post">
<input type="text" name='user' placeholder="Username" required>
<?php
if (isset($iderr)) {
 ?>
 <p class="error"><?php echo $iderr ?></p>
 <?php
 }
 ?>
 <input type="password" name='pass' placeholder="Password" required>
 <?php
  if (isset($passerror)) {
  ?>
  <p class="error"><?php echo $passerror ?></p>
  <?php
  }
  ?>
 <button type="submit">Login</button>
 <a href="./sign-up.php">Sign-up</a>
 </form>
 <a href="./index.php">back</a>
 </div>
 <?php
 if(isset($_GET['uploaded']) && $_GET['uploaded']){
 ?>
 <script>
 alert(" Signed Up sucessfully");
 </script>
 <?php
 }
 ?>
</body>
</html>
```

Admin Signin

```
<?php
require 'connection.php';
if ($_SERVER["REQUEST_METHOD"]=="POST") {
$username=$_POST["username"];
$password=$_POST["password"];
$sql="SELECT * FROM admin_login WHERE username = '$username';";
$result=mysqli_query($conn,$sql);
$num=mysqli_num_rows($result);
if ($num==1) {
$row=mysqli_fetch_assoc($result);
if ($password==$row['password']) {
session_start();
$_SESSION['loggedin']=true;
$_SESSION['uid']=$username;
header("location: ./dashboard.php");
}
else{
$passworderror="Password incorrect";
}
}
else{
$usernameerr="Error no such id exists, Create one first";
}
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Login Page</title>
<style>
  body {
  margin: 0;
  padding: 0;
  height: 100vh;
  display: flex;
  justify-content: center;
  align-items: center;
  background: linear-gradient(135deg, #667eea, #764ba2);
  overflow: hidden;
  }
```

```
.container {
position: relative;
width: 300px;
height: 350px;
background: rgba(255, 255, 255, 0.8);
border-radius: 10px;
box-shadow: 0 10px 20px rgba(0, 0, 0, 0.1);
overflow: hidden;
text-align: center;
animation: animateContainer 5s ease-in-out infinite alternate;
}
@keyframes animateContainer {
0% {
transform: scale(1);
}
100% {
transform: scale(1.05);
}
}
form {
padding: 20px;
}
input[type="text"], input[type="password"] {
width: 100%;
padding: 10px;
margin-bottom: 20px;
border: 1px solid #ccc;
border-radius: 5px;
box-sizing: border-box;
}
button {
width: 100%;
padding: 10px;
background: #667eea;
color: white;
border: none;
border-radius: 5px;
cursor: pointer;
transition: background 0.3s;
}
button:hover {
background: #764ba2;
}
</style>
```

```
</head>
<body>
<div class="container">
<h1>Sign in</h1>
<form action="admin_signin.php" method="post">
<input type="text" name='username' placeholder="Username" required>
<?php
 if (isset($usernameerr)) {
 ?>
<p class="error"><?php echo $usernameerr ?></p>
<?php
}
?>
<input type="password" name='password' placeholder="Password" required>
<?php
if (isset($passworderror)) {
?>
<p class="error"><?php echo $passworderror ?></p>
<?php
 }
?>
 <button type="submit">Login</button>
 </form>
 <a href="./index.php">back</a>
 </div>
 </body>
 </html
```

Homepage

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Welcome to Music Database</title>
<style>
body {
font-family: Arial, sans-serif;
margin: 0;
padding: 0;
background-repeat: no-repeat;
background-size: cover;
```

```
background-image: url(https://www.atlantaonthecheap.com/wp-content/uploads/2023/07/music-concept-Depositphotos_10701850_S-e1688665472811.jpg);
text-align: center;
 }
 header {
background-color: transparent;
color: white;
padding: 10px 0;
}
 h1 {
margin: 20px 0;
font-size: 4em;
font-family: 'Times New Roman', Times, serif;
 }
 .container {
max-width: 800px;
margin: 0 auto;
padding: 20px;
}
h4 {
font-family: Arial, Helvetica, sans-serif;
font-size: 2em;
line-height: 1.5;
}
 h6 {
font-family: Arial, Helvetica, sans-serif;
font-size: 1.5em;
line-height: 1.5;
 }
 a {
color: #333;
text-decoration: none;
padding: 5px 10px;
background-color: #f0f0f0;
border-radius: 5px;
 }
 a:hover {
background-color: #ccc;
}
</style>
</head>
<body>
<header>
```

```
<h1>Welcome to the Melomaniczz</h1>
</header>
<div class="container">
<h4>Melomaniczz contains a collection of music albums, artists, and songs.</h4>
 <h6>Explore to find your favorite music!</h6>
 <a href="./artists.php">View Artists</a>
 <a href="./album.php">View Albums</a>
 <a href="./song.php">View Songs</a>
  </div>
 <form method="GET">
 <div class="search">
 <span class="search-icon material-symbols-outlined">search</span>
 <input class="search-input" type="search" name="search" placeholder="search">
 <input type="submit" value="Search">
 </div>
 </form>
 <?php
 // Connect to your database
 $servername = "localhost:3306";
 $username = "root";
 $password = "";
 $dbname = "music";
 $conn = new mysqli($servername, $username, $password, $dbname);
 // Check connection
 if ($conn->connect_error) {
 die("Connection failed: " . $conn->connect_error);
 }
 $searchTerm = "";
 if(isset($_GET['search'])) {
 $searchTerm = $_GET['search'];
 // Check if search term is not empty
 if (!empty($searchTerm)) {
  $sql = "SELECT * FROM songs WHERE audio LIKE '%$searchTerm%'";
  $result = $conn->query($sql);
  if ($result->num_rows > 0) {
  while($row = $result->fetch_assoc()) {
  ?>
   <div class="song">
    <h3><?php echo $row['Title']; ?></h3>
    <p>Artist: <?php echo $row['Artist']; ?></p>
    <p>Album: <?php echo $row['Album']; ?></p>
 <audio controls>
 <source src="<?php echo $row['audio']; ?>" type="audio/mpeg">
```

```
 Your browser does not support the audio element.
 </audio>
</div>
<?php
}
} else {
echo "0 results";
}
} else {
echo "Please enter a search term.";
}
}
?>
</body>
</html>
```

Artist page

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,initial-scale=1.0">
<title>Artists</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<section class="header">
<nav>
<a href="home1.php"><img src=""></a>
<div class="nav-links">
<ul>
<li><a href="./home1.php">home</a></li>
<li><a href="./artists.php">artist</a></li>
<li><a href="./album.php">album</a></li>
<li><a href="./song.php">songs</a></li>
</ul>
</div>
<form action="" method="GET">
<div class="search">
<input class="search-input" type="search" name="search" placeholder="">
<input type="submit" value="Search">
</div>
```

```
</form>
</nav>
<div class="playlist">
Artists
</div>
<div class="box">
<?php
// Connect to your database
$servername = "localhost:3306";
$username = "root";
$password = "";
$dbname = "music";
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
// Initialize search term
$searchTerm = "";
// Check if search term is provided in the URL
if(isset($_GET['search'])) {
$searchTerm = $_GET['search'];
// Query to select artists based on search term
$sql = "SELECT * FROM artists WHERE Artist LIKE '%$searchTerm%'";
} else {
// Query to select all artists
$sql = "SELECT * FROM artists";
}
$result = $conn->query($sql);
// Display artists as cards
if ($result->num_rows > 0) {
while($row = $result->fetch_assoc()) {
 ?>
<div class="katera">
<a href="./navigate_artist.php?navigates=<?php echo $row['Artist'] ?>"><button>
<img src="<?php echo $row['Artist_cover']; ?>">
<li><h1 class="card-title"><?php echo $row['Artist']; ?></h1></li></button>
</a>
</div>
<?php
}
} else {
echo "0 results";
```

```
}
$conn->close();
?>
</div>
</section>
</body>
</html>
```

Album Page

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,initial-scale=1.0">
<title>album</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<section class="header">
<nav> <a href="home1.php"><img src=""></a>
<div class="nav-links">
<ul>
<li><a href="./home1.php">home</a></li>
<li><a href="./artists.php">artist</a></li>
<li><a href="./album.php">album</a></li>
<li><a href="./song.php">songs</a></li>
</ul>
</div>
<form action="" method="GET">
<div class="search">
<input class="search-input" type="search" name="search" placeholder="search">
<input type="submit" value="Search">
```

```
</div>
</form>
</nav>
<div class="playlist">Albums</div>
<div class="box2">
<?php
// Connect to your database
$servername = "localhost:3306";
$username = "root";
$password = "";
$dbname = "music";
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
// Initialize search term
$searchTerm = "";
// Check if search term is provided in the URL
if(isset($_GET['search'])) {
$searchTerm = $_GET['search'];
// Query to select albums based on search term
$sql = "SELECT * FROM albums WHERE Album LIKE '%$searchTerm%'";
} else {
// Query to select all albums
 $sql = "SELECT * FROM albums";
}
$result = $conn->query($sql);
// Display albums as cards
if ($result->num_rows > 0) {
```

```
while($row = $result->fetch_assoc()) {
?>
<div class="recent">
<a href="./navigate_album.php?navigate=<?php echo $row['Album'] ?>"><button>
<img src="<?php echo $row['Album_cover']; ?>">
<li><h1 class="card-title"><?php echo $row['Album']; ?></h1></li></button>
</a>
</div>
<?php
}
} else {
echo "0 results";
}
$conn->close();
?>
</div>
</section>
</body>
</html>
```

