MUSIC LIBRARY MANAGEMENT

A MINI-PROJECT REPORT

Submitted by

NITHILAN M 230701216

PRANAV RAM S 230701234

PRASANNA KUMAR M 230701237

In partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

NOVEMBER 2023

BONAFIDE CERTIFICATE

Certified that this project “MUSIC LIBRARY MANAGEMENT” is the

bonafide work of “NITHILAN M, PRANAV RAM S,PRASANNA KUMAR M” who carried out the project work under my supervision.

SIGNATURE

Ms. ASWANA LAL

ASSISTANT PROFESSOR

Computer Science and Engineering ,

Rajalakshmi Engineering College

(Autonomous),

Chennai

SIGNATURE

INTERNAL EXAMINER

EXTERNAL EXAMINER

This mini project report is submitted for the viva voce examination to be held on _____

TABLE OF CONTENTS

ABSTRACT

The Music Library Management System developed in Java with MySQL is an efficient and robust software solution for organizing, managing, and accessing music collections digitally.

This system offers an intuitive user interface and comprehensive functionalities to store, retrieve, and categorize music tracks seamlessly.

By integrating Java for the application logic and MySQL for the database, it ensures scalability, reliability, and performance. The project simplifies the management of music libraries for individual users or organizations, enabling efficient search, playback, and updates to the database. Its secure and user-friendly design provides a centralized platform to enhance the overall music management experience.

INTRODUCTION

In today's digital era, managing large collections of music has become a necessity for both personal and professional users. The Music Library Management System, developed using Java and MySQL, addresses this need by providing an efficient, user-friendly, and feature-rich platform for organizing and accessing music collections.

This project is designed to streamline the process of categorizing, retrieving, and updating music tracks, ensuring that users can manage their libraries with ease. By leveraging Java for the application logic and MySQL as the backend database, the system delivers high performance, scalability, and data security.

Whether for personal use, such as organizing playlists, or for professional scenarios, like managing music in studios or stores, this system offers a versatile and intuitive solution. It bridges the gap between traditional manual management and modern digital tools, enhancing productivity and the overall music experience.

The Music Library Management System is a step forward in integrating technology into everyday tasks, making it simpler to organize, explore, and enjoy music libraries effectively.

Key Features of the Music Library System in JAVA:

User-Friendly Interface:
Offers a simple and intuitive interface for easy navigation and usage.

Music Categorization:
Organizes tracks by genre, artist, album, or release year.

Search Functionality:
Enables users to quickly find music based on keywords, titles, or metadata.

Playlist Management:
Allows users to create, modify, and save personalized playlists.

Database Integration:
Utilizes MySQL for efficient data storage and retrieval, ensuring scalability.

Track Upload and Removal:
Facilitates adding new music tracks or removing outdated ones.

Secure Access Control:
Employs authentication features to ensure authorized access.

Dynamic Updates:
Provides options to edit metadata, including track name, artist, or album details.

Backup and Recovery:
Includes features to backup the database to prevent data loss.

Cross-Platform Compatibility:
Can run on multiple operating systems supporting Java runtime.

SCOPE OF THE PROJECT

The Music Library Management System has wide applicability, serving both individual and organizational needs. Its scope includes:

1. For Personal Use:

- Managing personal music collections.
- Organizing and playing tracks seamlessly.

2.For Businesses:

- Supporting libraries in music stores or studios.
- Managing and categorizing extensive music collections.

3.Feature Expansion:

- Adding advanced filters like mood-based music classification.
- Integrating streaming services to sync playlists.

4.Scalability:

- Designed to handle growing databases, making it suitable for long-term use.
-

5.Technological Integration:

- Potential to incorporate advanced technologies like AI-based recommendations or voice search.

This project lays the groundwork for an advanced music management tool that can be extended to support future developments in multimedia organization.

PROGRAM:-

Source code for music Library Management system(Executed in Eclipse IDE):

```
import javax.swing.*;

import javax.swing.table.DefaultTableModel;

import java.awt.*;
```

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class MusicLibrary extends JFrame {
    private JTextField titleField, artistField, albumField, genreField;
    private JTable songTable;
    private DefaultTableModel tableModel;
    private Connection connection;

    public MusicLibrary() {
        connectToDatabase();
        setTitle("Music Library Management System");
        setSize(600, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        JPanel inputPanel = new JPanel(new GridLayout(5, 2));
        inputPanel.add(new JLabel("Title:"));
        titleField = new JTextField();
        inputPanel.add(titleField);
        inputPanel.add(new JLabel("Artist:"));
        artistField = new JTextField();
        inputPanel.add(artistField);
        inputPanel.add(new JLabel("Album:"));
        albumField = new JTextField();
        inputPanel.add(albumField);
        inputPanel.add(new JLabel("Genre:"));
        genreField = new JTextField();
```

