**SingIT**

**A PROJECT REPORT**

***Submitted by***

CHAUHAN AAYUSH D. ET24MTCA073 SARIYA DHRUV J. ET24MTCA122

Under the Guidance Of

**Prof. KARUNA N. PATEL**

In fulfillment for the **subject *Mobile App Development with Web (MTCA16304)***

Master of Computer Application

At



**Sarvajanik College of Engineering & Technology, Surat**

**November, 2025**

**Sarvajanik Education Society**

#### Sarvajanik College of Engineering & Technology, Surat.

**Department of Computer Application Academic Year 2025-26**



Date:

CERTIFICATE

This is to certify that the project entitled “PROJECT TITLE” has been submitted by AGRAWAL NIMIT A. ET24MTCA069 CHAUHAN AAYUSH D. ET24MTCA073 SARIYA

DHRUV J. ET24MTCA122 towards fulfillment for the subject **Mobile App Development with Web (MTCA16304)** in 3rd Semester of MCA of Sarvajanik University, Surat during the academic year 2025-26.

Prof. KARUNA N. PATEL

**(Guides’ Signature)**

**Dr. Gayatri Kapadia (Head, M.C.A. Department)**

Examiners’ Signature:

1.

**2.**

**Acknowledgements**

We Agrawal Nimit , Chauhan Aayush , Sariya Dhruv have the opportunity to express our knowledge. We would like to express our gratitude to all those who gave us the possibility to complete our project. Success is such a comprehensive project can’t achieve single-handed. so, we would like to express our sincere thanks to all the dignitaries who were involved in making this project the great joy and turning it our into successful piece of work.

We are also indebted to our professor **Prof. Karuna N. Patel** who provided constant encouragement, support & valuable guidance before and during our project. It was her effort who led me to this place for project work. Her guidance and suggestion were invaluable.

We also like to thanks our all the professor who are always ready to give best guide. They are the person who give solution whenever needed.

Thank you.

Agrawal Nimit A. [ET24MTCA069]

Chauhan Aayush D. [ET24MTCA73]

Sariya Dhruv J. [ET24MTCA122]

# INDEX

|  |  |  |
| --- | --- | --- |
| **Sr.No** | **Content** | **Page No.** |
| 1. | Introduction   * 1. Proposed system and its Objectives   2. Core Components   3. Minimum and Maximum Software/Hardware requirements   4. Advantages and Limitations of the Proposed System |  |
| 2. | Requirement Determination & Analysis   * 1. Requirement Determination   2. Targeted Users |  |
| 3. | System Design   * 1. Use Case Diagram   2. Class Diagram   3. Interaction Diagram   4. Activity Diagram   5. Data Dictionary |  |
| 4. | Agile Documentation   * 1. Agile Project Charter   2. Agile Roadmap / Schedule   3. Agile Project Plan   4. Agile User Story ( Minimum 3 Tasks)   5. Agile Release Plan   6. Agile Sprint Backlog   7. Agile Test Plan   8. Earned-value and burn charts |  |
| 5. | Screenshots |  |
| 6. | Proposed Enhancements |  |
| 7. | Conclusion |  |
| 8. | Bibliography |  |

1. **INTRODUCTION**
   1. **PROPOSED SYSTEM AND ITS OBJECTIVES**

The proposed system, **SingIT**, is an Android-based interactive karaoke application designed to enhance the user’s singing experience through real-time audio processing and synchronized lyrics display. The system intelligently separates each song into three components: the original track with lyrics, the user’s vocal recording, and the instrumental (karaoke) version. A dynamic sound- bar mechanism is integrated to allow users to seamlessly switch or blend between these audio modes, creating a customizable listening and singing environment. The system further supports accurate lyric synchronization to guide users while they perform. Additionally, users can record their own vocals, which are then processed and overlaid onto the instrumental track, enabling them to create personalized karaoke mixes directly within the app. This proposed system aims to deliver a smooth, engaging, and highly interactive karaoke platform optimized for mobile devices.

### Objectives :-

* + - To develop an Android application that allows users to enjoy music in three distinct modes – original song with lyrics, user’s vocal recording, and instrumental (karaoke) version.
    - Users can sign up, log in, and manage their accounts.
    - To implement a **sound bar controller** that enables smooth switching or blending between original vocals, user vocals, and instrumental tracks.
    - To allow users to **record their own vocals** and overlay them on instrumental tracks.
    - To create a **user-friendly interface** that makes music interaction enjoyable and intuitive.

### CORE COMPONENTS

#### User:

* Register & Login
* Can view Artist and genre
* View and manage History and Playlist
* View song details
* Users can also record their voice and can equalizer songs
* Can also add Favourites

#### Admin:

* Can manage Genre, Artist and Languages
* Can view users
* Can set today’s special
* Manage Slider

### 1.3. Minimum and Maximum Software/Hardware requirements

* **Hardware Requirement**

|  |  |
| --- | --- |
| RAM | 4 GB or above |
| Processor | Intel core i3 or above |
| Hard Disk | 100 GB |
| Internet Connection | Stable broadband connection |

### Software Requirement

|  |  |
| --- | --- |
| Operating System | Windows 10 / Linux / macOS |
| Web Server | XAMPP(v3.3.0) |
| Front-End Technologies | HTML5, CSS3, Bootstrap5, JavaScript, JSON,AJAX, Flutter |
| Back-End | PHP (8.2.12) |
| Database | MySQL(5.7) |
| Code Editor | VS Code, Android Studio |

**1.4 Advantages and Limitations of the Proposed System**

#### Advantages

* **Dynamic Audio Control** – Users can smoothly switch or blend between the original track, karaoke version, and their own vocals using the sound-bar.
* **Real-Time Lyric Synchronization** – Accurate lyric timing helps users sing along easily and improves performance.
* **Personalized Vocal Recording** – Users can record their voice and overlay it with the instrumental track to create custom karaoke mixes.
* **Enhanced Karaoke Experience** – The system provides an interactive, engaging, and easy-to-use platform for singing and practicing anytime on mobile.

#### Limitations

* + **Audio Quality Issues** – Vocal and instrumental separation may not always be 100% accurate, leading to background noise or distortion.
  + **Performance Constraints** – Real-time processing of songs requires high CPU/GPU power, which may affect performance on low-end Android devices.
  + **Internet Dependency** –Streaming and Storing may need a stable internet connection as it directly been stored on Cloud.
  + **Background Noise in Recording** – User’s recording quality depends on the device microphone and environment.

# REQUIREMENTS DETERMINATION &

**ANALYSIS**

### Requirement Determination

The requirement determination process focuses on understanding what the SingIT karaoke system needs to achieve. Information is collected through user observations, existing karaoke app studies, and understanding user expectations. The core needs identified include real-time lyric synchronization, audio separation, smooth switching between tracks, and the ability for users to record and mix their vocals with the instrumental track. These requirements form the foundation for developing an interactive and user-friendly Android karaoke application.

### Requirement Analysis

The gathered requirements are analysed to define the system's functional and non- functional expectations.

### Functional Requirements

* + Display synchronized lyrics during playback.
  + Separate audio into original vocals, instrumental track, and user vocal recording.
  + Allow switching or blending between audio modes via the sound-bar.
  + Record the user’s voice and mix it with the instrumental.
  + Provide playback options for the mixed recording.

### Non-Functional Requirements

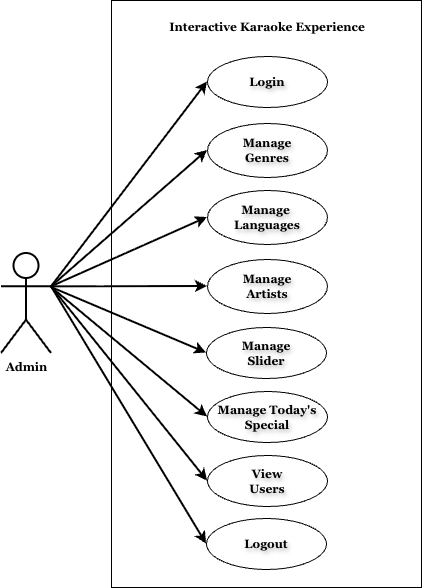
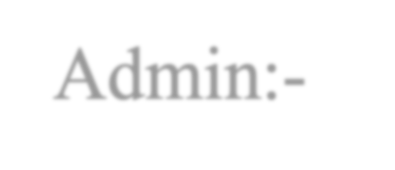
* + **Performance:** Smooth real-time audio processing with minimal delay.
  + **Usability:** Simple and intuitive user interface for all age groups.
  + **Reliability:** Stable playback and recording without crashes.
  + **Security:** Proper handling of user audio files and storage permissions.