Songs Page

```
<?php
// Step 1: Connect to the database
$conn = mysqli_connect("localhost:3306", "root", "", "music");
// Check connection
if (!$conn) {
die("Connection failed: " . mysqli_connect_error());
}
// Step 2: Fetch audio files from the database
$sql = "SELECT Title, Artist, Album, audio FROM songs";
$result = mysqli_query($conn, $sql);
```

```
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Album</title>
</head>
<body style="background-color: indigo; color: magenta;">
<h1></h1>
<?php
// Step 3: Display audio players for each audio file fetched from the database
if (mysqli_num_rows($result) > 0) {
while ($row = mysqli_fetch_assoc($result)) {
 $Album = $row["Title"];
 $audio = $row["audio"];
 echo "<h3>$Album</h3>";
 echo "<audio controls>";
 echo "<source src='$audio' type='audio/mpeg'>";
 echo "Your browser does not support the audio element.";
 echo "</audio>";
 echo "<br>";
}
} else {
echo "0 results";
}
?>
<a href="home1.php" style="color: pink;">Back</a>
</body>
</html>
```

navigate_album

```
<?php
// Step 1: Connect to the database
$servername = "localhost:3306";
$username = "root";
$password = "";
$dbname = "music";
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
// Step 2: Fetch album details and album cover from the database
$album_name = $_GET['navigate'];
$sql_album = "SELECT * FROM albums WHERE Album = '$album_name'";
$res_album = mysqli_query($conn, $sql_album);
$album_row = mysqli_fetch_assoc($res_album);
// Step 3: Fetch audio files from the database for a specific album
$sql_songs = "SELECT * FROM songs WHERE Album = '$album_name'";
$res_songs = mysqli_query($conn, $sql_songs);
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Album</title>
</head>
<body style="background: linear-gradient(135deg, #667eea, #764ba2);background-position: center;
 background-size: cover;
 position: relative; color: magenta; text-align: center; ">
```

```
<h1><?php echo $album_name; ?></h1>
<?php           // Display album name and cover
if ($album_row) {
$album_cover = $album_row['Album_cover'];
echo "<img src='$album_cover' alt='$album_name' style='width: 400px; height: 300px;'>";
 } else {
 echo "Album details not found";
 }       // Step 4: Display audio players for each audio file fetched from the database
if (mysqli_num_rows($res_songs) > 0) {
while ($row = mysqli_fetch_assoc($res_songs)) {
 $audio = $row["audio"];
 $title = $row["Title"];
 echo "<p>$title</p>";
 echo "<audio controls>";
 echo "<source src='$audio' type='audio/mpeg'>";
 echo "Your browser does not support the audio element.";
 echo "</audio>";
 echo "<br>";
 }
} else {
echo "No songs found for this album";
}
?>
<a href="album.php" style="color: pink;">Back</a>
</body>
</html>
<?php
// Close the database connection
mysqli_close($conn);
?>
```

navigate_artist

```
<?php
// Step 1: Connect to the database
$servername = "localhost:3306";
$username = "root";
$password = "";
$dbname = "music";
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}        // Step 2: Fetch audio files from the database for a specific album
$album_name = $_GET['navigates'];
$sql = "SELECT * FROM albums WHERE Album = '$album_name'";
$res = mysqli_query($conn, $sql);
$sql_artist = "SELECT * FROM artists WHERE Artist = '$album_name'";
$res_artist = mysqli_query($conn, $sql_artist);
$artist_row = mysqli_fetch_assoc($res_artist);
$sql_songs = "SELECT a.Artist, a.no_songs, b.audio, b.Title
 FROM artists a  INNER JOIN songs b ON a.Artist = b.Artist
 WHERE a.Artist = '$album_name'";
$res_songs = mysqli_query($conn, $sql_songs);
?>
<!DOCTYPE html>
<html lang="en">
<head> <meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Artists</title> </head>
<body style="background: linear-gradient(135deg, #667eea, #764ba2);background-position: center;
  background-size: cover;
```

```
  position: relative; color: magenta; text-align: center; ">
<?php       // Step 3: Display artist name and photo
if ($artist_row) {
$artist_photo = $artist_row['Artist_cover'];
echo "<h3>$album_name</h3>";
echo "<img src='$artist_photo' alt='$album_name' style='width: 200px; height: auto;'>";
 } else {
 echo "Artist details not found";
 }       // Step 4: Display audio players for each audio file fetched from the database
if (mysqli_num_rows($res_songs) > 0) {
while ($row = mysqli_fetch_assoc($res_songs)) {
 $audio = $row["audio"];
 $title = $row["Title"];
 echo "<p>$title</p>";
 echo "<audio controls>";
 echo "<source src='$audio' type='audio/mpeg'>";
 echo "Your browser does not support the audio element.";
 echo "</audio>";
 echo "<br>";
}
} else {
echo "No songs found for this artist";
}
?>
<a href="artist.php" style="color: pink;">Back</a>
</body>
</html>
<?php     // Close the database connection
mysqli_close($conn);
?>
```

Dashboard

```
<?php
session_start();
if (!isset($_SESSION['loggedin'])||$_SESSION['loggedin']!=true) {
 header("location: admin_signin.php");
 exit;
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Music Database Dashboard</title>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>
<style>
 body {
  margin: 0;
  overflow:scroll;
  padding: 0;
  justify-content:center;
  align-items: top;
  background: linear-gradient(135deg, #667eea, #764ba2);
  font-family: Arial, sans-serif;
  }
```

```
.navbar {
background: linear-gradient(135deg, #ea6666, #764ba2);
color: white;
padding: 10px 20px;
display: flex;
justify-content: space-between;
align-items: center;
position: sticky;
top:0px;
}
.navbar h1 {
 margin: 0;
}
.sidebar {
position: fixed;
left: 0;
top: 60px;
height: calc(100% - 60px);
width: 250px;
background: linear-gradient(135deg, #ea6666, #764ba2);
padding-top: 20px;
transition: all 0.3s;
}
.sidebar a {
display: block;
color: white;
padding: 10px 20px;
text-decoration: none;
transition: background 0.3s;
}
```

```
.sidebar a:hover {
background: #555;
}
.content {
margin-left: 250px;
padding: 20px;
transition: all 0.3s;
}h2 {
margin-top: 0;
}
table {
width: 100%;
border-collapse: collapse;
margin-top: 20px;
}
table, th, td {
border: 1px solid #ddd;
padding: 8px;
text-align: left;
}
th {
background-color: #f2f2f2;
}
</style>
</head>
<body>
<div class="navbar">
<h1>Music Database Dashboard</h1>
<div>
<span>Welcome, Admin</span>
```

```
  <a href="./logout.php">Logout</a>
  </div>
  </div>
  <div class="sidebar" id="sidebar">
<a href="./dashboard.php" class="active"><i class="fas fa-database"></i> Database</a>
<a href="./admin_songs.php"><i class="fas fa-music"></i> Songs</a>
<a href="./admin_albums.php"><i class="fas fa-compact-disc"></i> Albums</a>
<a href="./admin_artists.php"><i class="fas fa-user"></i> Artists</a>
<a href="./admin_users.php"><i class="fas fa-users"></i> Users</a>
</div>
<div class="content" id="content">
<h2>Database Overview</h2>
<p>Provide an overview of the music database here.</p>
<!-- <div class="row"> -->
<div class="col-sm-6">
<div class="card">
<div class="card-body">
<h1 class="card-title">
<?php
include "connection.php";
$sql = "SELECT COUNT(*) AS song_count FROM songs;";
$res=mysqli_query($conn, $sql);
$row=mysqli_fetch_assoc($res);
echo $row['song_count'];
?>
</h1>
<p class="card-text"><h5>Songs count</h5></p>
<a href="./admin_songs.php" class="btn btn-primary">view songs</a>
</div>
</div>
```

```
</div>
<div class="col-sm-6">
<div class="card">
<div class="card-body">
<h1 class="card-title">
<?php
include "connection.php";
$sql = "SELECT COUNT(*) AS album_count FROM albums;";
$res=mysqli_query($conn, $sql);
$row=mysqli_fetch_assoc($res);
echo $row['album_count'];
?>
</h1>
<p class="card-text"><h5>Albums Counts</h5></p>
<a href="./admin_albums.php" class="btn btn-primary">view Albums</a>
</div>
</div>
</div>
<!-- </div> -->
<h2></h2>
<!-- <div class="row"> -->
<div class="col-sm-6">
<div class="card">
<div class="card-body">
<h1 class="card-title">
<?php
include "connection.php";
$sql = "SELECT COUNT(*) AS artist_count FROM artists;";
$res=mysqli_query($conn, $sql);
$row=mysqli_fetch_assoc($res);
```

```
echo $row['artist_count'];
?>
</h1> <p class="card-text"><h5>Artist count</h5></p>
<a href="./admin_artists.php" class="btn btn-primary">view Artists</a>
</div>
</div>
</div>
<div class="col-sm-6">
<div class="card">
<div class="card-body">
<h1 class="card-title">
<?php
include "connection.php";
$sql = "SELECT COUNT(*) AS user_count FROM user_login;";
$res=mysqli_query($conn, $sql);
$row=mysqli_fetch_assoc($res);
echo $row['user_count'];
?>
</h1> <p class="card-text"><h5>Users Counts</h5></p>
<a href="./admin_users.php" class="btn btn-primary">view Users</a>
</div>
</div>
</div>
<!-- </div> -->
<!-- <script>
document.getElementById('sidebar').addEventListener('click', function(e) {
var links = document.querySelectorAll('.sidebar a');
for (var i = 0; i < links.length; i++) {
links[i].classList.remove('active');
}
```

```
e.target.classList.add('active');
});
</script> -->
</div>
</body>
</html>
```

admin_upload_songs

```
<?php
session_start();
if (!isset($_SESSION['loggedin']) || $_SESSION['loggedin'] != true) {
header("location: admin_signin.php");
exit;
}
if ($_SERVER["REQUEST_METHOD"] == "POST") {
include "connection.php";
$title = $_POST["Title"];
$artist = $_POST["Artist"];
$album = $_POST["Album"];
$target_dir = "audio/";
$audio_file = $target_dir . basename($_FILES["audio"]["name"]);
move_uploaded_file($_FILES["audio"]["tmp_name"], $audio_file);
$sql = "INSERT INTO songs (title, artist, album, audio) VALUES ('$title', '$artist', '$album', '$audio
_file')";
if (mysqli_query($conn, $sql)) {
header("Location: admin_songs.php");
exit;
} else {
echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}
mysqli_close($conn);
```

```
}
?>
```

admin_add_songs

```
<?php
session_start();
if (!isset($_SESSION['loggedin'])||$_SESSION['loggedin']!=true) {
header("location: admin_signin.php");
exit;
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Add Song</title>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">
<style>
body {
margin: 0;
padding: 0;
height: 100vh;
justify-content:center;
align-items: top;
background: linear-gradient(135deg, #667eea, #764ba2);
overflow: hidden;
font-family: Arial, sans-serif;
}
.navbar {
```

```
background: #333;
color: white;
padding: 10px 20px;
display: flex;
justify-content: space-between;
align-items: center;
}
.navbar h1 {
margin: 0;
}
.form-container {
max-width: 500px;
margin: 20px auto;
padding: 20px;
background: #fff;
border-radius: 8px;
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}
form {
display: flex;
flex-direction: column;
align-items: center;
}
label {
margin-bottom: 10px;
}
input[type="text"],
input[type="file"] {
width: 100%;
padding: 8px;
```

```
margin-bottom: 10px;
border: 1px solid #ccc;
border-radius: 4px;
box-sizing: border-box;
}
button {
padding: 8px 16px;
background-color: #333;
color: white;
border: none;
border-radius: 4px;
cursor: pointer;
}
button:hover {
background-color: #555;
}
</style>
</head>
<body>
<div class="navbar">
<h1>Add/Edit Song</h1>
<a href="./admin_songs.php" style="color: white; text-decoration: none;"><i class="fas fa-arrow-left"></i> Back to Songs</a>
</div>
<div class="form-container">
<h2>Add Song</h2>
<form action="admin_upload_song.php" method="post" enctype="multipart/form-data">
<label for="title">Title:</label>
<input type="text" id="title" name="Title" required>
<label for="audio">Audio:</label>
```

```
<input type="file" id="audio" name="audio" accept=".mp3" required>
<label for="artist">Artist:</label>
<select id="artist" name="Artist" required>
<?php
include "connection.php";
// Fetch artist names from the database
$sql = "SELECT * FROM artists;";
$res = mysqli_query($conn, $sql);
// Check if there are any results
if (mysqli_num_rows($res) > 0) {
// Output data of each row
while ($row = mysqli_fetch_assoc($res)) {
echo "<option value="' . $row['Artist'] . '">' . $row['Artist'] . "</option>";
}
} else {
echo "<option value="">No artists found</option>";
}
// Close connection
mysqli_close($conn);
?>
</select>
<label for="album">Album:</label>
<select id="album" name="Album" required>
<?php
include "connection.php";
// Fetch album names from the database
$sql = "SELECT * FROM albums;";
$res = mysqli_query($conn, $sql);
// Check if there are any results
if (mysqli_num_rows($res) > 0) {
```

```
// Output data of each row
while ($row = mysqli_fetch_assoc($res)) {
echo "<option value="' . $row['Album'] . '">" . $row['Album'] . "</option>";
}
} else {
echo "<option value="">No albums found</option>";
}
// Close connection
mysqli_close($conn);
?>
</select>
<input type="submit" value="Add Song" name="submit">
</form>
</div>
<?php
if(isset($_GET['Error']) && $_GET['Error']){
?>
<script>
alert("Audio uploading failed");
</script>
<?php
}
?>
</body>
</html>
```