```
    inputPanel.add(genreField);
    JButton addButton = new JButton("Add Song");
    JButton deleteButton = new JButton("Delete Song");
    inputPanel.add(addButton);
    inputPanel.add(deleteButton);
    tableModel = new DefaultTableModel(new String[]{"ID", "Title", "Artist", "Album", "Genre"}, 0);
    songTable = new JTable(tableModel);
    loadSongs();
    add(new JScrollPane(songTable), BorderLayout.CENTER);
    add(inputPanel, BorderLayout.SOUTH);
    addButton.addActionListener(new ActionListener() {
      @Override
      public void actionPerformed(ActionEvent e) {
        addSong();
      }
    });

    deleteButton.addActionListener(new ActionListener() {
      @Override
      public void actionPerformed(ActionEvent e) {
        deleteSong();
      }
    });
  }
  private void connectToDatabase() {
    try {
      connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/musiclibrarydb", "root", "2006");
      System.out.println("Database connected successfully.");
```

```
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
private void loadSongs() {
    try {
        Statement statement = connection.createStatement();
        ResultSet rs = statement.executeQuery("SELECT * FROM Songs");

        tableModel.setRowCount(0); // Clear existing rows
        while (rs.next()) {
            int id = rs.getInt("id");
            String title = rs.getString("title");
            String artist = rs.getString("artist");
            String album = rs.getString("album");
            String genre = rs.getString("genre");
            tableModel.addRow(new Object[]{id, title, artist, album, genre});
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
private void addSong() {
    String title = titleField.getText();
    String artist = artistField.getText();
    String album = albumField.getText();
    String genre = genreField.getText();
```

```
    try {
        PreparedStatement ps = connection.prepareStatement("INSERT INTO Songs (title, artist,
album, genre) VALUES (?, ?, ?, ?)");
        ps.setString(1, title);
        ps.setString(2, artist);
        ps.setString(3, album);
        ps.setString(4, genre);
        ps.executeUpdate();

        loadSongs(); // Refresh table
        clearFields();
    } catch (SQLException e) {
        e.printStackTrace();
    }
  }
  private void deleteSong() {
    int selectedRow = songTable.getSelectedRow();
    if (selectedRow == -1) {
        JOptionPane.showMessageDialog(this, "Please select a song to delete.");
        return;
    }
    int songId = (int) tableModel.getValueAt(selectedRow, 0);

    try {
        PreparedStatement ps = connection.prepareStatement("DELETE FROM Songs WHERE id
= ?");
        ps.setInt(1, songId);
        ps.executeUpdate();
```