### Targeted Users

* **Music Enthusiasts:** Users who enjoy singing and practicing karaoke.
* **Aspiring Singers:** Individuals who want to improve their vocal skills using audio blending and recording features.
* **General Users:** Anyone seeking entertainment through an easy-to-use karaoke app.
* **Content Creators:** Users who want to create custom karaoke mixes for social media or personal use.

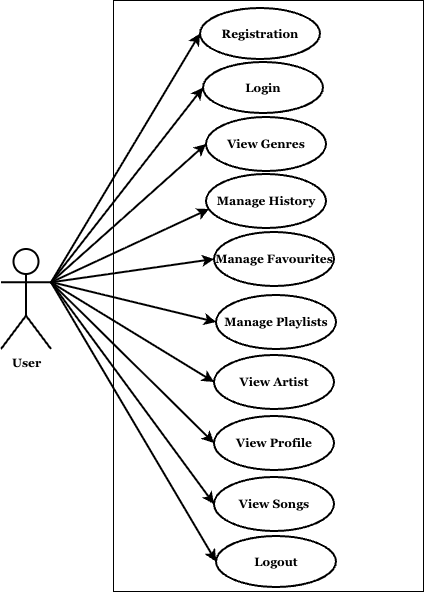
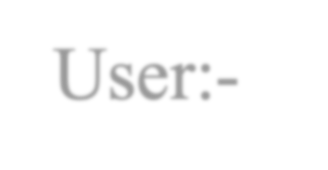
# System Design

* 1. **Use-Case Diagram:**

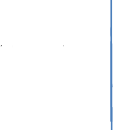
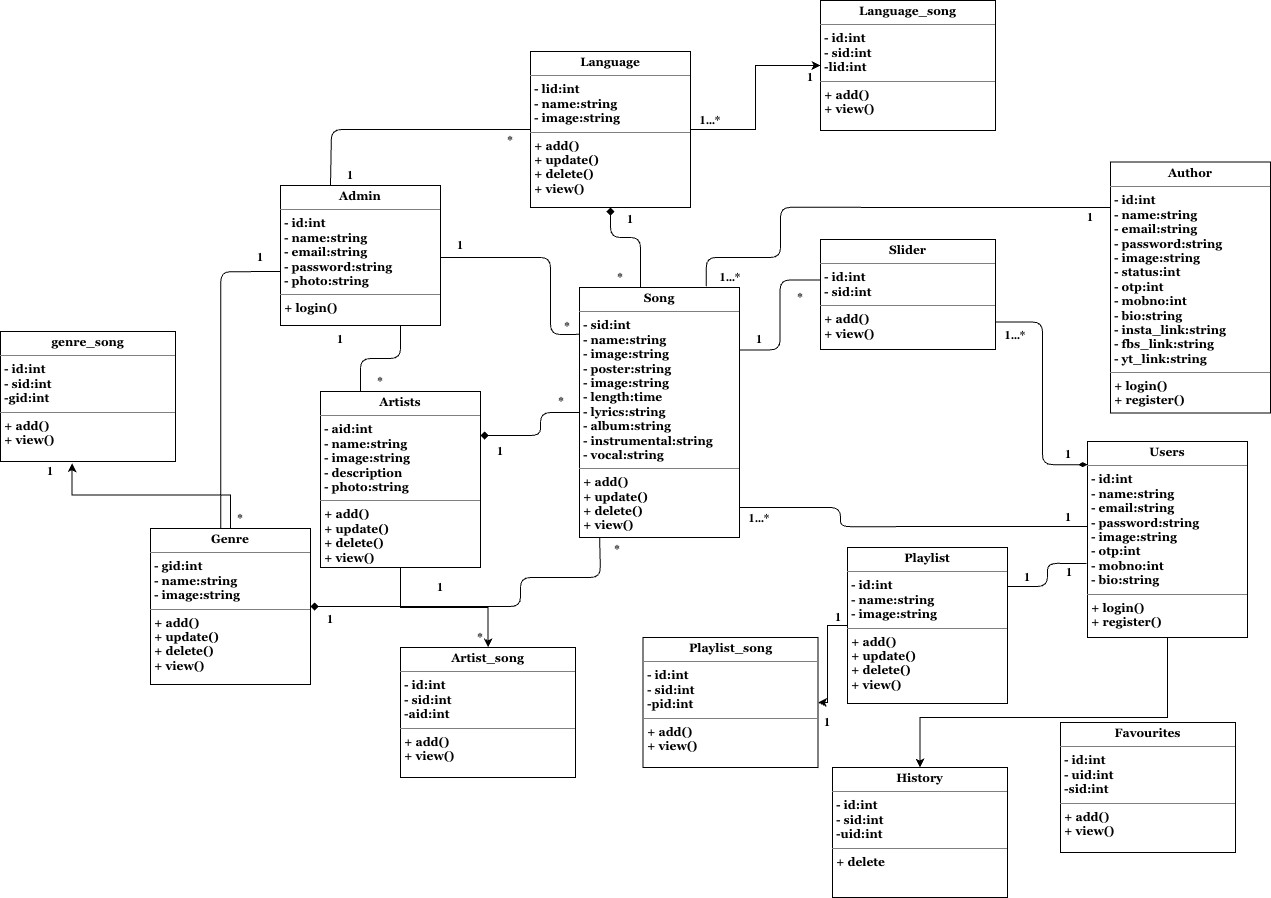
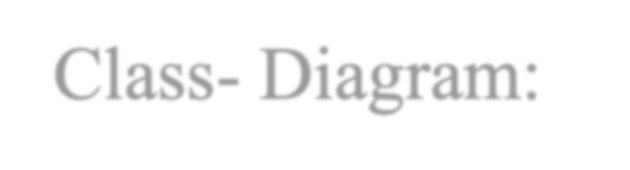
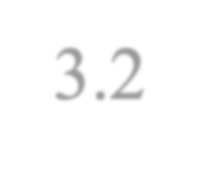