Admin_edit_song

```
<?php
session_start();
if (!isset($_SESSION['loggedin'])||$_SESSION['loggedin']!=true) {
header("location: admin_signin.php");
```

```
exit;
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Edit Song</title>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.4/css/all.min.css">
<style>
body {
margin: 0;
padding: 0;
height: 100vh;
justify-content: center;
align-items: top;
background: linear-gradient(135deg, #667eea, #764ba2);
overflow: hidden;
font-family: Arial, sans-serif;
}
.navbar {
background: #333;
color: white;
padding: 10px 20px;
display: flex;
justify-content: space-between;
align-items: center;
}
```

```
.navbar h1 {
margin: 0;
}
.form-container {
max-width: 500px;
margin: 20px auto;
padding: 20px;
background: #fff;
border-radius: 8px;
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}
form {
display: flex;
flex-direction: column;
align-items: center;
}
label {
margin-bottom: 10px;
}
input[type="text"] {
width: 100%;
padding: 8px;
margin-bottom: 10px;
border: 1px solid #ccc;
border-radius: 4px;
box-sizing: border-box;
}
button {
padding: 8px 16px;
background-color: #333;
```

```
color: white;
border: none;
border-radius: 4px;
cursor: pointer;
}
button:hover {
background-color: #555;
}
</style>
</head>
<body>
<div class="navbar">
<h1>Edit Song</h1>
<a href="./admin_songs.php" style="color: white; text-decoration: none;"><i class="fas fa-arrow-left"></i> Back to
Songs</a>
</div>
<div class="form-container">
<h2>Edit Song</h2>
<?php
include "connection.php";
$title = $_GET['title'];
$sql = "SELECT * from songs where Title = '$title'";
$res = mysqli_query($conn, $sql);
$row = mysqli_fetch_assoc($res);
$title = $row['Title'];
$Artist = $row['Artist'];
$Album = $row['Album'];
$audio = $row['audio'];
?>
```

```
<form action="admin_edit_song.php" method="post" enctype="multipart/form-data">
<label for="title">Edit Title:</label>
<input type="text" id="title" name="Title" value="<?php echo $title ?>" required>
<input type="hidden" id="title" name="title" value="<?php echo $title ?>" required>
<label for="artist">Edit Artist:</label>
<input type="text" id="Artist" name="Artist" value="<?php echo $Artist ?>" required>
<label for="album">Edit Album:</label>
<input type="text" id="Album" name="Album" value="<?php echo $Album ?>" required>
<label for="album">Edit Audio:</label>
<input type="text" id="audio" name="audio" value="<?php echo $audio ?>" required>
<input type="submit" value="Edit Song" name="submit">
</form>
<?php
if (isset($_POST['submit'])) {
$Title = $_POST['Title'];
$title = $_POST['title'];
$Artist = $_POST['Artist'];
$Album = $_POST['Album'];
$audio = $_POST['audio'];
$sql = "UPDATE songs SET Title = '$Title', Artist = '$Artist', Album = '$Album', audio = '$audio'
WHERE Title = '$title';";
$res = mysqli_query($conn, $sql);
if ($res) {
header("Location: admin_songs.php?Edited=true?$sql");
} else {
header("Location: admin_editing_song.php?Error=true");
}
}
?>
</div>
```

```
<?php
if (isset($_GET['Error']) && $_GET['Error']) {
?>
<script>
alert("Audio uploading failed");
</script>
<?php
}
?>
</body>
</html>
```

Admin_delete_songs

```
<?php
session_start();
if (!isset($_SESSION['loggedin'])||$_SESSION['loggedin']!=true) {
header("location: admin_signin.php");
exit;
}
?>
<?php
include "connection.php";
$title = $_GET['title'];
$sql = "DELETE FROM songs WHERE `songs`.`Title` = '$title';";
$res = mysqli_query($conn, $sql);
if ($res) {
header("Location: admin_songs.php?Deleted=true?$sql");
} else {
header("Location: admin_songs.php?Error=true");
}
?>
```