```
            loadSongs(); // Refresh table
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    private void clearFields() {
        titleField.setText("");
        artistField.setText("");
        albumField.setText("");
        genreField.setText("");
    }
    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            new MusicLibrary().setVisible(true);
        });
    }
}
```

Database Query(Executed in MySql):

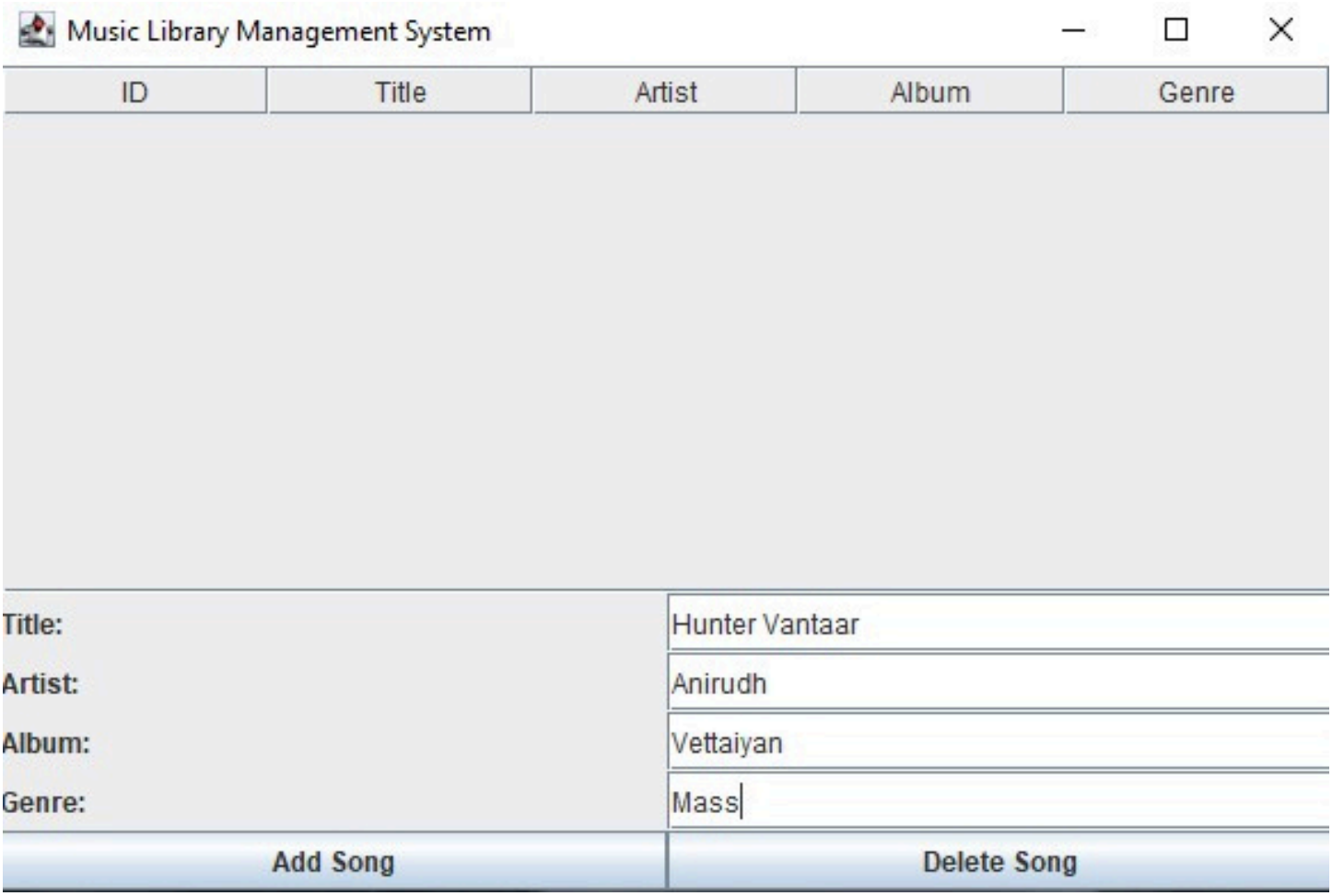
```
CREATE DATABASE musiclibrarydb;

USE musiclibrarydb;