Admin:-

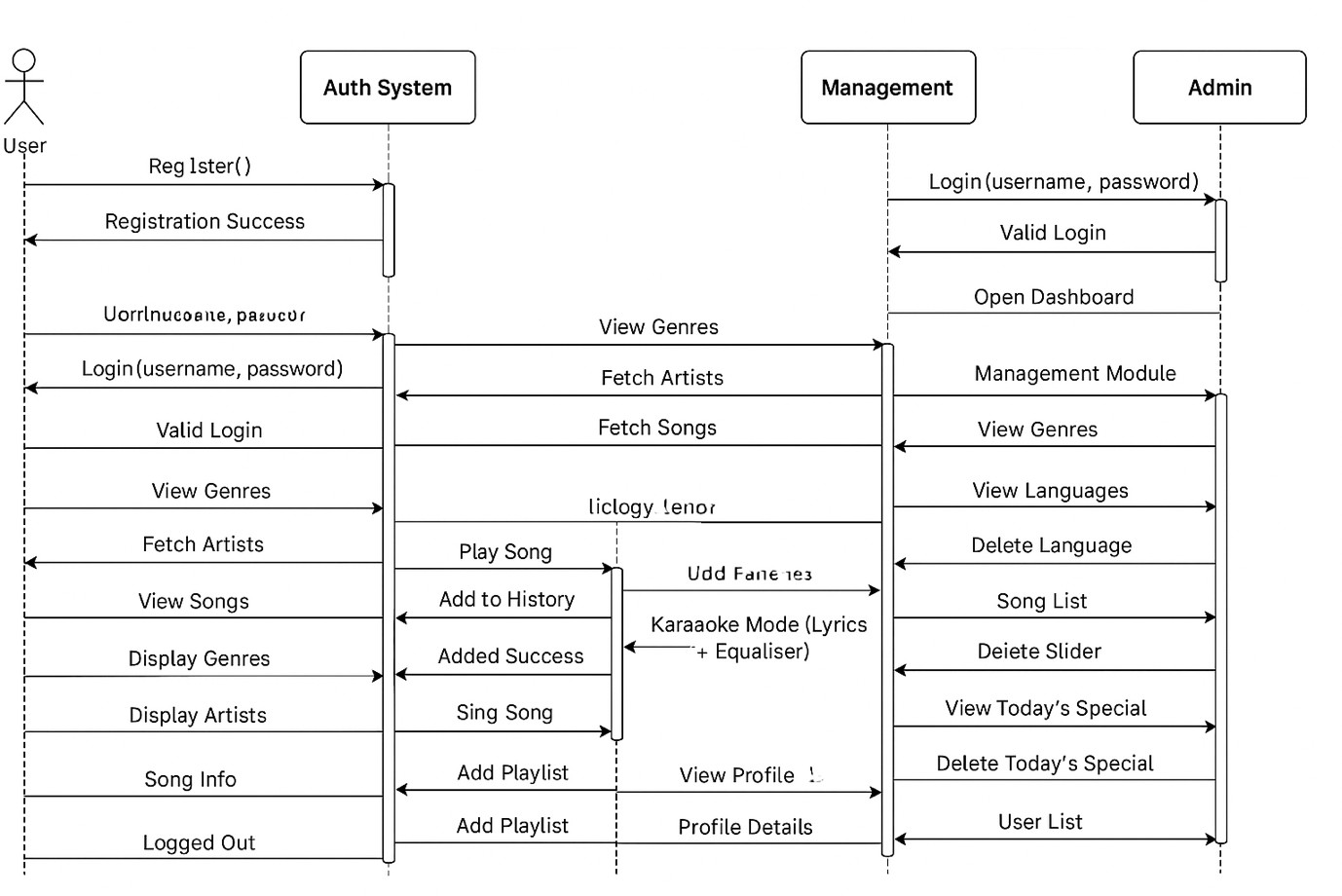
User:-

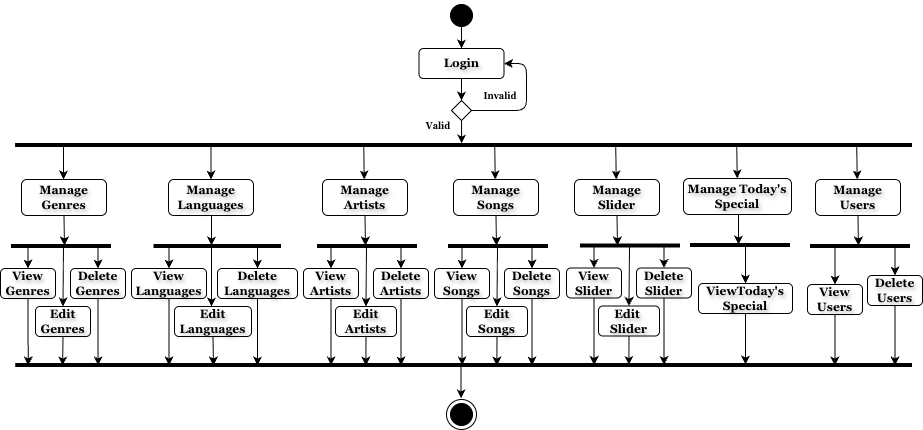


* 1. Class- Diagram:

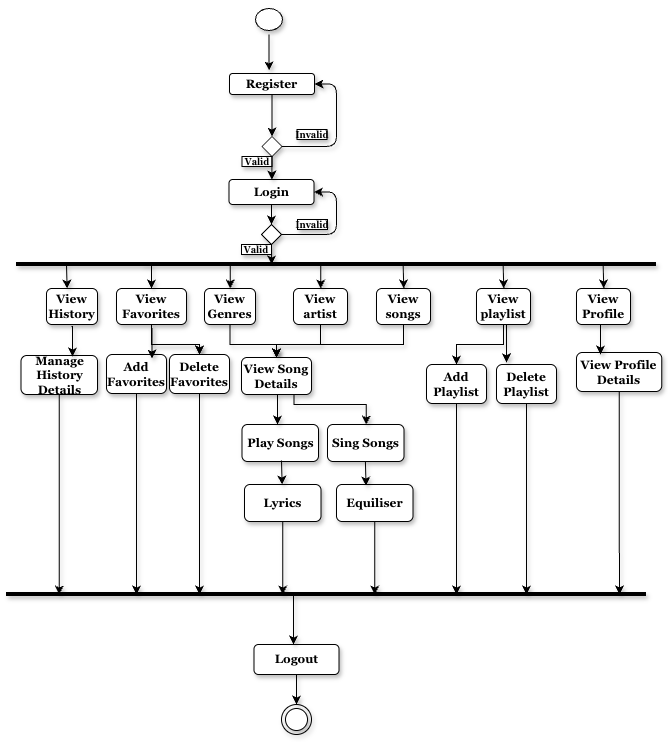


* 1. **Interaction-Diagram:**



* 1. **Activity-Diagram: Admin:**

**User:-**



* 1. **Data-Dictionary: Admin-Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR NO.** | **NAME** | **DATA TYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| 1. | id | int (11) | Primary Key | Admin Id |
| 2. | name | varchar (20) | NOT NULL | Admin Name |
| 3. | email | varchar (50) | NOT NULL | Admin Email Id |
| 4. | password | varchar (255) | NOT NULL | Password |
| 5. | photo | varchar (255) | NOT NULL | Admin Image |

**Artist-Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR NO.** | **NAME** | **DATA TYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| 1. | id | int (11) | Primary Key | Admin Id |
| 2. | name | varchar (20) | NOT NULL | Admin Name |
| 3. | email | varchar (50) | NOT NULL | Admin Email Id |
| 4. | password | varchar (255) | NOT NULL | Password |
| 5. | photo | varchar (255) | NOT NULL | Admin Image |

**User-Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR NO.** | **NAME** | **DATA TYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| 1. | id | int (11) | Primary Key | User Id |
| 2. | name | varchar (20) | NOT NULL | User Name |
| 3. | email | varchar (50) | NOT NULL | User Email Id |
| 4. | password | varchar (255) | NOT NULL | Password |
| 5. | photo | varchar (255) | NOT NULL | User Image |

**Genre-Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR NO.** | **NAME** | **DATA TYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| 1. | gid | int (11) | Primary Key | Genre Id |
| 2. | name | varchar(10) | NOT NULL | Genre Name |
| 3. | image | varchar(255) | NOT NULL | Genre Image |

**Artist-Song Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR NO.** | **NAME** | **DATA TYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| 1. | id | int (11) | Primary Key | Id |
| 2. | song\_id | int (11) | Foreign Key | Song Id |
| 3. | artist\_id | int (11) | Foreign Key | Artist id |

**Genre-Song Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR NO.** | **NAME** | **DATA TYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| 1. | id | int (11) | Primary Key | Id |
| 2. | song\_id | int (11) | Foreign Key | Song Id |
| 3. | genre\_id | int (11) | Foreign Key | Genre id |

**Song-Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR NO.** | **NAME** | **DATA TYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| 1. | sid | int (10) | Primary Key | Song Id |
| 2. | name | varchar(50) | NOT NULL | Name |
| 3. | image | varchar(255) | NOT NULL | Photo |
| 4. | length | time | NOT NULL | Length |
| 5. | lyrics | text | NOT NULL | Lyrics |
| 6. | album | varchar(100) | NOT NULL | Album Name |
| 7. | instrumental | varchar(255) | NOT NULL | Instrumental Music |
| 8. | vocal | varchar(255) | NOT NULL | Vocal Music |

**Favorite-Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR NO.** | **NAME** | **DATA TYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| 1. | id | int (11) | Primary Key | Id |
| 2. | song\_id | int (11) | Foreign Key | Song Id |
| 3. | user\_id | int (11) | Foreign Key | User id |

**History-Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR NO.** | **NAME** | **DATA TYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| 1. | id | int (11) | Primary Key | Id |
| 2. | song\_id | int (11) | Foreign Key | Song Id |
| 3. | User\_id | int (11) | Foreign Key | User id |

**Language-Table:**

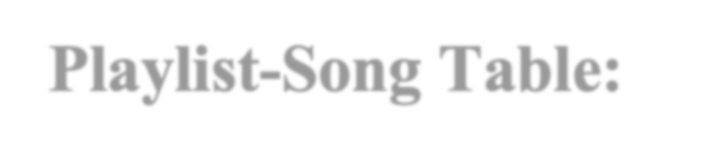
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR NO.** | **NAME** | **DATA TYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| 1. | id | int (11) | Primary Key | Id |
| 2. | name | int (11) | String | Language-Name |
| 3. | image\_id | int (11) | Foreign Key | Image id |

**Language-Song:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR NO.** | **NAME** | **DATA TYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| 1. | id | int (11) | Primary Key | Id |
| 2. | song\_id | int (11) | Foreign Key | Song Id |
| 3. | lid | int (11) | Foreign Key | Language id |