5.2 Backend

```
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Mar 17, 2024 at 11:52 AM
-- Server version: 10.4.32-MariaDB
-- PHP Version: 8.0.30

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
--
-- Database: `music`
--
-- --------------------------------------------------------
--
-- Table structure for table `admin_login`
--
CREATE TABLE `admin_login` (
`username` varchar(20) NOT NULL,
`password` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
--
-- Dumping data for table `admin_login`
--
INSERT INTO `admin_login` (`username`, `password`) VALUES
('admin1', '12345'),
('admin2', '23456'),
('admin3', '34567');
-- --------------------------------------------------------
--
-- Table structure for table `albums`
--
CREATE TABLE `albums` (
`Album` varchar(20) NOT NULL,
`no_songs` int(11) DEFAULT NULL,
`dates` varchar(10) DEFAULT NULL,
`Album_cover` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
--
-- Dumping data for table `albums`
--
INSERT INTO `albums` (`Album`, `no_songs`, `dates`, `Album_cover`) VALUES
('ek-love-ya', 2, '2021', 'images\\ek love ya.jpg'),
('kantara', 3, '2022', 'images\\Kantara pos.jpg'),
('katera', 3, '2023', 'images\\katera pos.jpg'),
('KGF2', 2, '2022', 'images\\kgf2.jpg'),
('Kiss', 3, '2020', 'images\\kiss pos.jpg'),
('Seetharama Kalyana', 1, '2018', 'images/seetha rama pos.jpg'),
('Unknown Album', NULL, '-', 'images/unknown albm.jpg'),
('Vikranthrona', 3, '2021', 'images\\vikaranth pos.jpg');
-- --------------------------------------------------------
```

```
--
-- Table structure for table `artists`
--
CREATE TABLE `artists` (
  `Artist` varchar(20) NOT NULL,
  `no_songs` int(11) DEFAULT NULL,
  `Artist_cover` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
--
-- Dumping data for table `artists`
--
INSERT INTO `artists` (`Artist`, `no_songs`, `Artist_cover`) VALUES
('Ananya Bhat', 3, 'images\\ananya.jpg'),
('Arjun Janya', 3, 'images\\arjun.jpg'),
('Armaan Malik', 1, 'images\\arman.jpg'),
('Chandan Shetty', 1, 'images\\chandan.jpg'),
('Sanjeeth Hegde', 1, 'images\\sanjeeth.jpg'),
('Shreya Ghoshal', 1, 'images\\shreya.jpg'),
('Sonu Nigam', 2, 'images\\sonu.jpg'),
('unknown', 3, 'images\\unknown.jpg'),