CREATE TABLE Songs (
    id INT PRIMARY KEY AUTO_INCREMENT,
    title VARCHAR(255) NOT NULL,
    artist VARCHAR(255),
```

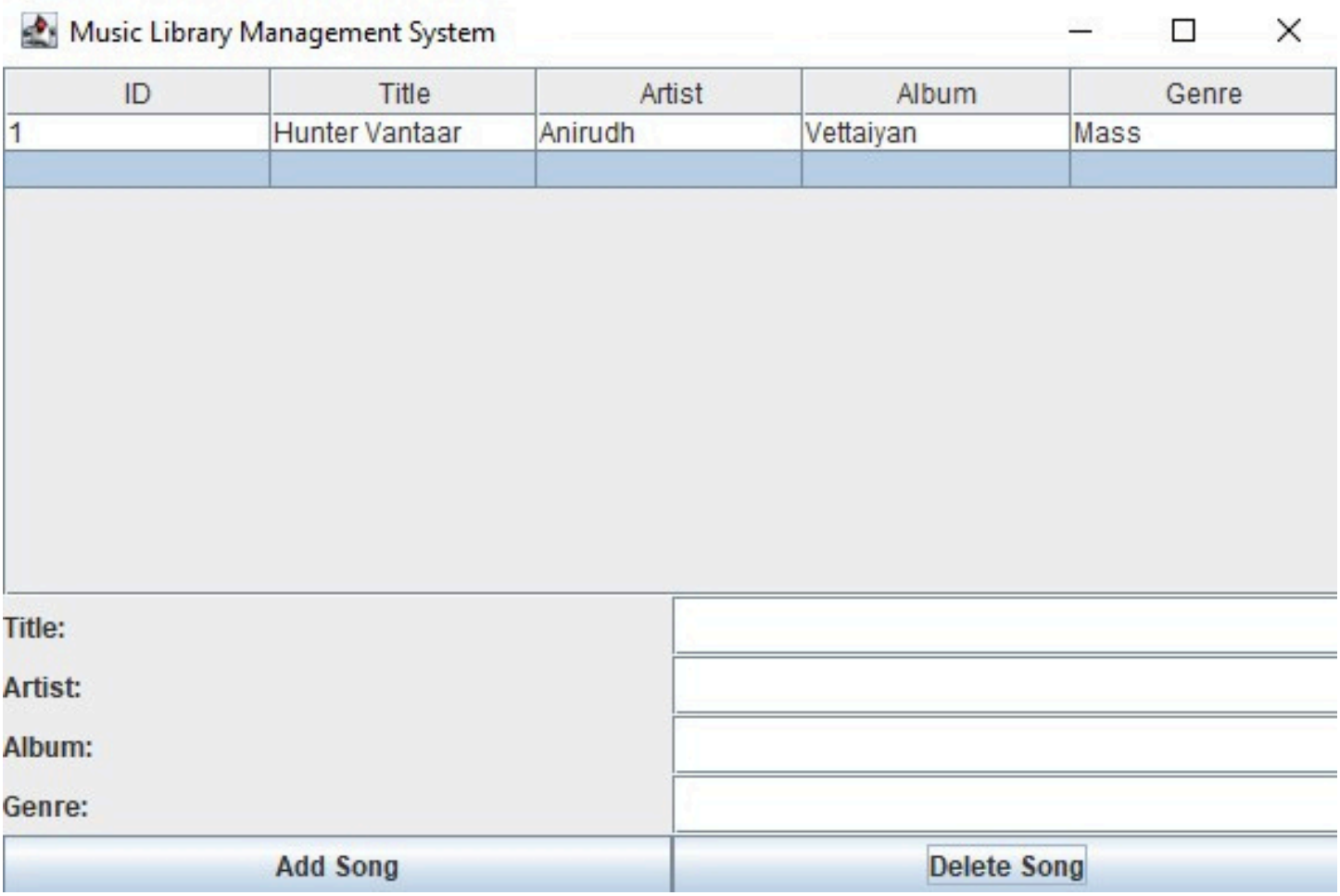
```
    album VARCHAR(255),
    genre VARCHAR(100)
);
```

OUTPUT:

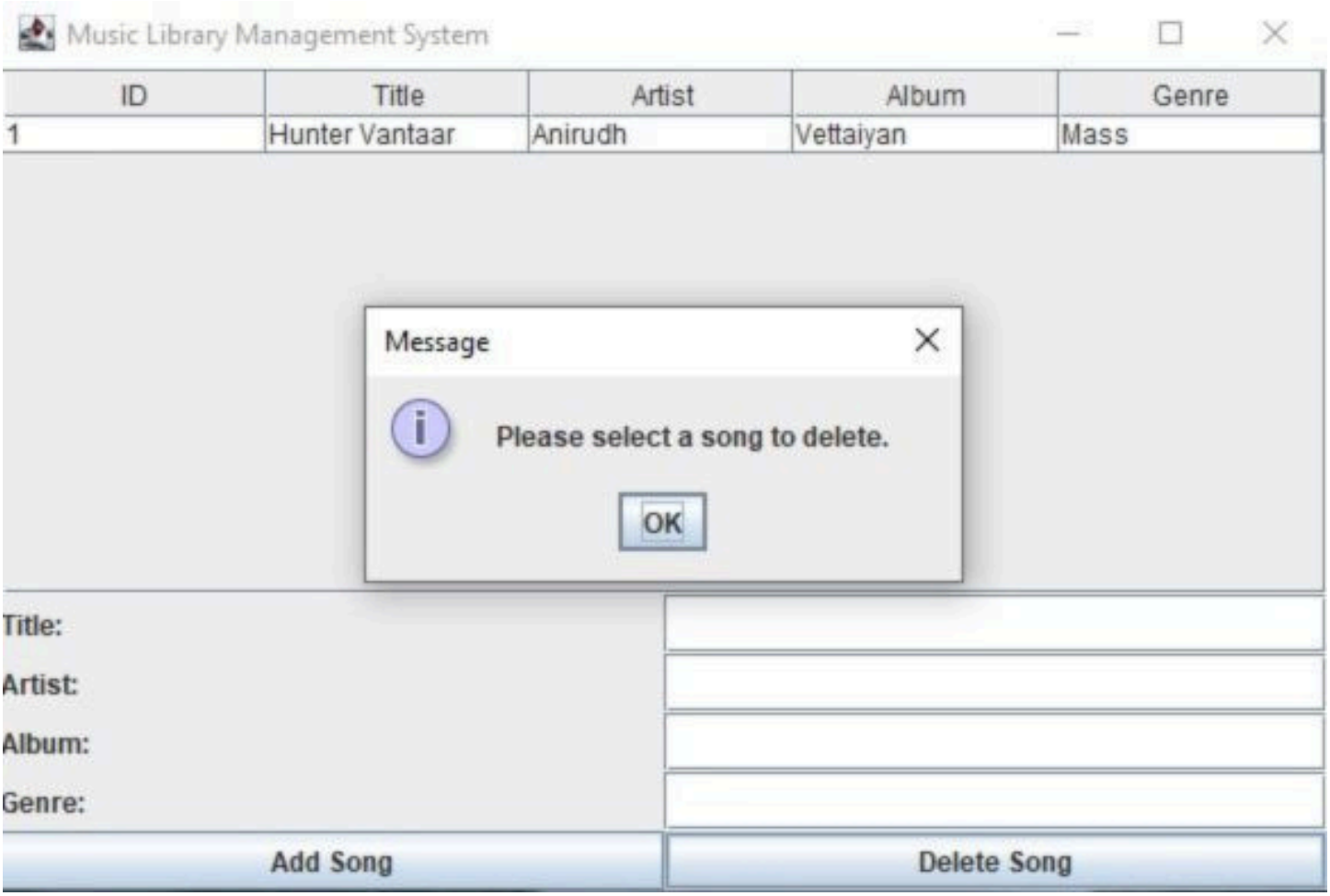
JAVA OUTPUTS:

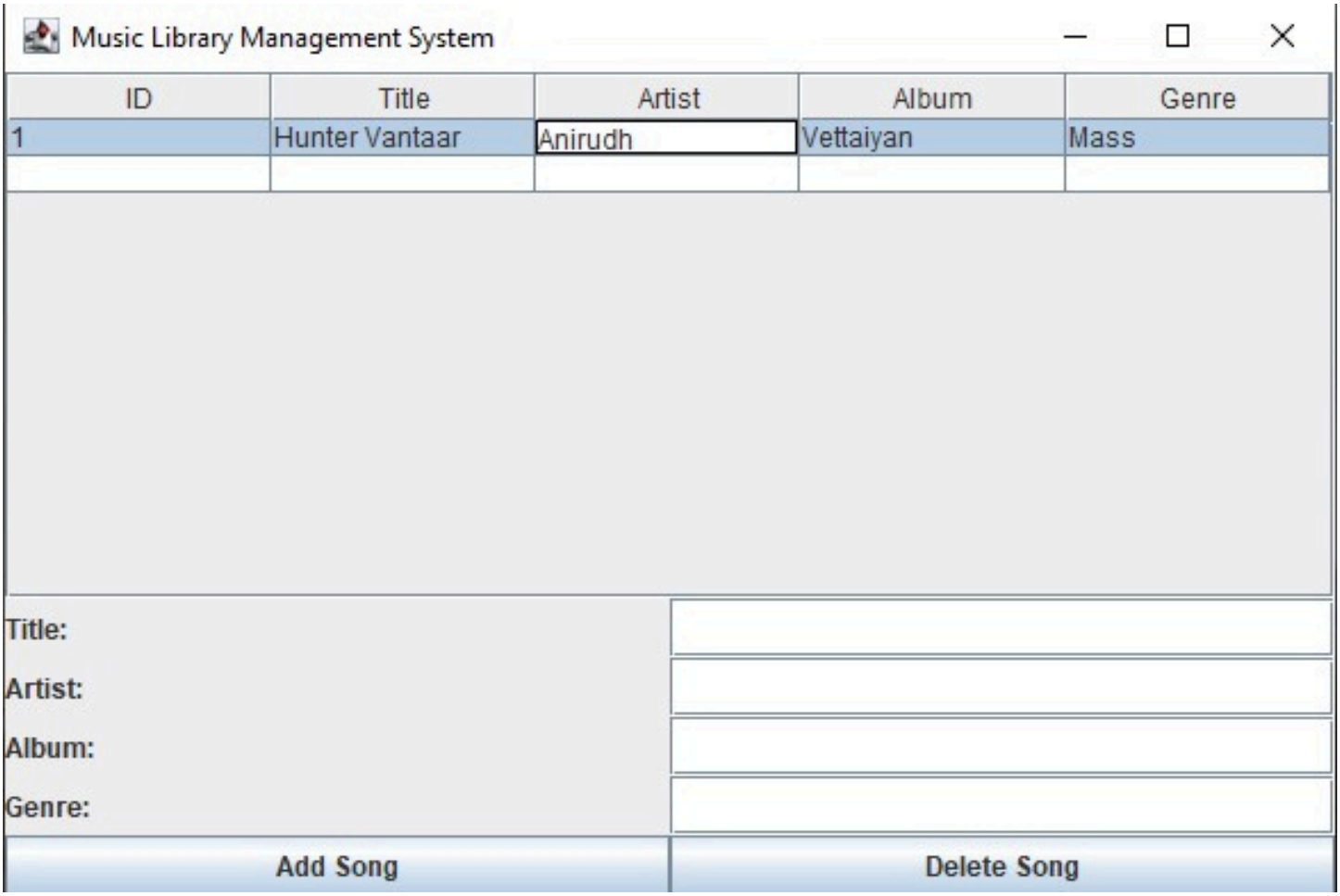
ADDITION OF SONGS :



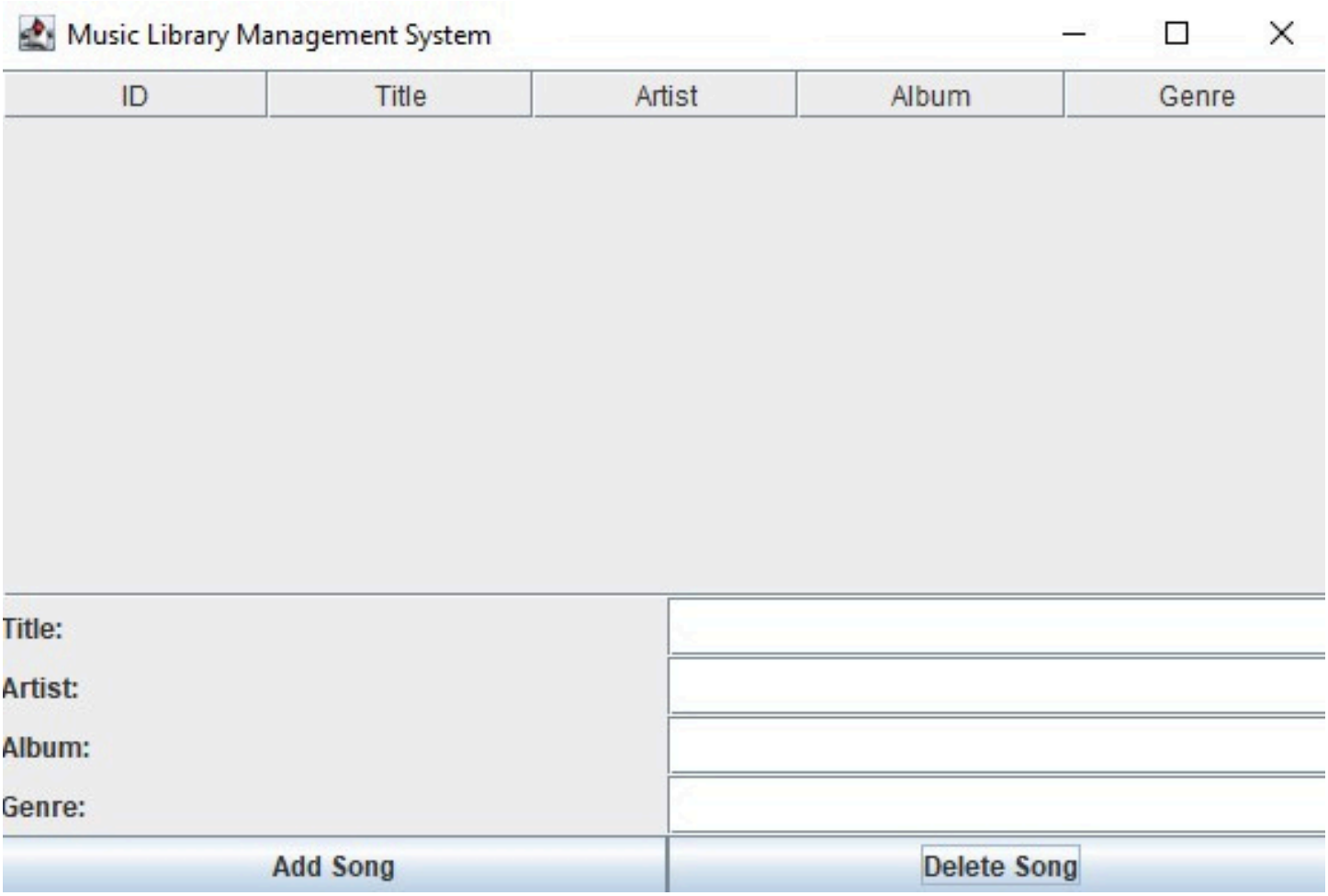


SELECTION OF SONG WHICH IS NEEDED TO BE DELETED :


Music Library Management System
ID
Title
Artist
Album
Genre
1
Hunter Vantaar
Anirudh
Vettaiyan
Mass
Message
Please select a song to delete.
OK
Title:
Artist:
Album:
Genre:
Add Song
Delete Song

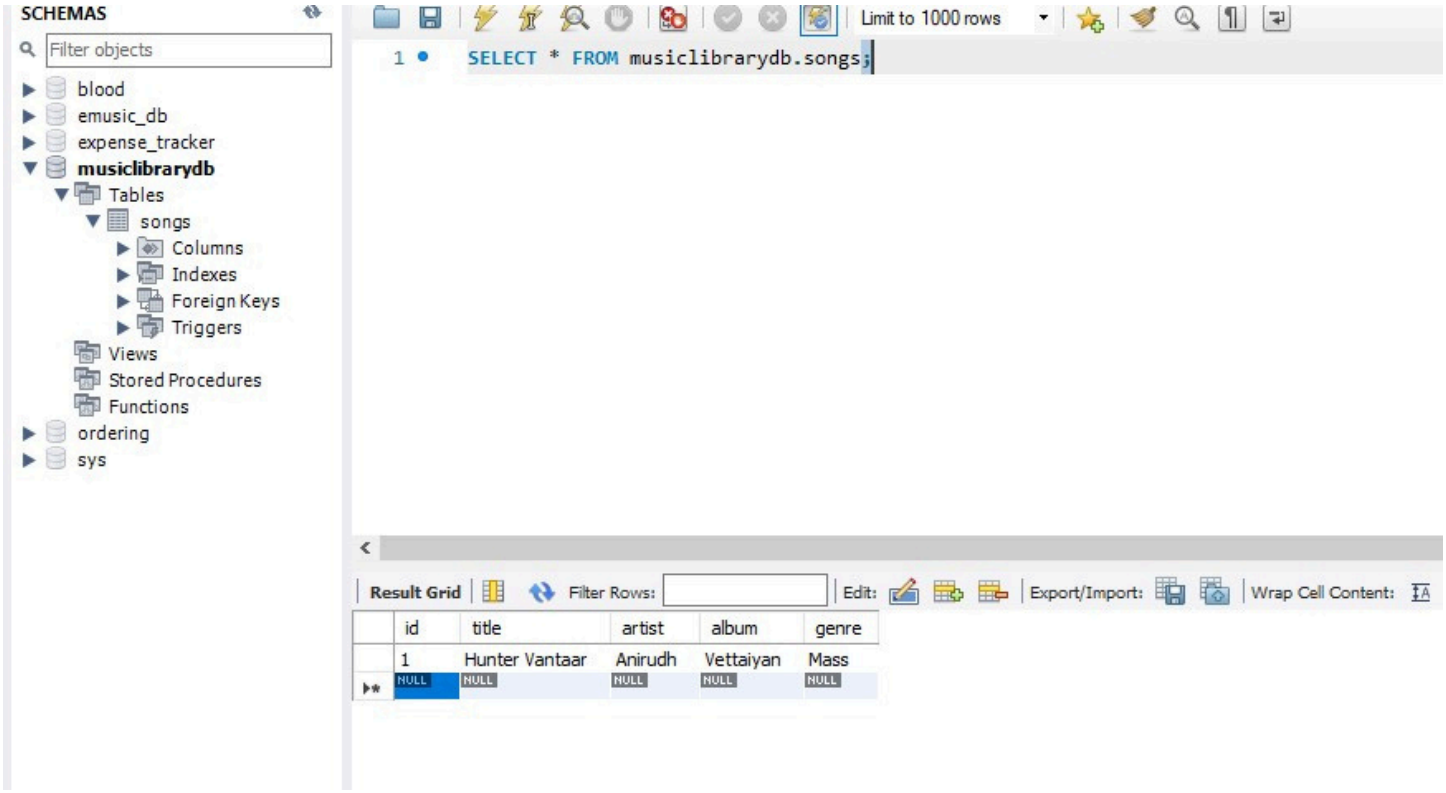


INTERFACE AFTER DELETION OF A SONG :