**Playlist:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR NO.** | **NAME** | **DATA TYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| 1. | id | int (11) | Primary Key | Id |
| 2. | name | string | NOT NULL | Song Name |
| 3. | image | string | NOT NULL | Image url |



**Playlist-Song Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR NO.** | **NAME** | **DATA TYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| 1. | id | int (11) | Primary Key | Id |
| 2. | song\_id | int (11) | Foreign Key | Song Id |
| 3. | pid | int (11) | Foreign Key | Playlist id |

**Slider Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR NO.** | **NAME** | **DATA TYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| 1. | id | int (11) | Primary Key | Id |
| 2. | song\_id | int (11) | Foreign Key | Song Id |

# AGILE DOCUMENTATION

1. **1. Agile Project Charter**

|  |  |
| --- | --- |
| **General Project Details** | |
| **Project Name** | SingIT |
| **Project By** | Nimit Aayush Dhruv |
| **Start Date** | 01/ July / 2025 |
| **End Date** | 15 /Nov / 2025 |

* 1. **Agile Roadmap / Schedule**

|  |  |  |
| --- | --- | --- |
| **Phase** | **Duration** | **Deliverables** |
| Requirement Gathering | **01–10 July 2025** | SRS, user stories |
| UI/UX Design | **11–25 July 2025** | Wireframes, layout |
| Database + Backend Setup | **26 July – 10 Aug 2025** | ERD, API setup |
| Core Development – Part 1 | **11 Aug – 10 Sept 2025** | Player, lyrics sync |
| Core Development – Part 2 | **11 Sept – 10 Oct 2025** | Sound-bar, recording |
| System Integration | **11 – 25 Oct 2025** | Combine modules |
| Testing + Fixing | **26 Oct – 10 Nov 2025** | Test reports |
| Final Review & Submission | **11 – 15 Nov 2025** | Final build |

* 1. **Agile Project Plan**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Release** | **Task Name** | **Responsible** | **Start** | **End** | **Days** | **Status** |
| Release 1 – | UI Wireframe Design | Nimit | 05 | 12 | 7 | Completed |
| Core |  |  | Jul | Jul |  |  |
| Foundation |  |  | 2025 | 2025 |  |  |
| Build |  |  |  |  |  |  |
| Release 1 – | Database Schema + | Aayush | 08 | 18 | 10 | Completed |
| Core | ERD |  | Jul | Jul |  |  |
| Foundation |  |  | 2025 | 2025 |  |  |
| Build |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Release 1 – | Song Loading Module | Aayush | 15 | 30 | 15 | Completed |
| Core |  |  | Jul | Jul |  |  |
| Foundation |  |  | 2025 | 2025 |  |  |
| Build |  |  |  |  |  |  |
| Release 1 – | Lyrics Sync Engine | Dhruv | 20 | 05 | 16 | Completed |
| Core |  |  | Jul | Aug |  |  |
| Foundation |  |  | 2025 | 2025 |  |  |
| Build |  |  |  |  |  |  |
| Release 2 – | Sound-Bar UI | Nimit | 10 | 20 | 10 | Completed |
| Audio Engine + | Development |  | Aug | Aug |  |  |
| Mode |  |  | 2025 | 2025 |  |  |
| Switching |  |  |  |  |  |  |
| Release 2 – | Audio Track Switching | Aayush | 12 | 02 | 21 | Completed |
| Audio Engine + | Backend |  | Aug | Sep |  |  |
| Mode |  |  | 2025 | 2025 |  |  |
| Switching |  |  |  |  |  |  |
| Release 2 – | Audio Blending | Dhruv | 15 | 05 | 21 | Completed |
| Audio Engine + | Algorithm |  | Aug | Sep |  |  |
| Mode |  |  | 2025 | 2025 |  |  |
| Switching |  |  |  |  |  |  |
| Release 3 – | Recording UI | Nimit | 10 | 20 | 10 | Completed |
| Vocal |  |  | Sep | Sep |  |  |
| Recording & |  |  | 2025 | 2025 |  |  |
| Mixing |  |  |  |  |  |  |
| Release 3 – | Audio Storage API | Aayush | 12 | 30 | 18 | Completed |
| Vocal |  |  | Sep | Sep |  |  |
| Recording & |  |  | 2025 | 2025 |  |  |
| Mixing |  |  |  |  |  |  |
| Release 3 – | Vocal + Instrumental | Dhruv | 15 | 05 | 20 | Completed |
| Vocal | Mixer |  | Sep | Oct |  |  |
| Recording & |  |  | 2025 | 2025 |  |  |
| Mixing |  |  |  |  |  |  |
| Release 4 – | Playlist Module UI | Nimit | 06 | 15 | 9 | Completed |
| User |  |  | Oct | Oct |  |  |
| Personalization |  |  | 2025 | 2025 |  |  |
| & Library |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Release 4 – | Favorites & History | Aayush | 07 | 22 | 15 | Completed |
| User | Backend |  | Oct | Oct |  |  |
| Personalization |  |  | 2025 | 2025 |  |  |
| & Library |  |  |  |  |  |  |
| Release 4 – | Filters | Dhruv | 10 | 25 | 15 | Completed |
| User | (Artist/Genre/Language) |  | Oct | Oct |  |  |
| Personalization |  |  | 2025 | 2025 |  |  |
| & Library |  |  |  |  |  |  |
| Release 5 – | Final UI Polish | Nimit | 28 | 05 | 8 | Completed |
| Final Release & |  |  | Oct | Nov |  |  |
| Deployment |  |  | 2025 | 2025 |  |  |
| Release 5 – | Backend Optimization | Aayush | 28 | 08 | 11 | Completed |
| Final Release & |  |  | Oct | Nov |  |  |
| Deployment |  |  | 2025 | 2025 |  |  |
| Release 5 – | QA Testing & Bug | Dhruv | 01 | 12 | 11 | Completed |
| Final Release & | Fixing |  | Nov | Nov |  |  |
| Deployment |  |  | 2025 | 2025 |  |  |
| Release 5 – | Deployment Package | All | 12 | 15 | 3 | Completed |
| Final Release & |  | Members | Nov | Nov |  |  |
| Deployment |  |  | 2025 | 2025 |  |  |

* 1. **Agile User Story**

## User Story 1 – UI Display

#### As a user, I didn’t liked the App UI. Tasks:

* + 1. Build lyrics display UI
    2. Implement time-based highlighting
    3. Test scrolling speed & timing