('Vijay Prakash', 2, 'images\\vijay.jpg');
-- --------------------------------------------------------
--
-- Table structure for table `songs`
--
CREATE TABLE `songs` (
`Title` varchar(20) NOT NULL,
`Artist` varchar(20) NOT NULL,
`Album` varchar(20) NOT NULL,
`audio` varchar(300) NOT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--

-- Dumping data for table `songs`

--

INSERT INTO `songs` (`Title`, `Artist`, `Album`, `audio`) VALUES

('Chikki Bombe', 'Vijay Prakash', 'Vikranthrona', 'audio\\Chikki Bombe.mp3'),

('Gagana-Nee', 'unknown', 'KGF2', 'audio\\Gagana-Nee-Suchetha-Basrur.mp3'),

('I-Love-You-Idiot', 'Sanjeeth Hegde', 'Kiss', 'audio\\I-Love-You-Idiot-Sanjith-Hegde.mp3'),

('Karma Song', 'unknown', 'kantara', 'audio\\Karma Song(RaagaBeat.In)-KannadaMaza.Com.mp3'),

('katera Title', 'unknown', 'katera', 'audio\\katera Title.mp3'),

('Lullaby Song', 'Vijay Prakash', 'Vikranthrona', 'audio\\Lullaby Song.mp3'),

('Matthe Nodabeda', 'Arjun Janya', 'ek-love-ya', 'audio\\Matthe Nodabeda.mp3'),

('Meet Madana Illa Dat', 'Arjun Janya', 'ek-love-ya', 'audio\\Meet Madana Illa Date Madana.mp3'),

('Mehabooba', 'Ananya Bhat', 'KGF2', 'audio\\Mehabooba.mp3'),

('Neene-Modalu', 'Shreya Ghoshal', 'Kiss', 'audio\\Neene-Modalu-Shreya-Ghoshal.mp3'),

('Ninna raja nanu', 'Armaan Malik', 'Seetharama Kalyana', 'audio/[iSongs.info] 01 - Ninna Raja Naanu
Nanna Rani Neenu.mp3'),

('pasandaagavne', 'Ananya Bhat', 'katera', 'audio\\pasandaagavne.mp3'),

('Ra Ra Rakkamma', 'Sonu Nigam', 'Vikranthrona', 'audio\\Ra Ra Rakkamma.mp3'),

('Sheela-Susheela', 'Chandan Shetty', 'Kiss', 'audio\\Sheela-Susheela-Chandan-Shetty.mp3'),

('Singara Siriye', 'Ananya Bhat', 'kantara', 'audio\\Singara Siriye.mp3'),

('Varaha Roopam Daiva ', 'Arjun Janya', 'kantara', 'audio\\Varaha Roopam Daiva Rihstam-
KannadaMaza.Com.mp3'),

('Yava Janumada Gelath', 'Sonu Nigam', 'katera', 'audio\\Yava Janumada Gelathi.mp3');

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

--

-- Table structure for table `user_login`

--

CREATE TABLE `user_login` (

  `id` varchar(20) NOT NULL,

  `email` varchar(20) NOT NULL,
```

```
  `pass` varchar(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
--
-- Dumping data for table `user_login`