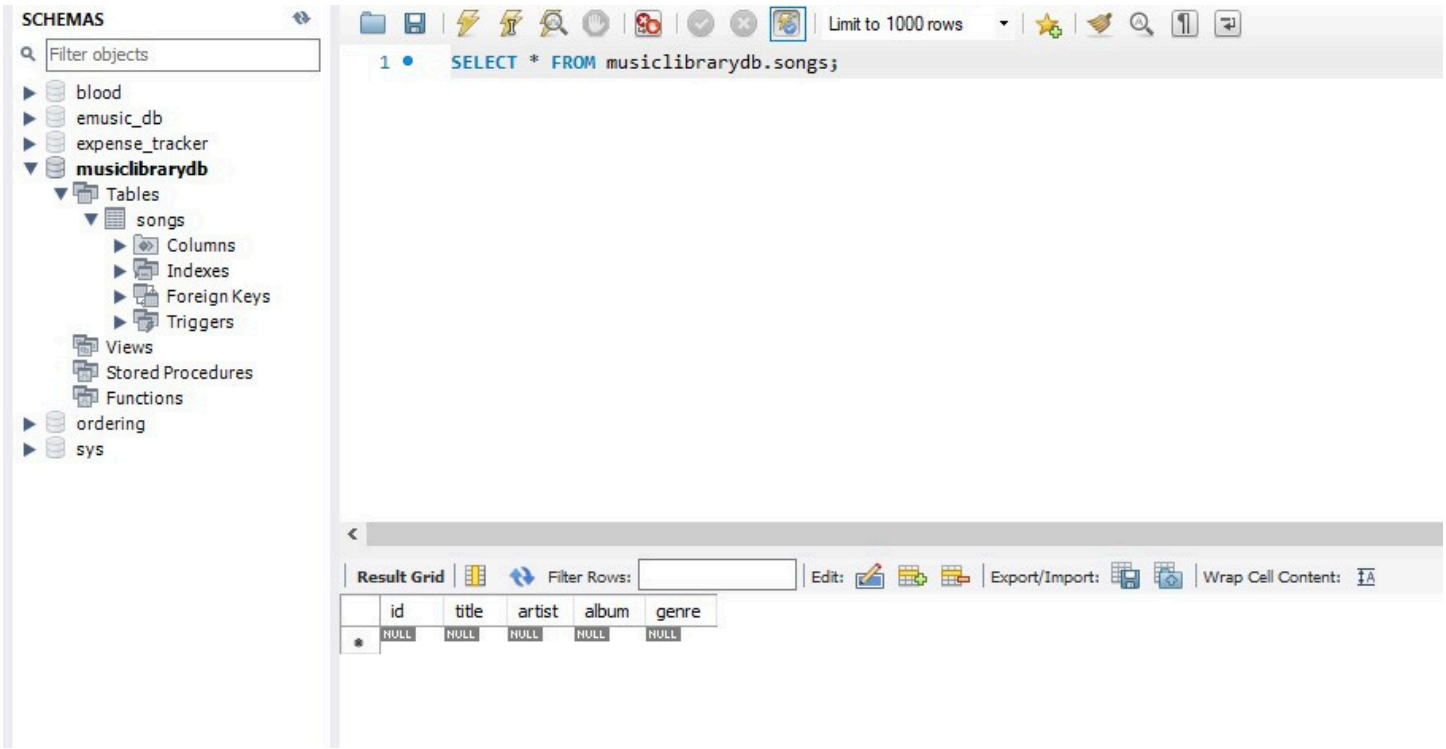


MYSQL OUTPUTS:

Addition of song:



DELETION OF SONG:



CONCLUSION

The Music Library Management System is a comprehensive solution for organizing and managing digital music collections efficiently. By leveraging Java's robust programming capabilities and MySQL's reliable database management, the system ensures a seamless user experience for personal and professional applications.

The system’s intuitive interface, secure access, and customizable features make it an essential tool for users managing large-scale or personal music libraries. It simplifies tasks such as organizing tracks, creating playlists, and updating metadata, significantly enhancing productivity. With its scalable