## User Story 2 – Song Commenting

#### As a user, I want to comment on songs so that artist can edit their songs Tasks:

1. Make system that I can rate songs.
2. I want to also make comments on songs

## User Story 3 – Vocal Recording Download

#### As a user, I want to download my own recorded singing with instrumental music. Tasks:

1. Implement AudioRecord API
2. Save and mix recorded audio
3. Add playback of mixed output
4. Improve noise handling

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **4.5 Agile Release Plan** | | | | | | | |
|  | **Release** | **Task Name** | **Responsible** | **Start** | **End** | **Days** | **Status** |
| Release 1 – | UI Wireframe Design | Nimit | 05 | 12 | 7 | Completed |
| Core |  |  | Jul | Jul |  |  |
| Foundation |  |  | 2025 | 2025 |  |  |
| Build |  |  |  |  |  |  |
| Release 1 – | Database Schema + | Aayush | 08 | 18 | 10 | Completed |
| Core | ERD |  | Jul | Jul |  |  |
| Foundation |  |  | 2025 | 2025 |  |  |
| Build |  |  |  |  |  |  |
| Release 1 – | Song Loading Module | Aayush | 15 | 30 | 15 | Completed |
| Core |  |  | Jul | Jul |  |  |
| Foundation |  |  | 2025 | 2025 |  |  |
| Build |  |  |  |  |  |  |
| Release 1 – | Lyrics Sync Engine | Dhruv | 20 | 05 | 16 | Completed |
| Core |  |  | Jul | Aug |  |  |
| Foundation |  |  | 2025 | 2025 |  |  |
| Build |  |  |  |  |  |  |
| Release 2 – | Sound-Bar UI | Nimit | 10 | 20 | 10 | Completed |
| Audio Engine | Development |  | Aug | Aug |  |  |
| + Mode |  |  | 2025 | 2025 |  |  |
| Switching |  |  |  |  |  |  |
| Release 2 – | Audio Track Switching | Aayush | 12 | 02 | 21 | Completed |
| Audio Engine | Backend |  | Aug | Sep |  |  |
| + Mode |  |  | 2025 | 2025 |  |  |
| Switching |  |  |  |  |  |  |
| Release 2 – | Audio Blending | Dhruv | 15 | 05 | 21 | Completed |
| Audio Engine | Algorithm |  | Aug | Sep |  |  |
| + Mode |  |  | 2025 | 2025 |  |  |
| Switching |  |  |  |  |  |  |
| Release 3 – | Recording UI | Nimit | 10 | 20 | 10 | Completed |
| Vocal |  |  | Sep | Sep |  |  |
| Recording & |  |  | 2025 | 2025 |  |  |
| Mixing |  |  |  |  |  |  |
| Release 3 – | Audio Storage API | Aayush | 12 | 30 | 18 | Completed |
| Vocal |  |  | Sep | Sep |  |  |
| Recording & |  |  | 2025 | 2025 |  |  |
| Mixing |  |  |  |  |  |  |
| Release 3 – | Vocal + Instrumental | Dhruv | 15 | 05 | 20 | Completed |
| Vocal | Mixer |  | Sep | Oct |  |  |
| Recording & |  |  | 2025 | 2025 |  |  |
| Mixing |  |  |  |  |  |  |
| Release 4 – | Playlist Module UI | Nimit | 06 | 15 | 9 | Completed |
| User |  |  | Oct | Oct |  |  |
| Personalization |  |  | 2025 | 2025 |  |  |
| & Library |  |  |  |  |  |  |
|  | | | | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | |
|  | Release 4 – | Favorites & History | Aayush | 07 | 22 | 15 | Completed |
| User | Backend |  | Oct | Oct |  |  |
| Personalization |  |  | 2025 | 2025 |  |  |
| & Library |  |  |  |  |  |  |
| Release 4 – | Filters | Dhruv | 10 | 25 | 15 | Completed |
| User | (Artist/Genre/Language) |  | Oct | Oct |  |  |
| Personalization |  |  | 2025 | 2025 |  |  |
| & Library |  |  |  |  |  |  |
| Release 5 – | Final UI Polish | Nimit | 28 | 05 | 8 | Completed |
| Final Release |  |  | Oct | Nov |  |  |
| & Deployment |  |  | 2025 | 2025 |  |  |
| Release 5 – | Backend Optimization | Aayush | 28 | 08 | 11 | Completed |
| Final Release |  |  | Oct | Nov |  |  |
| & Deployment |  |  | 2025 | 2025 |  |  |
| Release 5 – | QA Testing & Bug | Dhruv | 01 | 12 | 11 | Completed |
| Final Release | Fixing |  | Nov | Nov |  |  |
| & Deployment |  |  | 2025 | 2025 |  |  |
| Release 5 – | Deployment Package | All | 12 | 15 | 3 | Completed |
| Final Release |  | Members | Nov | Nov |  |  |
| & Deployment |  |  | 2025 | 2025 |  |  |
|  | | | | | | | |

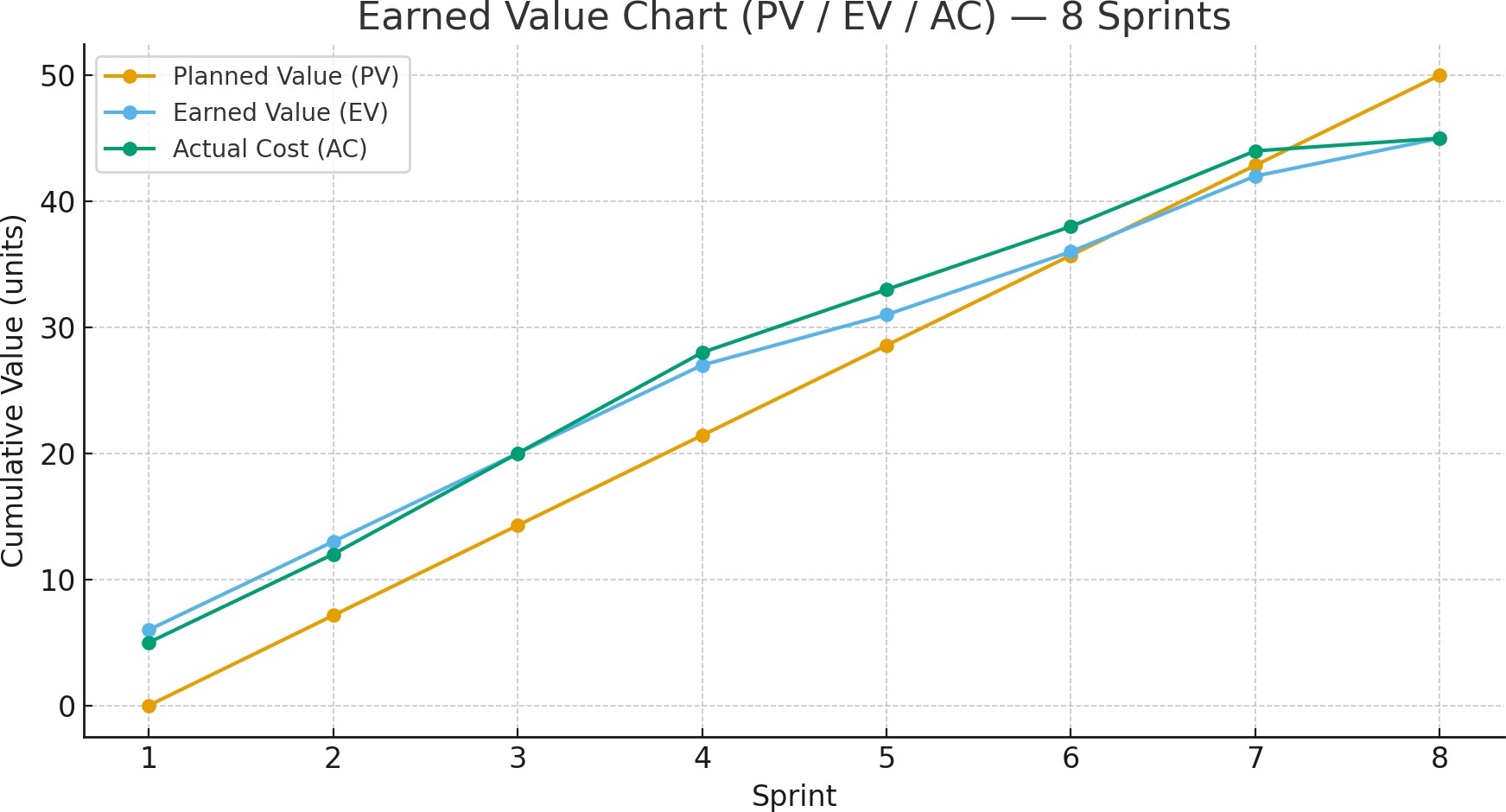
* 1. **Agile Sprint Backlog**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Name** | **Assign To** | **Status** | **Original**  **Estimate Days** | **Actual Result Days** |
| UI Wireframe Design | Nimit | Completed | 7 | **6 (early)** |
| Sound-Bar UI  Development | Nimit | Completed | 10 | **11 (late)** |
| Recording UI | Nimit | Completed | 10 | **9 (early)** |
| Playlist Module UI | Nimit | Completed | 9 | **10 (late)** |
| Final UI Polish | Nimit | Completed | 8 | **8 (on time)** |
| Database Schema + ERD | Aayush | Completed | 10 | **9 (early)** |
| Song Loading Module | Aayush | Completed | 15 | **16 (late)** |
| Audio Track Switching  Backend | Aayush | Completed | 21 | **20 (early)** |
| Audio Storage API | Aayush | Completed | 18 | **19 (late)** |
| Backend Optimization | Aayush | Completed | 11 | **10 (early)** |
| Lyrics Sync Engine | Dhruv | Completed | 16 | **17 (late)** |
| Audio Blending Algorithm | Dhruv | Completed | 21 | **20 (early)** |
| Vocal + Instrumental  Mixer | Dhruv | Completed | 20 | **22 (late)** |
| Filters  (Artist/Genre/Language) | Dhruv | Completed | 15 | **14 (early)** |
| QA Testing & Bug Fixing | Dhruv | Completed | 11 | **12 (late)** |
| Deployment Package | All Members | Completed | 3 | **3 (on time)** |