--
INSERT INTO `user_login` (`id`, `email`, `pass`) VALUES
('chandana', 'chandu@gmail.com', 'chandu123'),
('chandrika', 'chandrika@gmail.com', '123123'),
('darshan', 'darshan.gmail.com', 'darshan'),
('deeksha', 'dee@gmail.com', '123deeksha'),
('impana', 'abc@gmail.com', '123456');
--
-- Indexes for dumped tables
--
--
-- Indexes for table `admin_login`
--
ALTER TABLE `admin_login`
ADD PRIMARY KEY (`username`);
--
-- Indexes for table `albums`
--
ALTER TABLE `albums`
ADD PRIMARY KEY (`Album`);
--
-- Indexes for table `artists`
--
ALTER TABLE `artists`
ADD PRIMARY KEY (`Artist`);
```

```
--
-- Indexes for table `songs`
--
ALTER TABLE `songs`
ADD PRIMARY KEY (`Title`),
ADD KEY `Album` (`Album`),
ADD KEY `Artist` (`Artist`);
--
-- Indexes for table `user_login`
--
ALTER TABLE `user_login`
  ADD PRIMARY KEY (`id`);
--
-- Constraints for dumped tables
--
--
-- Constraints for table `songs`
--
ALTER TABLE `songs`
  ADD CONSTRAINT `songs_ibfk_1` FOREIGN KEY (`Album`) REFERENCES `albums` (`Album`) ON UPDATE CASCADE,
  ADD CONSTRAINT `songs_ibfk_2` FOREIGN KEY (`Artist`) REFERENCES `artists` (`Artist`) ON UPDATE CASCADE;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Table 5.2.1 Details of admin_login Table

username	password
admin1	12345
admin2	23456
admin3	34567
admin4	45678

Table 5.2.2 Details of user_login Table

id	email	pass
chandana	chandu@gmail.com	chandu123
chandrika	chandrika@gmail.com	123123
darshan	darshan.gmail.com	darshan
deeksha	dee@gmail.com	123deeksha
impana	abc@gmail.com	123456

Table 5.2.3 Details of Artist Table

Artist	no_songs	Artist_cover
Ananya Bhat	3	images\ananya.jpg
Arjun Janya	3	images\arjun.jpg
Armaan Malik	1	images\arman.jpg
Chandan Shetty	1	images\chandan.jpg
Sanjeeth Hegde	1	images\sanjeeth.jpg
Shreya Ghoshal	1	images\shreya.jpg
Sonu Nigam	2	images\sonu.jpg
unknown	3	images\unknown.jpg
Vijay Prakash	2	images\vijay.jpg

Table 5.2.4 Details of Album Table

Album	no_songs	dates	Album_cover
ek-love-ya	2	2021	images\ek love ya.jpg
kantara	3	2022	images\Kantara pos.jpg
katera	3	2023	images\katera pos.jpg
KGF2	2	2022	images\kgf2.jpg
Kiss	3	2020	images\kiss pos.jpg
Seetharama Kalyana	1	2018	images/seetha rama pos.jpg
Unknown Album	*NULL*	-	images/unknown albm.jpg