architecture, the project is well-suited for future expansions, allowing it to adapt to evolving user needs and technological advancements.

In conclusion, this project successfully demonstrates how technology can be utilized to modernize and streamline the management of music libraries, making it accessible and efficient for users across various domains.

FUTURE WORK

The Music Library Management System has significant potential for further enhancements and improvements. Future work could focus on the following aspects:

1. Integration with Streaming Platforms:
    - Connect with online streaming services to enable seamless syncing and updates of playlists.
2. AI-Based Recommendations:

    - Implement machine learning algorithms to provide personalized music recommendations based on user preferences and listening history.
3. Mobile Application Development:
    - Extend the system to mobile platforms for easier access and management on the go.
4. Advanced Search and Filters:
    - Incorporate advanced search capabilities like mood-based, tempo-based, or occasion-specific music classification.
5. Voice and Gesture Control:
    - Enable voice commands or gesture-based navigation for enhanced usability.
6. Cloud Storage and Backup:
    - Introduce cloud integration to store and access music libraries remotely with automatic backup and recovery options.
7. Multimedia Support:
    - Expand the system to manage other types of media, such as videos and podcasts, creating a unified media management platform.
8. Collaborative Features:
    - Allow multiple users to collaborate on creating and sharing playlists or managing a collective music database.
9. Enhanced Security:
    - Strengthen security measures by incorporating encryption and advanced authentication mechanisms.
10. Analytics and Insights:
    - Provide usage statistics, trends, and insights to help users understand their music preferences and habits.

By implementing these features, the Music Library Management System can evolve into a more versatile and innovative tool, catering to the diverse and growing needs of its users.

REFERENCE

- www.google.com
- https://chat.openai.com/
- https://chatuml.com/