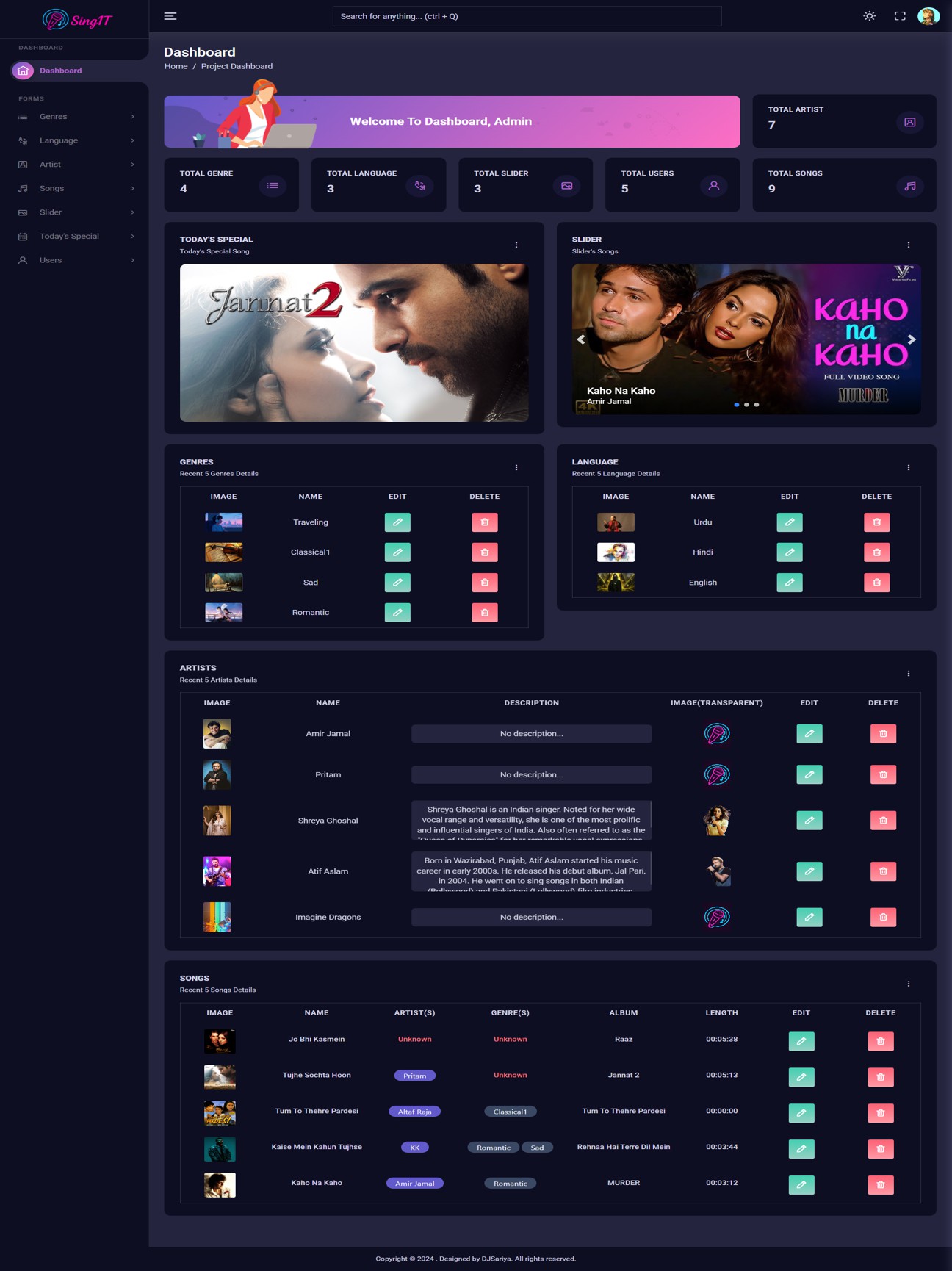
* 1. **Agile Test Plan**

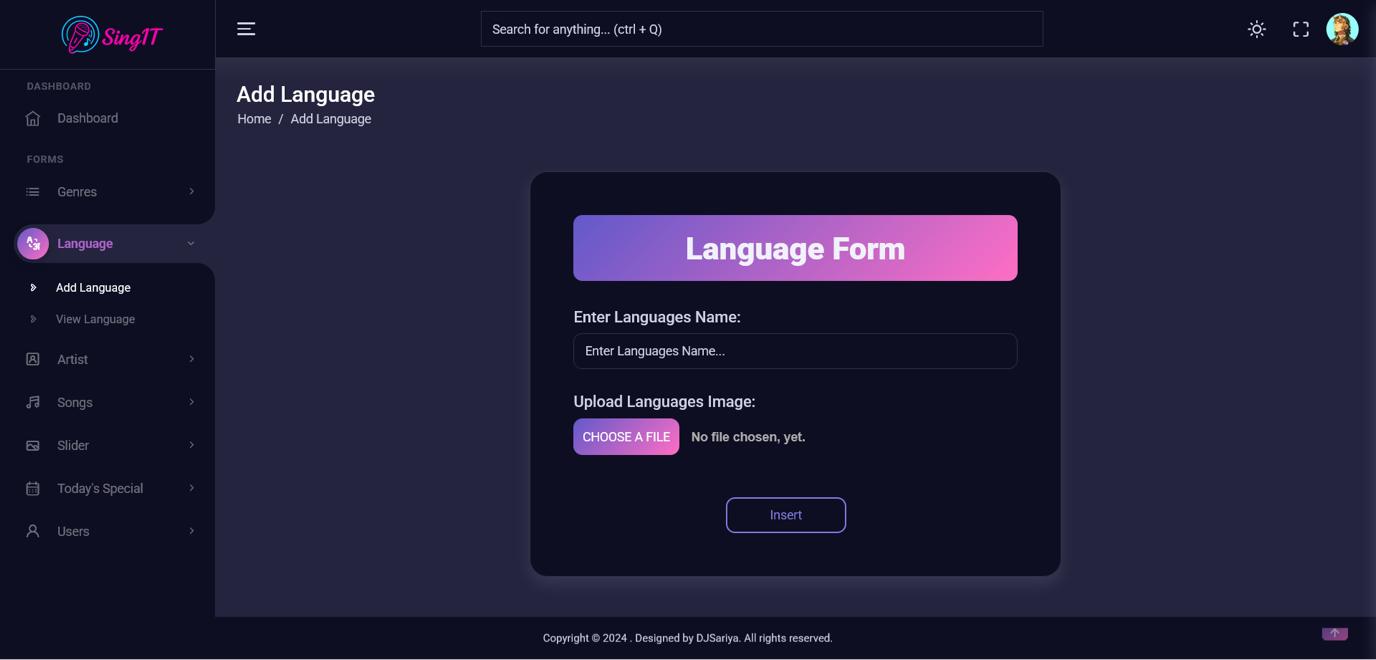
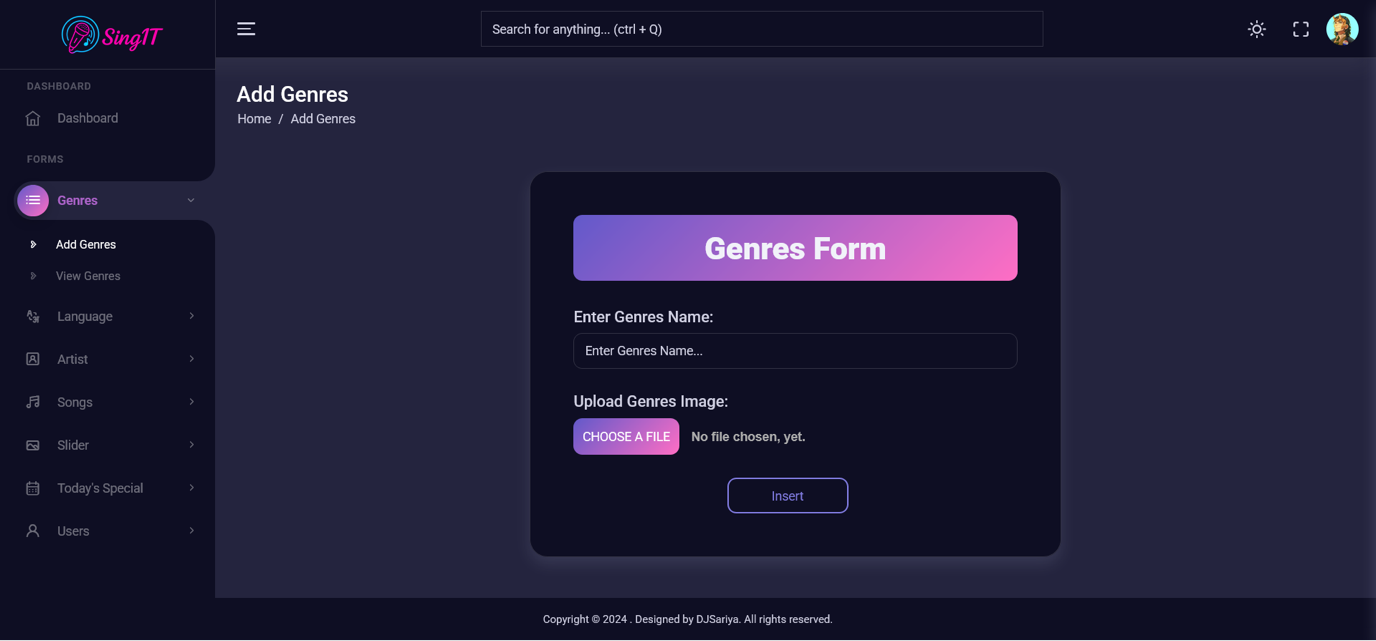
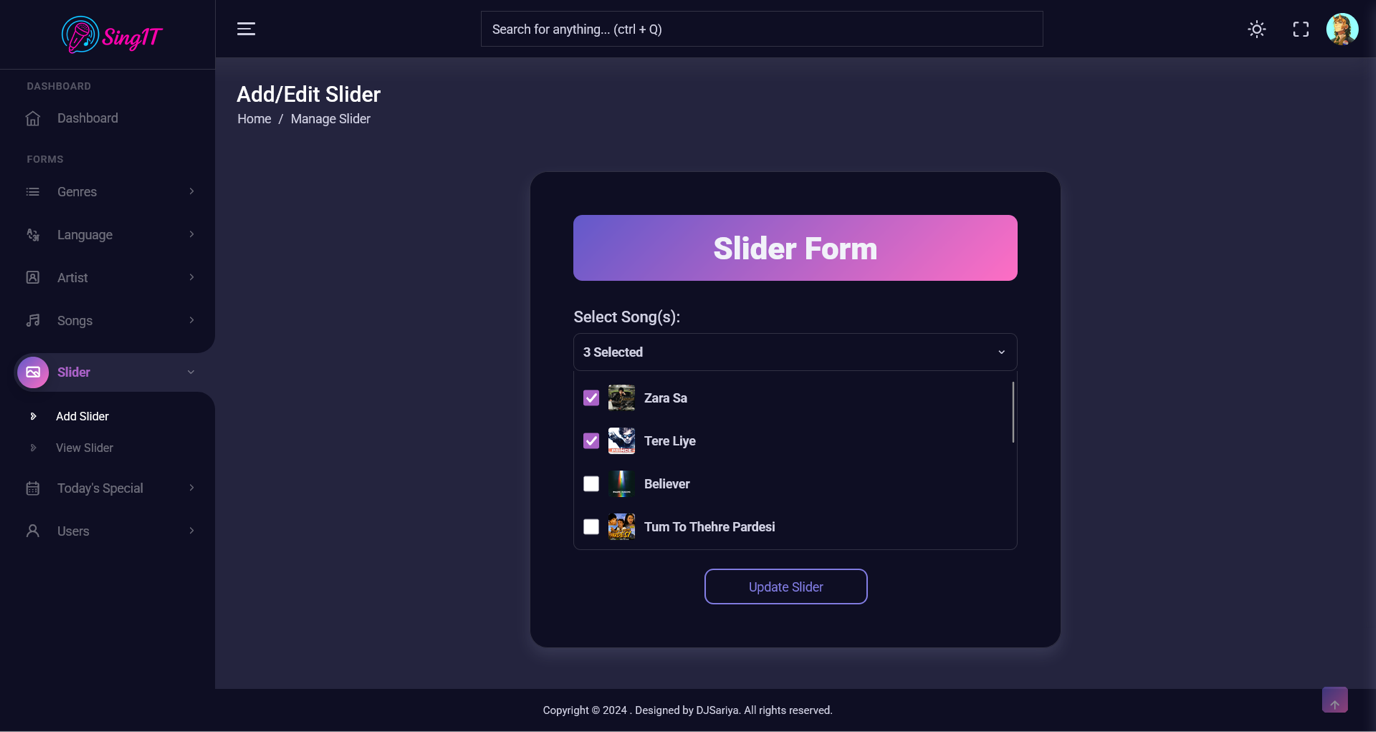
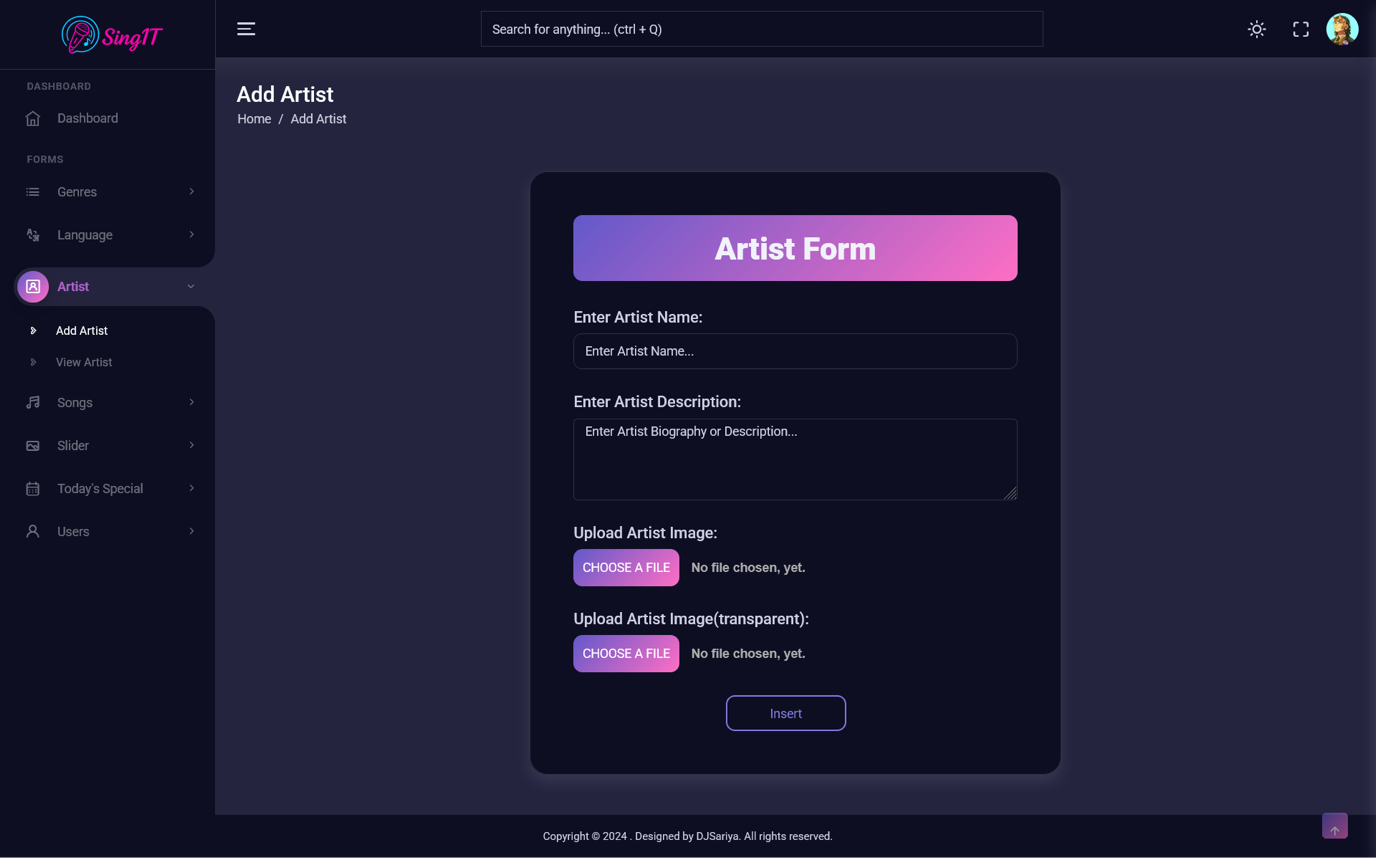
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test**  **ID** | **Action** | **Expected Result** | **Actual**  **Result** | **Status** |
| T001 | Open the application home  screen | Home screen loads within  2 seconds | As  expected | Passed |
| T002 | Login with valid credentials | User successfully logs in  and dashboard appears | As  expected | Passed |
| T003 | Login with invalid password | System shows “Invalid  Credentials” message | As  expected | Passed |
| T004 | Load a song in the player | Song loads and starts  playing | As  expected | Passed |
| T005 | Test lyrics sync feature | Lyrics sync accurately  with music timing | As  expected | Passed |
| T006 | Switch between two audio tracks | Audio switches smoothly within 1 second | As expected | Passed |
| T007 | Record audio via microphone | Recording starts and audio is captured clearly | As expected | Passed |
| T008 | Save recorded audio | Audio file stores successfully in user  library | As expected | Passed |
| T009 | Create a playlist | Playlist created and  visible in library | As  expected | Passed |
| T010 | Add song to favorites | Song appears under  “Favorites” list | As  expected | Passed |
| T011 | Search filter  (Artist/Genre/Language) | Accurate filtered results  displayed | As  expected | Passed |
| T012 | Run full app stability test (30  mins usage) | No crash, smooth  navigation | As  expected | Passed |
| T013 | App logout | User logs out and returns to login screen | As expected | Passed |