Vikranthrona	3	2021	images\vikaranth pos.jpg

Table 5.2.5 Details of Songs Table

Title	Artist ▲ 1	Album	audio
Mehabooba	Ananya Bhat	KGF2	audio\Mehabooba.mp3
pasandaagavne	Ananya Bhat	katera	audio\pasandaagavne.mp3
Singara Siriye	Ananya Bhat	kantara	audio\Singara Siriye.mp3
Matthe Nodabeda	Arjun Janya	ek-love-ya	audio\Matthe Nodabeda.mp3
Meet Madana Illa Dat	Arjun Janya	ek-love-ya	audio\Meet Madana Illa Date Madana.mp3
Varaha Roopam Daiva	Arjun Janya	kantara	audio\Varaha Roopam Daiva Rihstam-KannadaMaza.Com....
Ninna raja nanu	Armaan Malik	Seetharama Kalyana	audio/[iSongs.info] 01 - Ninna Raja Naanu Nanna Ra...
Sheela-Susheela	Chandan Shetty	Kiss	audio\Sheela-Susheela-Chandan-Shetty.mp3
I-Love-You-Idiot	Sanjeeth Hegde	Kiss	audio\I-Love-You-Idiot-Sanjith-Hegde.mp3
Neene-Modalu	Shreya Ghoshal	Kiss	audio\Neene-Modalu-Shreya-Ghoshal.mp3
Ra Ra Rakkamma	Sonu Nigam	Vikranthrona	audio\Ra Ra Rakkamma.mp3
Yava Janumada Gelath	Sonu Nigam	katera	audio\Yava Janumada Gelathi.mp3
Gagana-Nee	unknown	KGF2	audio\Gagana-Nee-Suchetha-Basrur.mp3
Karma Song	unknown	kantara	audio\Karma Song(RaagaBeat.In)-KannadaMaza.Com.mp3
katera Title	unknown	katera	audio\katera Title.mp3
Chikki Bombe	Vijay Prakash	Vikranthrona	audio\Chikki Bombe.mp3
Lullaby Song	Vijay Prakash	Vikranthrona	audio\Lullaby Song.mp3

6.Snapshots

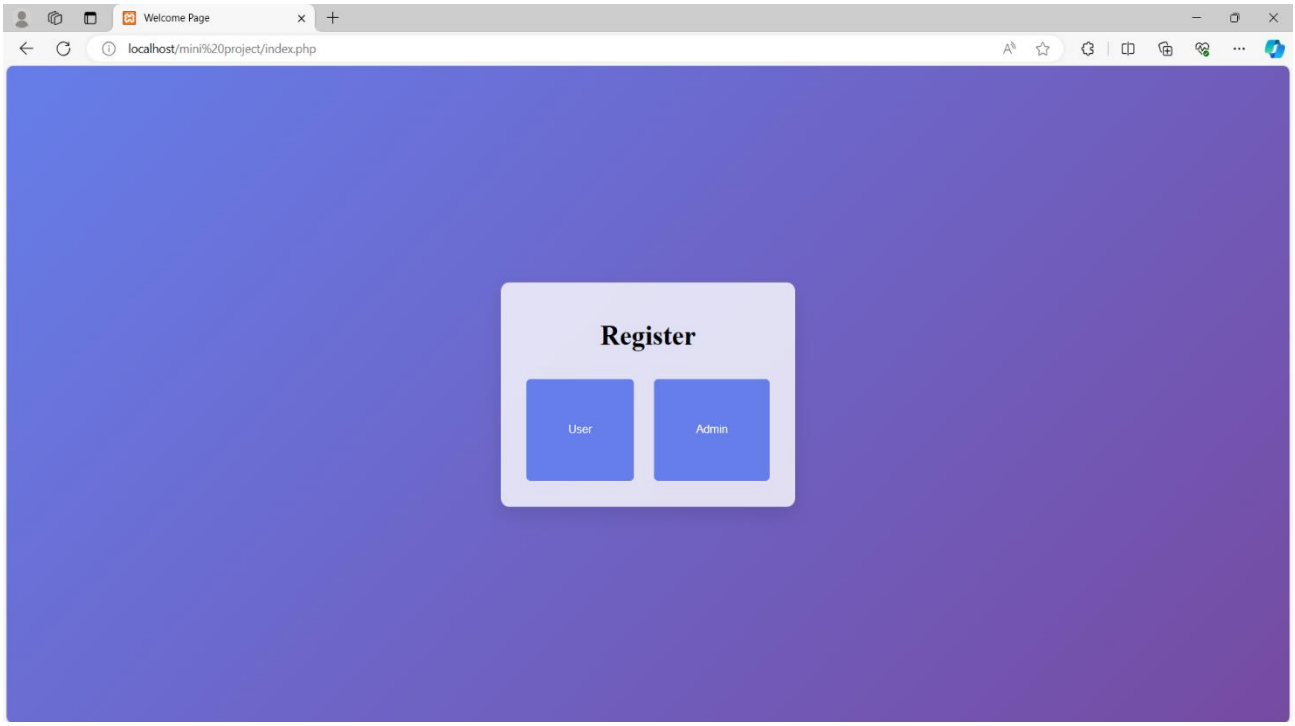

Figure 6.1 Home page

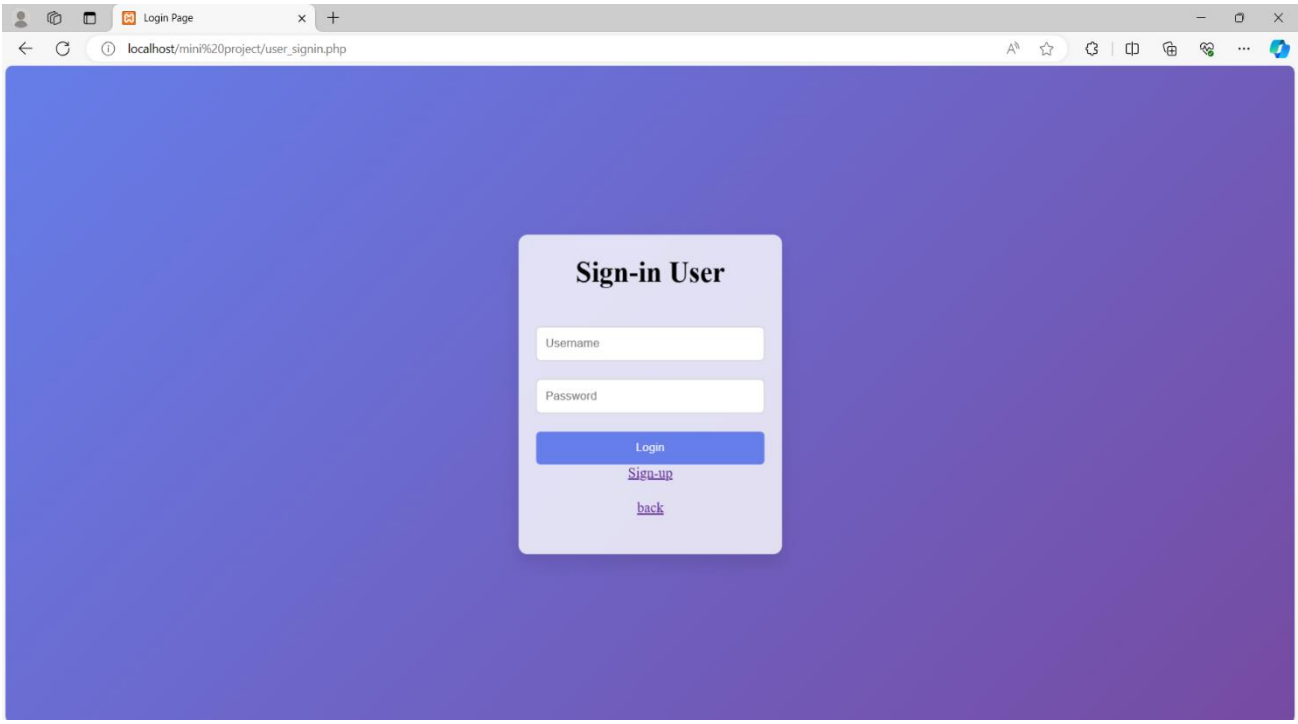

Figure 6.2 Sign-in page

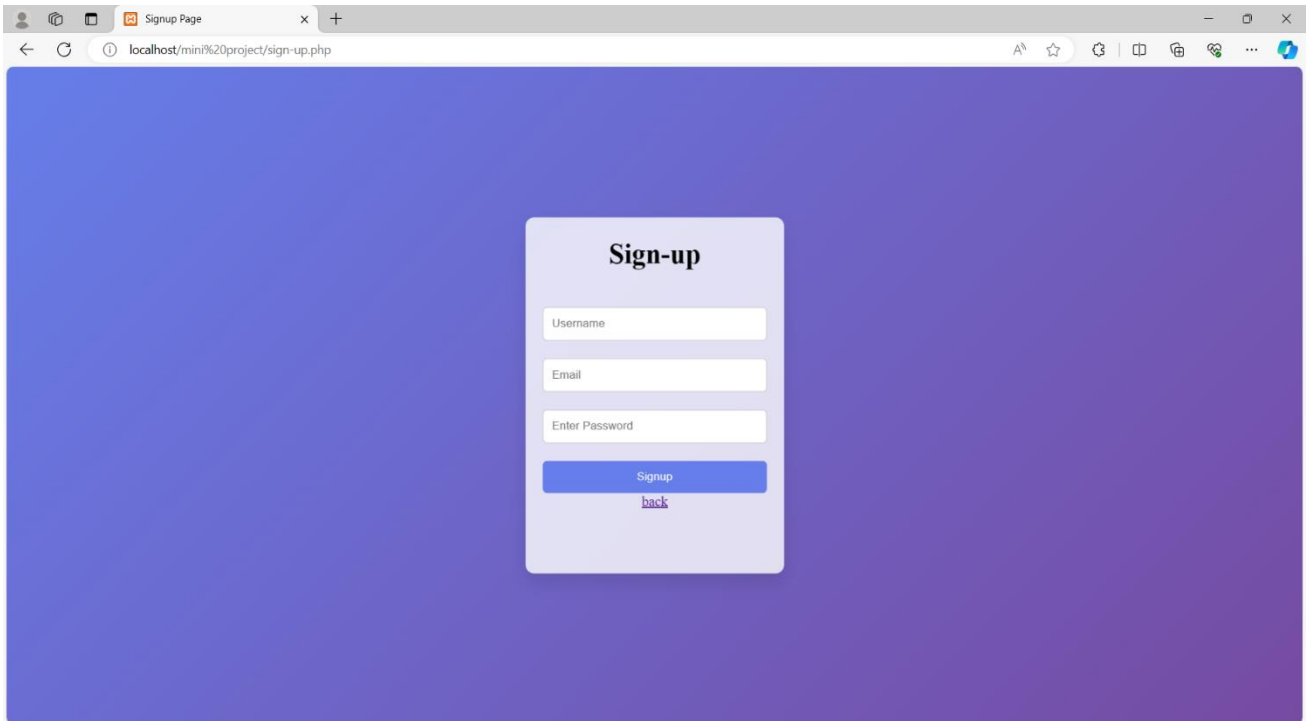

Figure 6.3 Sign-up page

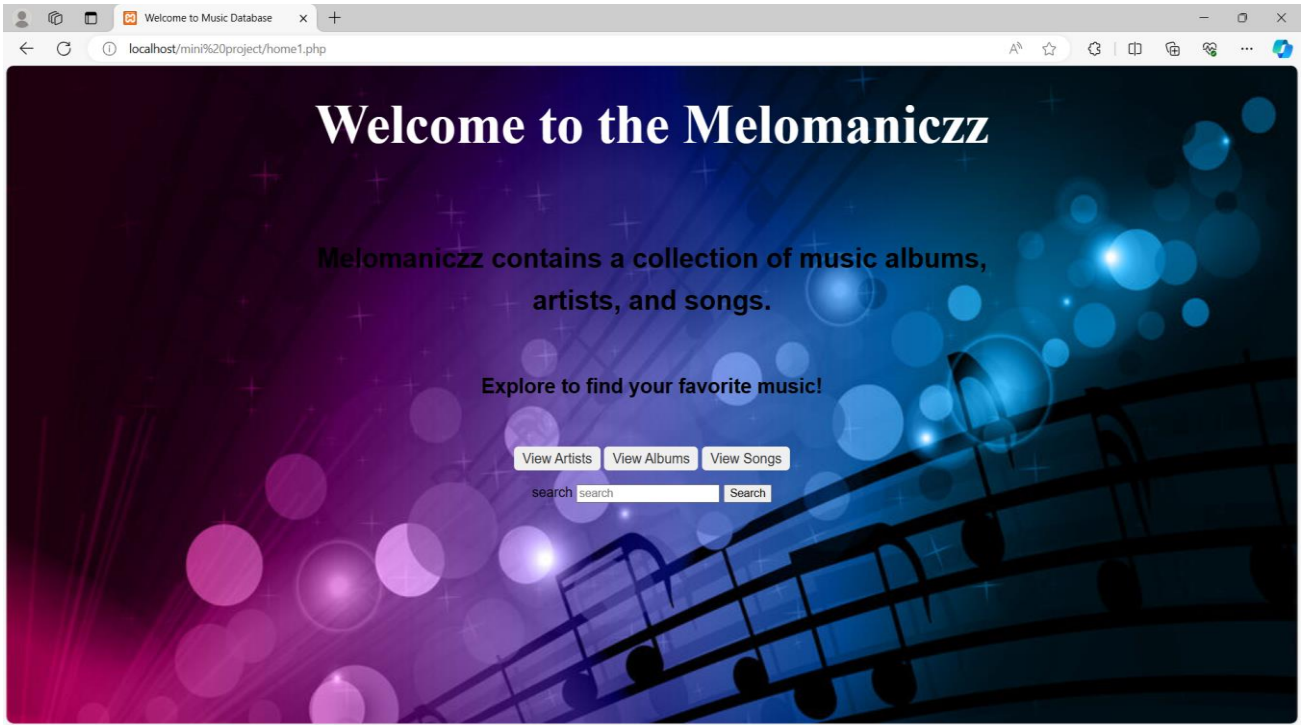

Figure 6.4 Welcome page

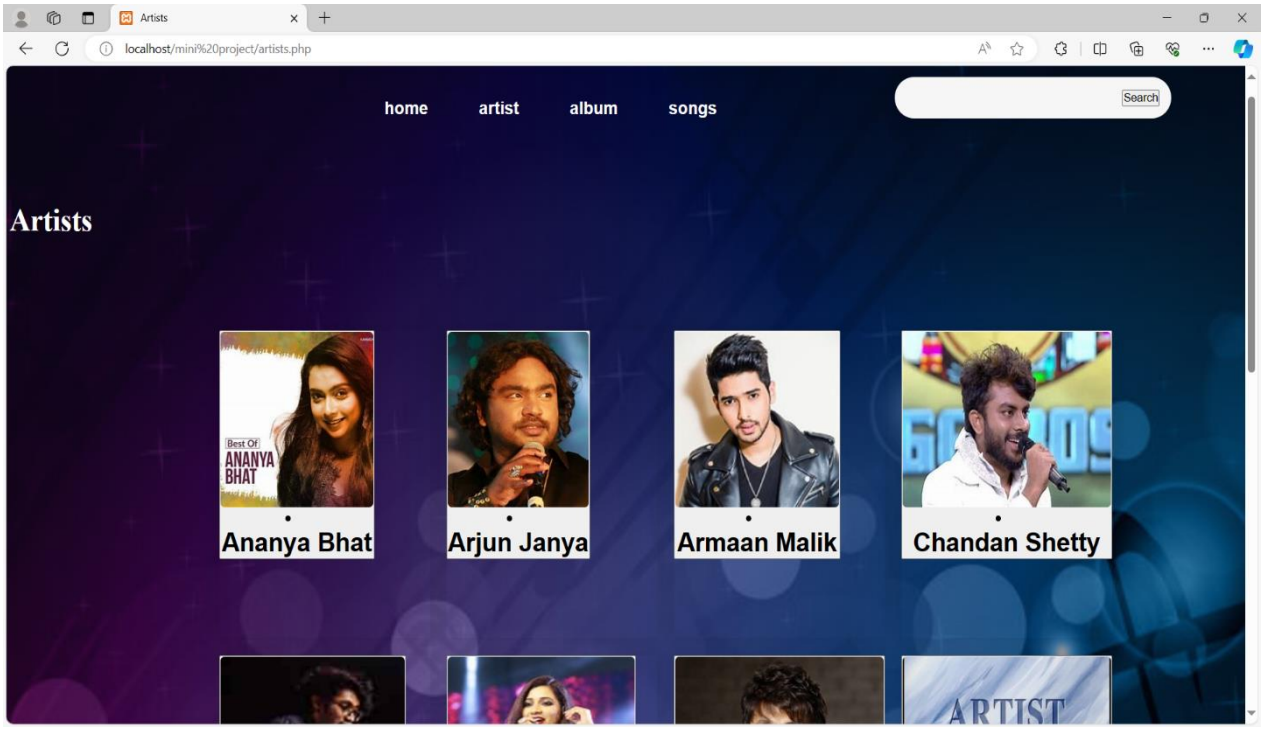

Figure 6.5 Artists page

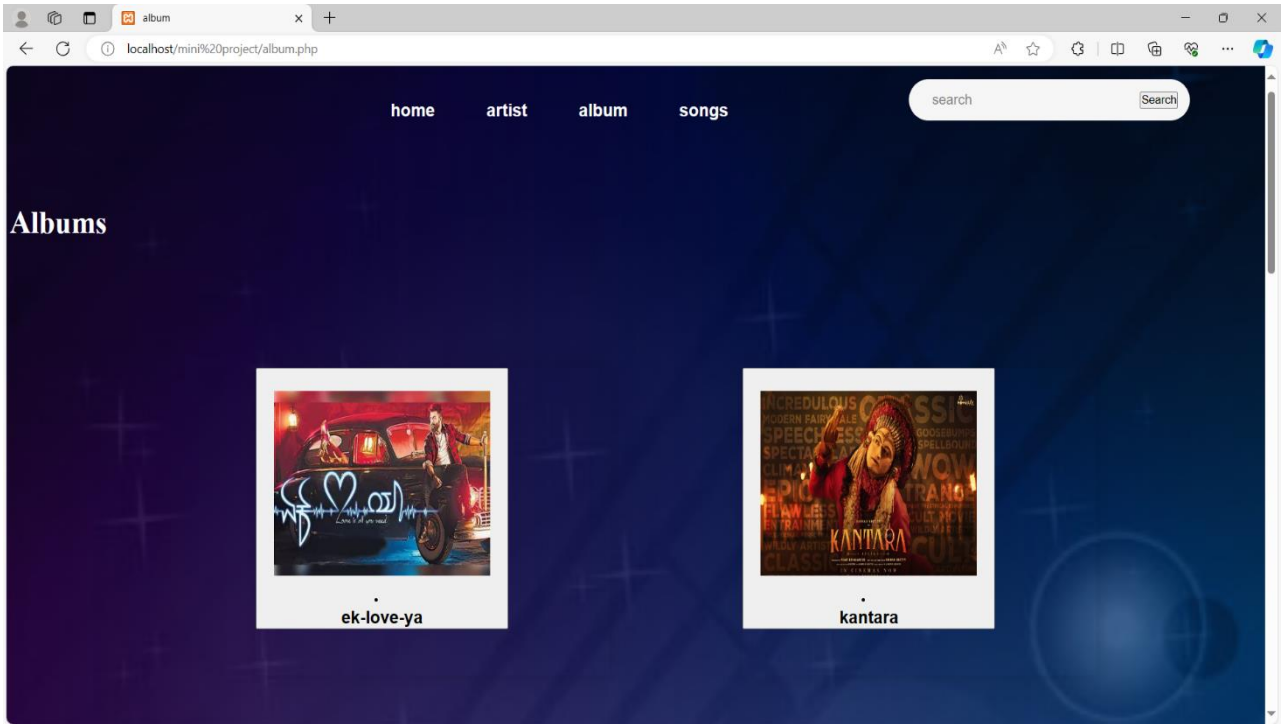

Figure 6.6 Albums page

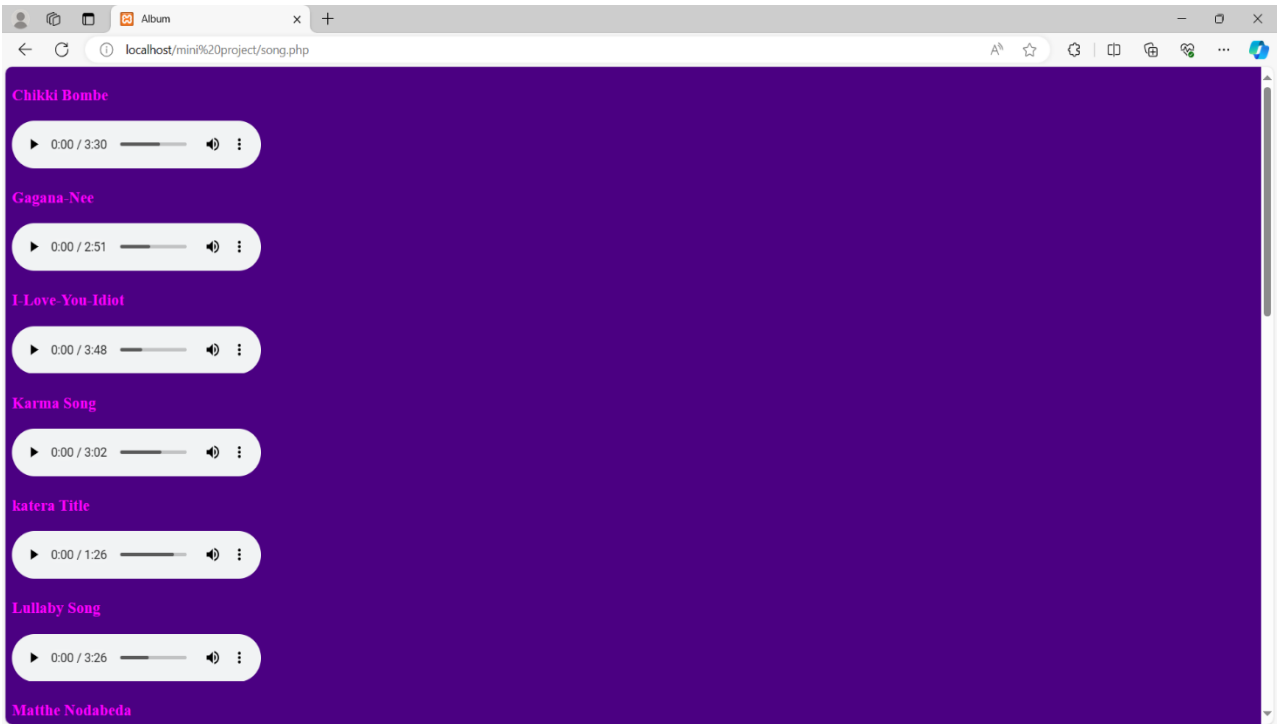

Figure 6.7 Songs page

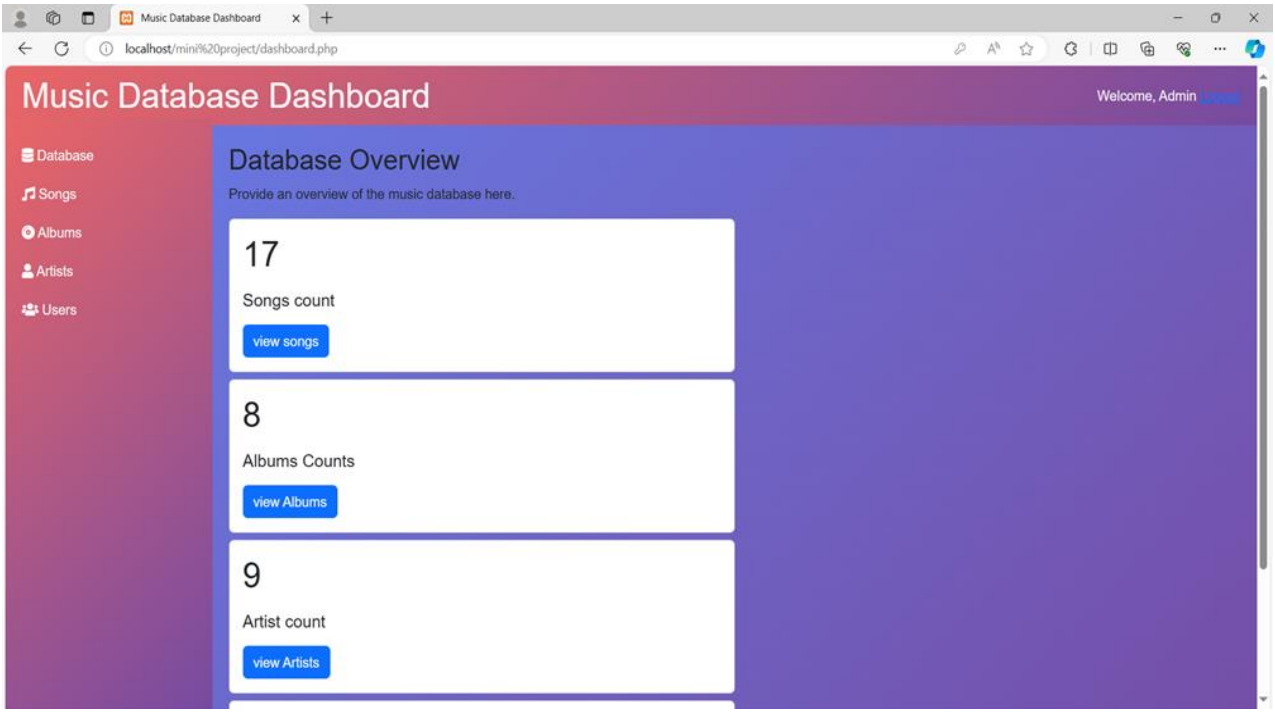

Figure 6.8 Admin Dashboard

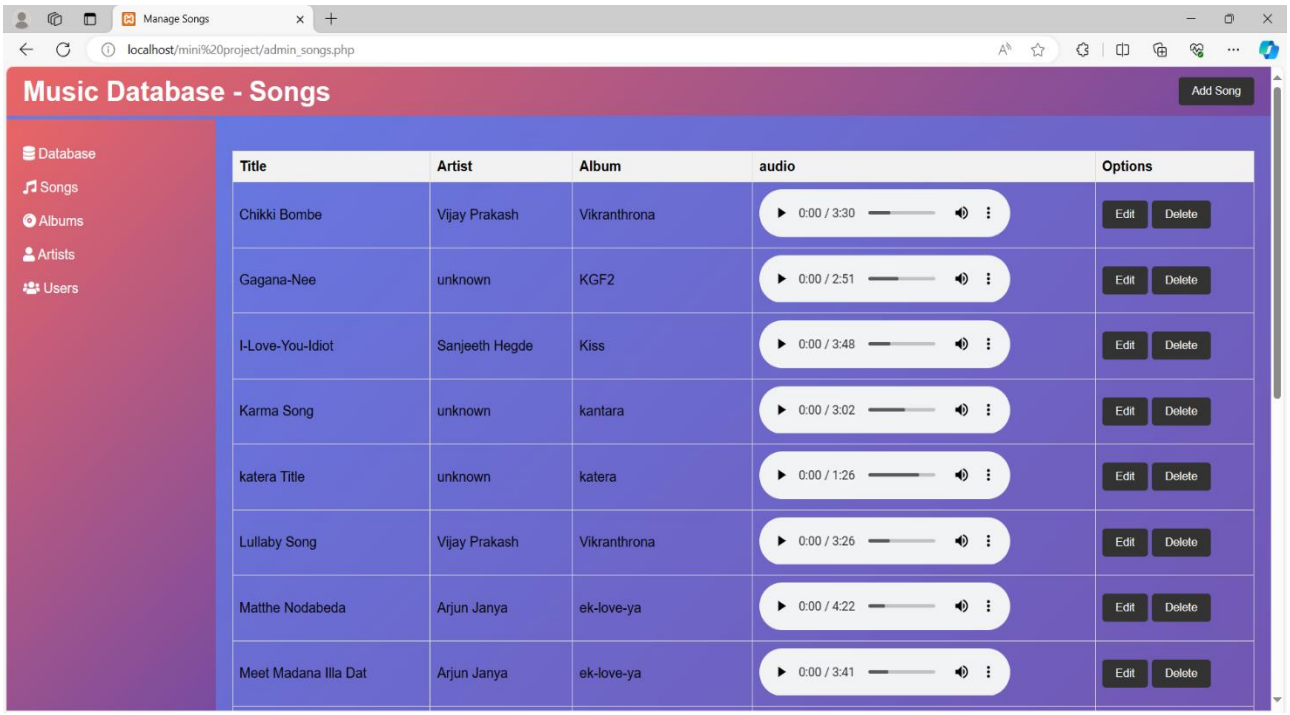

Figure 6.9 Dashboard-Songs page

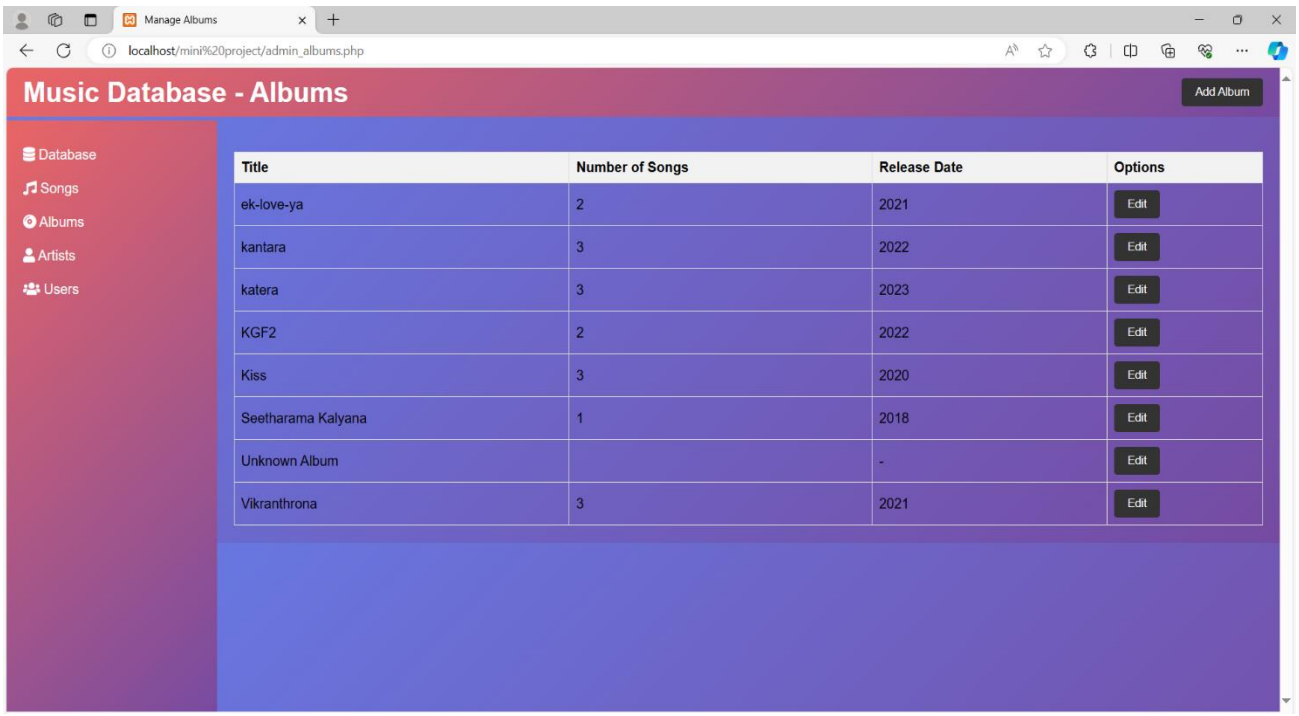

Figure 6.10 Dashboard-Album page

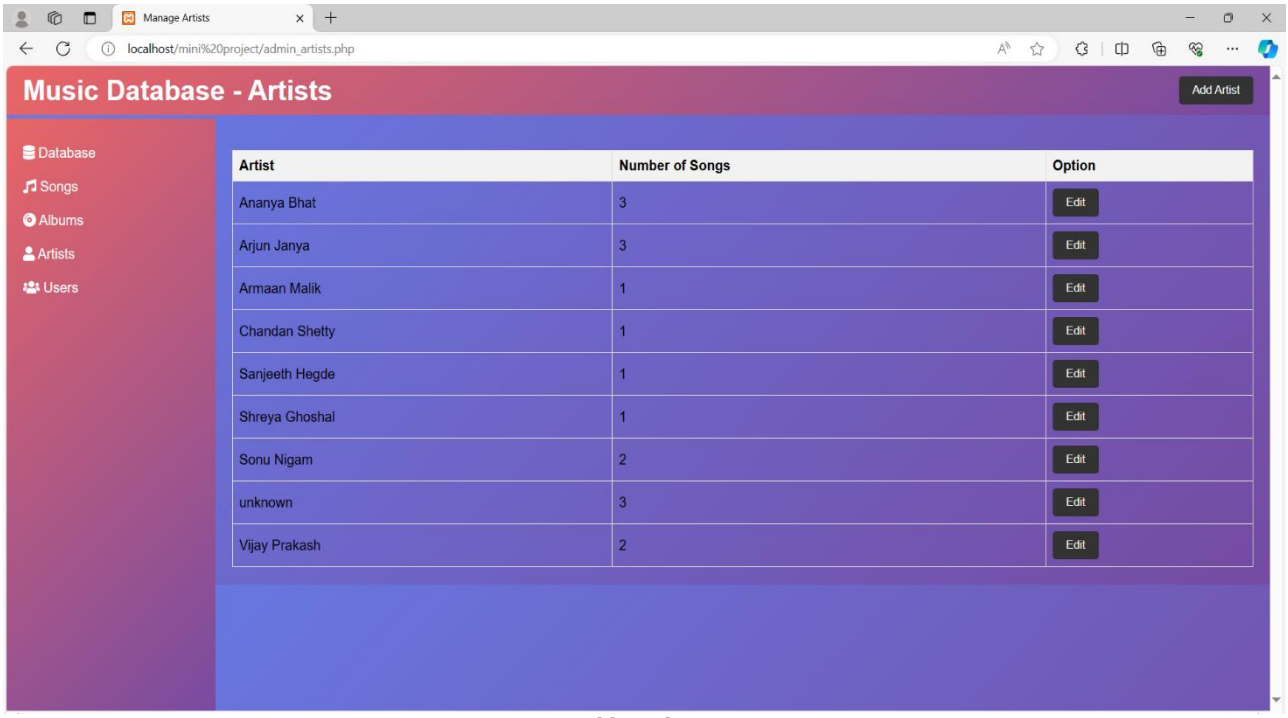

Figure 6.11 Dashboard-Artist page

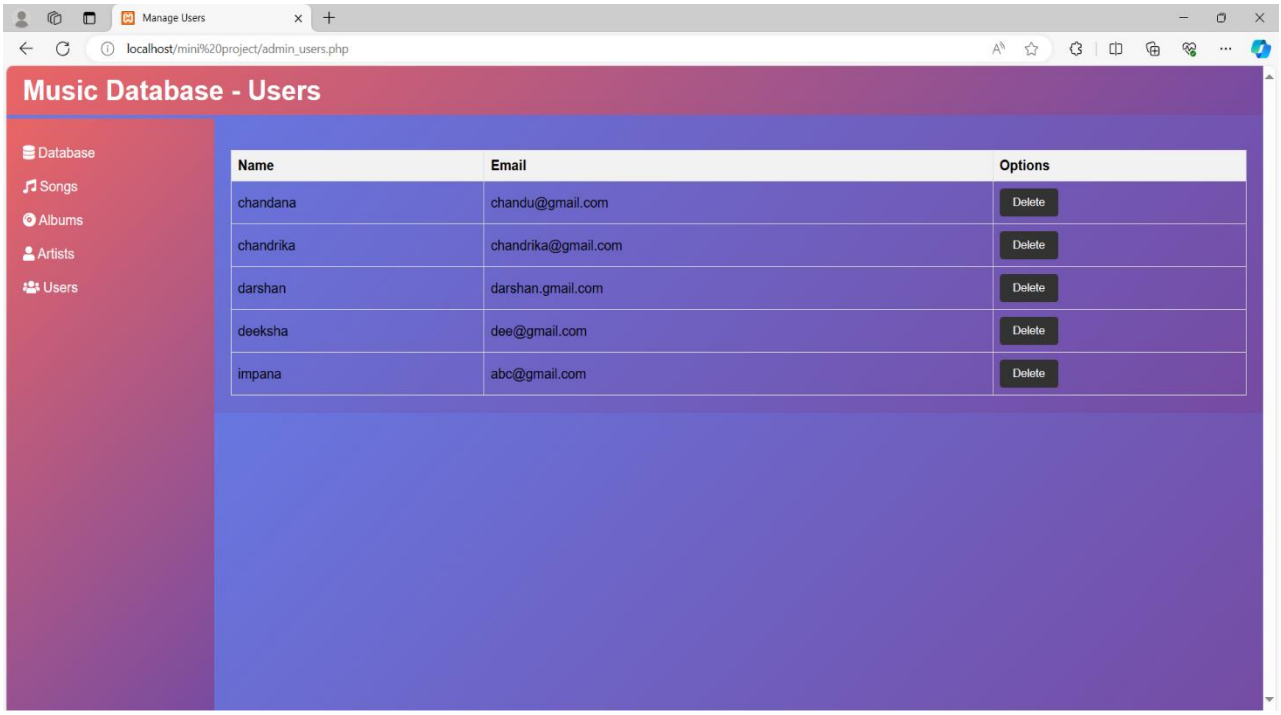

Figure 6.12 Dashboard-User information table

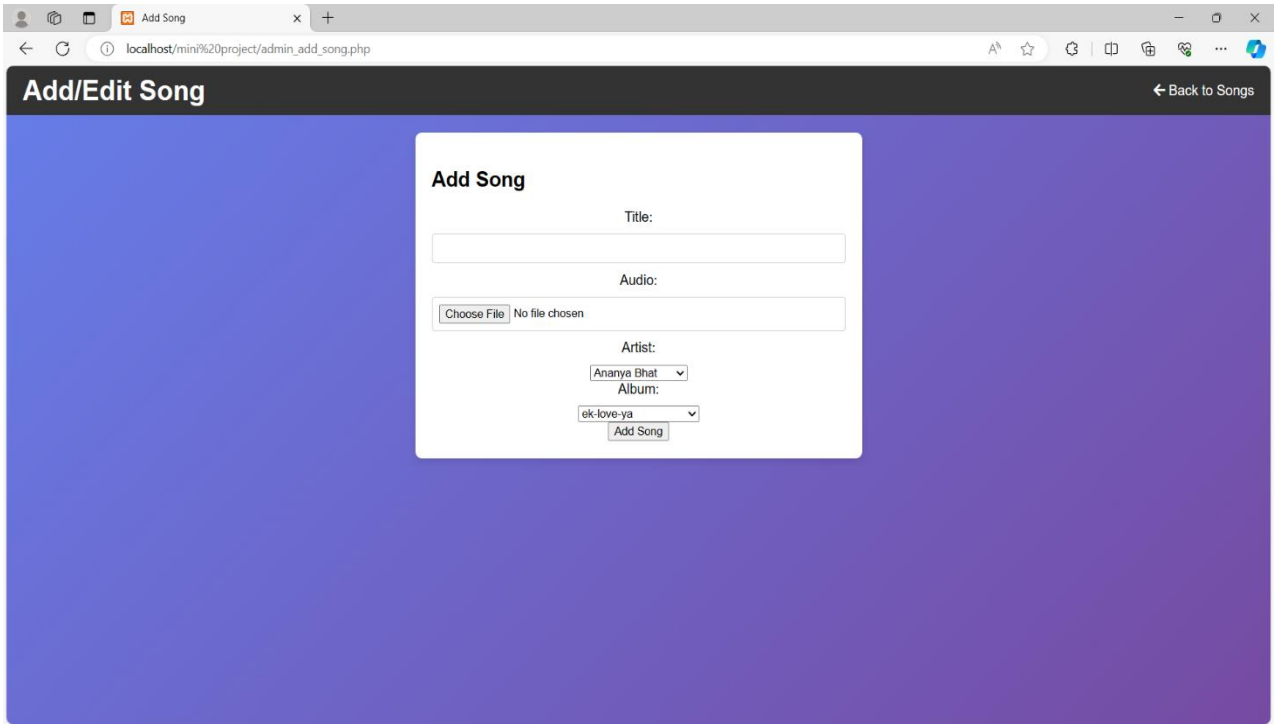

Figure 6.13 Add / edit songs page

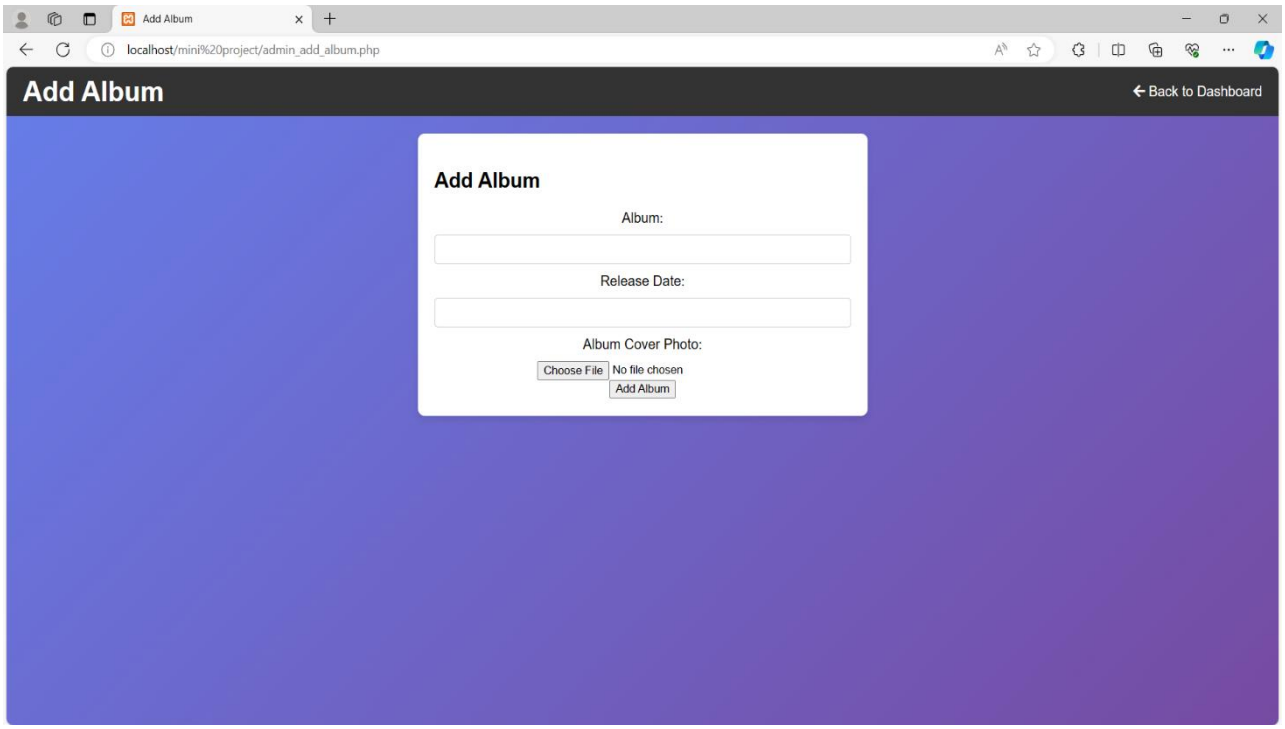

Figure 6.14 Add album page

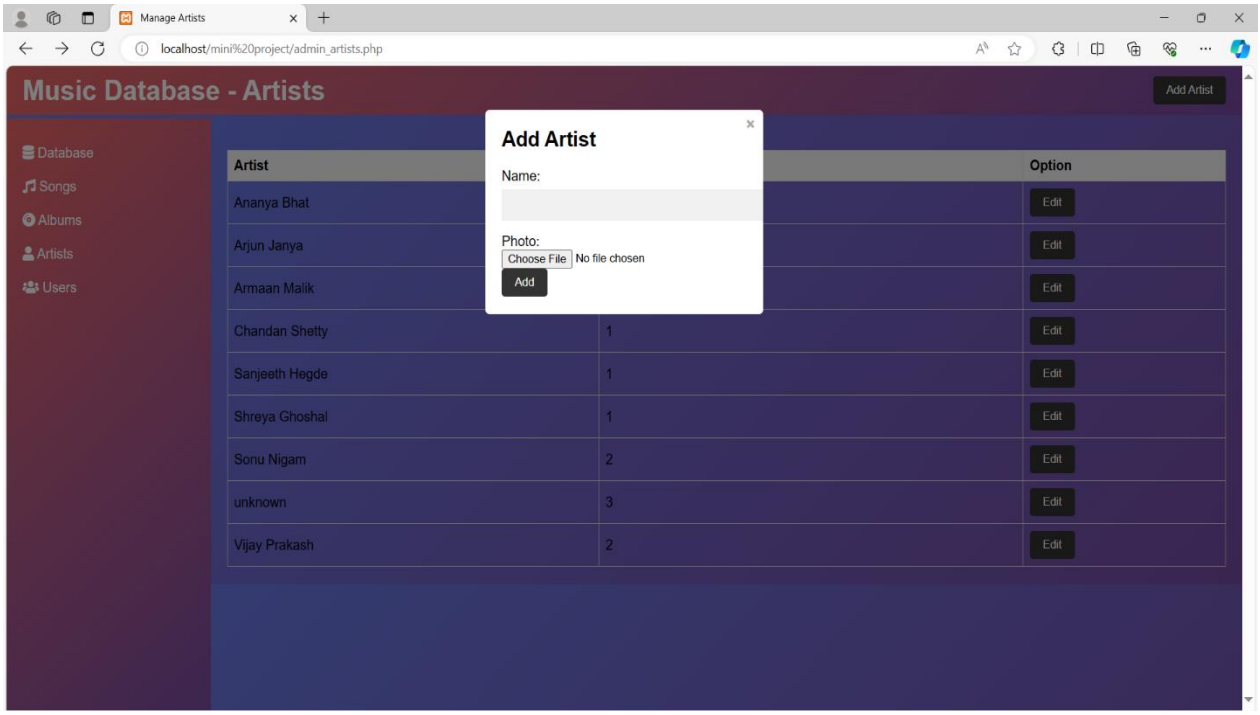

Figure 6.15 Add artist page

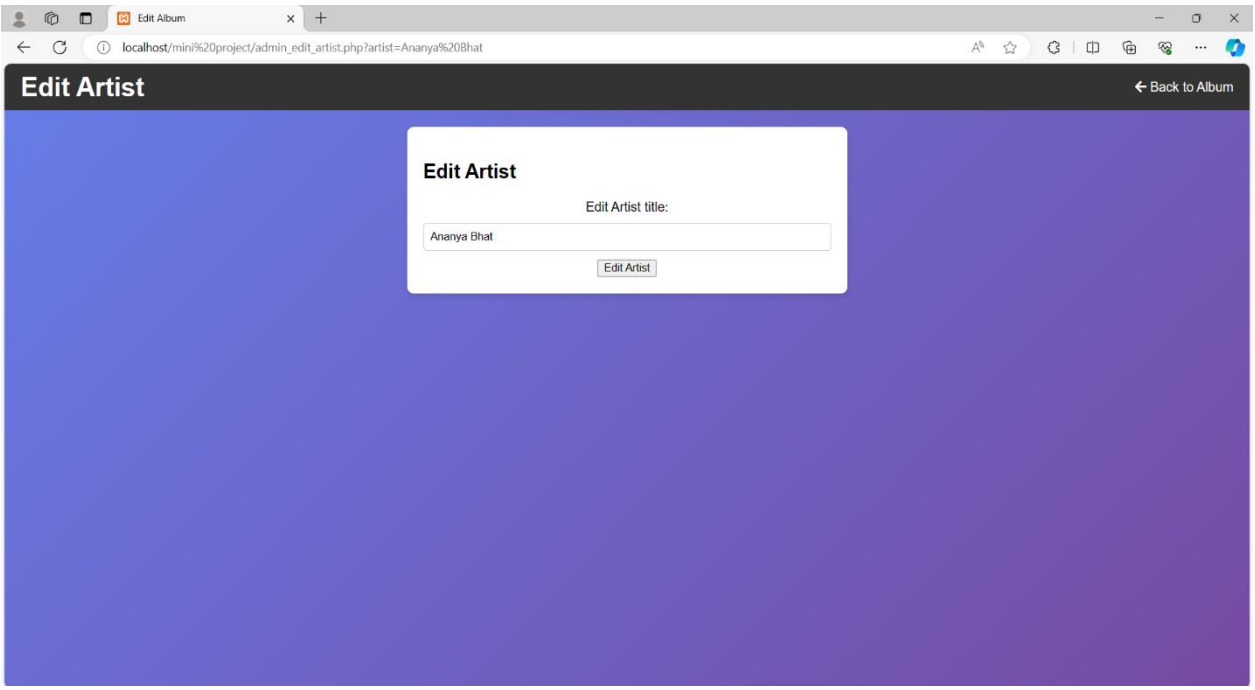

Figure 6.16 Edit Artist page

7. Conclusion & Future Work

We conclude that in this project an efficient music management system was built which allows the admin to manage the music library efficiently ,in a organized way. This project allow admin to add , remove and edit the different attributes like songs ,artists and albums in the music system .By this music management system named Melomaniczz user can experience the organized music player behaviour which allow them to search for the songs of their choice to listen and download. This project related to database management systems, helped us to know how to create and manage a database in SQL using PHP for any of our data . The result of the project is the efficient music player system that can either run without the internet connection also. Melomaniczz has a very outstanding frontend look with proper user friendly nature to attract the users.

While the system we have developed is functional, there are several areas that can be improved in future versions. One potential area of improvement is to include a feature that allows the user to keep track of the song they have listened and downloaded ,and can make users to rate the song they like . The search could be made much more efficient by making user to search based on categories like genre , languages etc .And also the voice recognition for search could help the user .More improvement like the lyrics providing also can be done to make system more efficient. These are some future improvements that can be made to the existing system , still more can be done based on individual creative thinking.

8.References

[1] Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.

[2] Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill .

[3] Abraham Silberschatz, Henry F. Korth and S. Sudarshan's Database System Concepts 6th EditionTata Mcgraw Hill Education Private Limited .

[4] https://www.w3schools.com

[5] https://www.google.com

[6] https://www.smartdraw.com