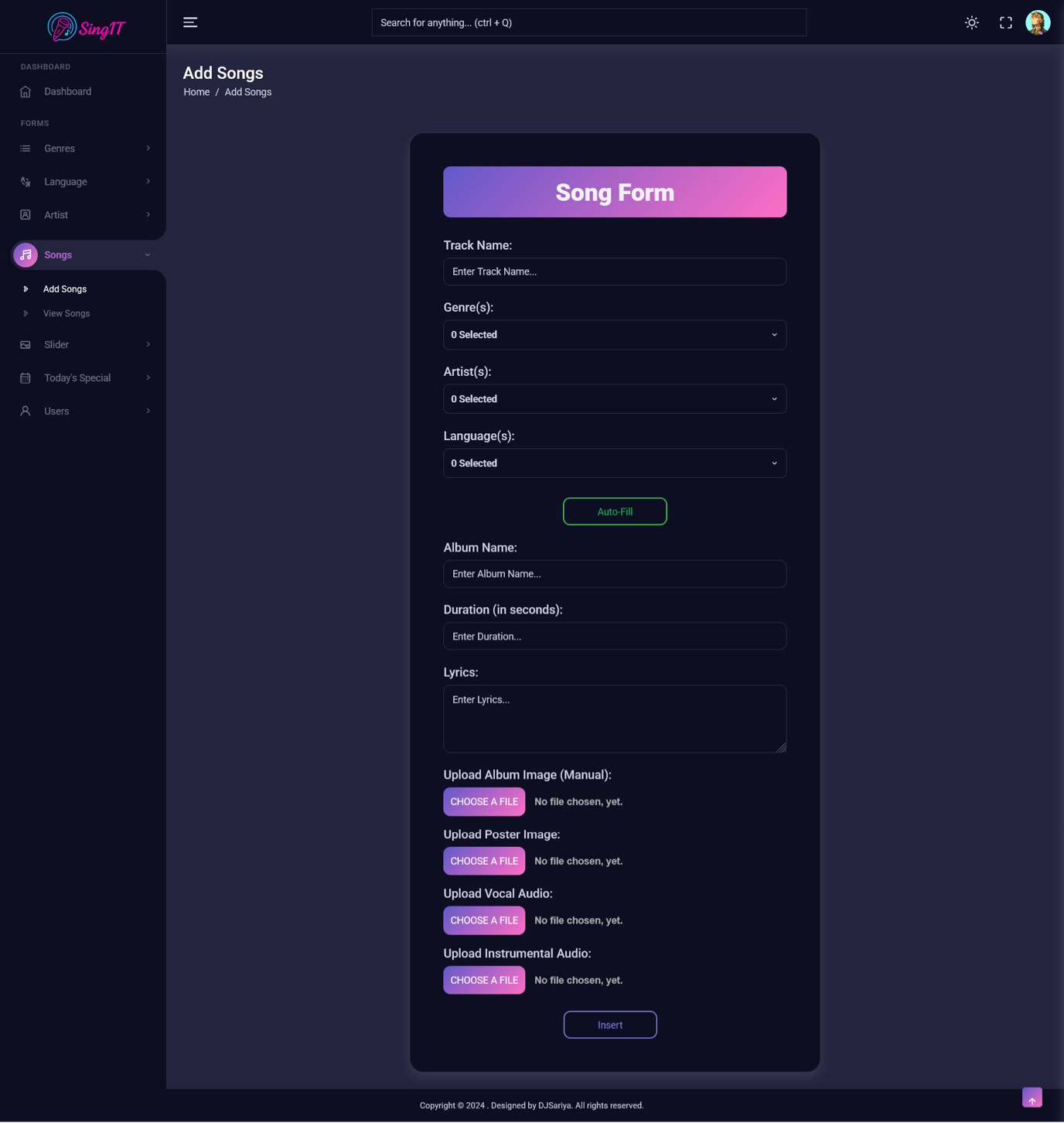
* 1. **Earned-value and burn-chart**

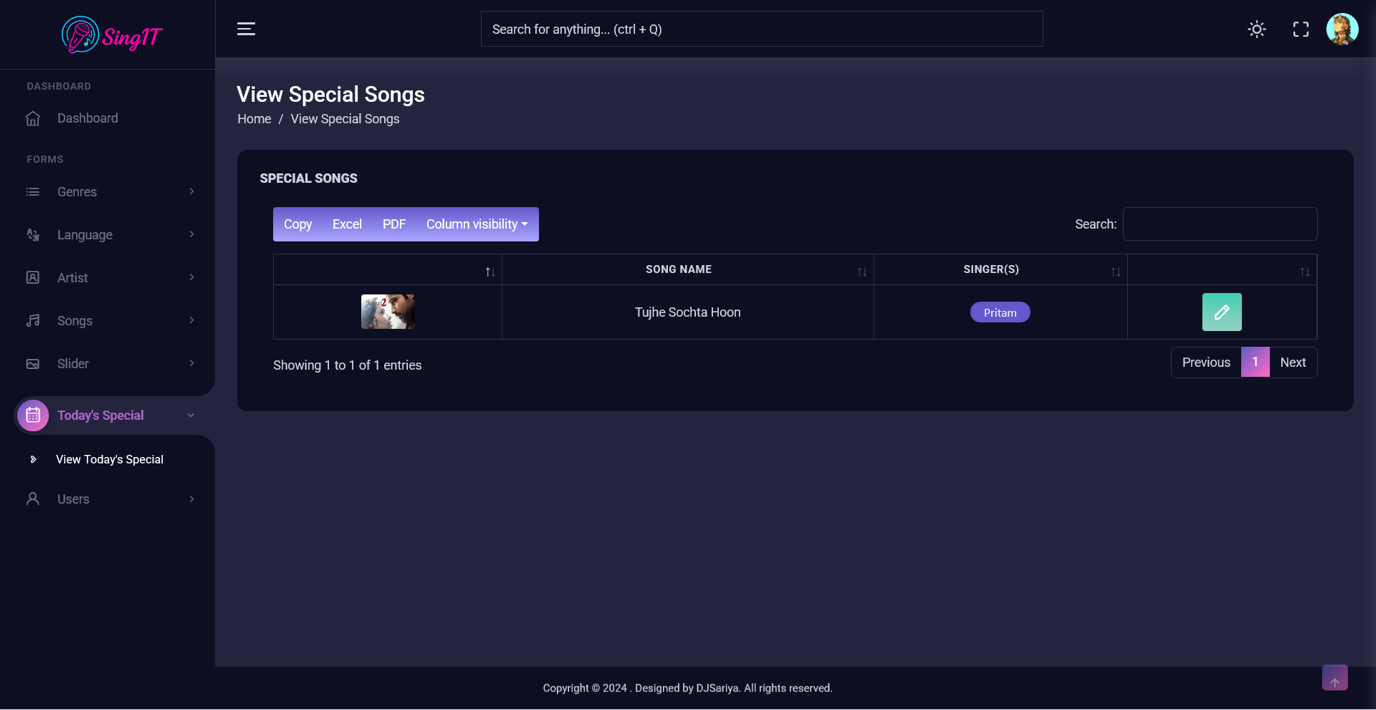
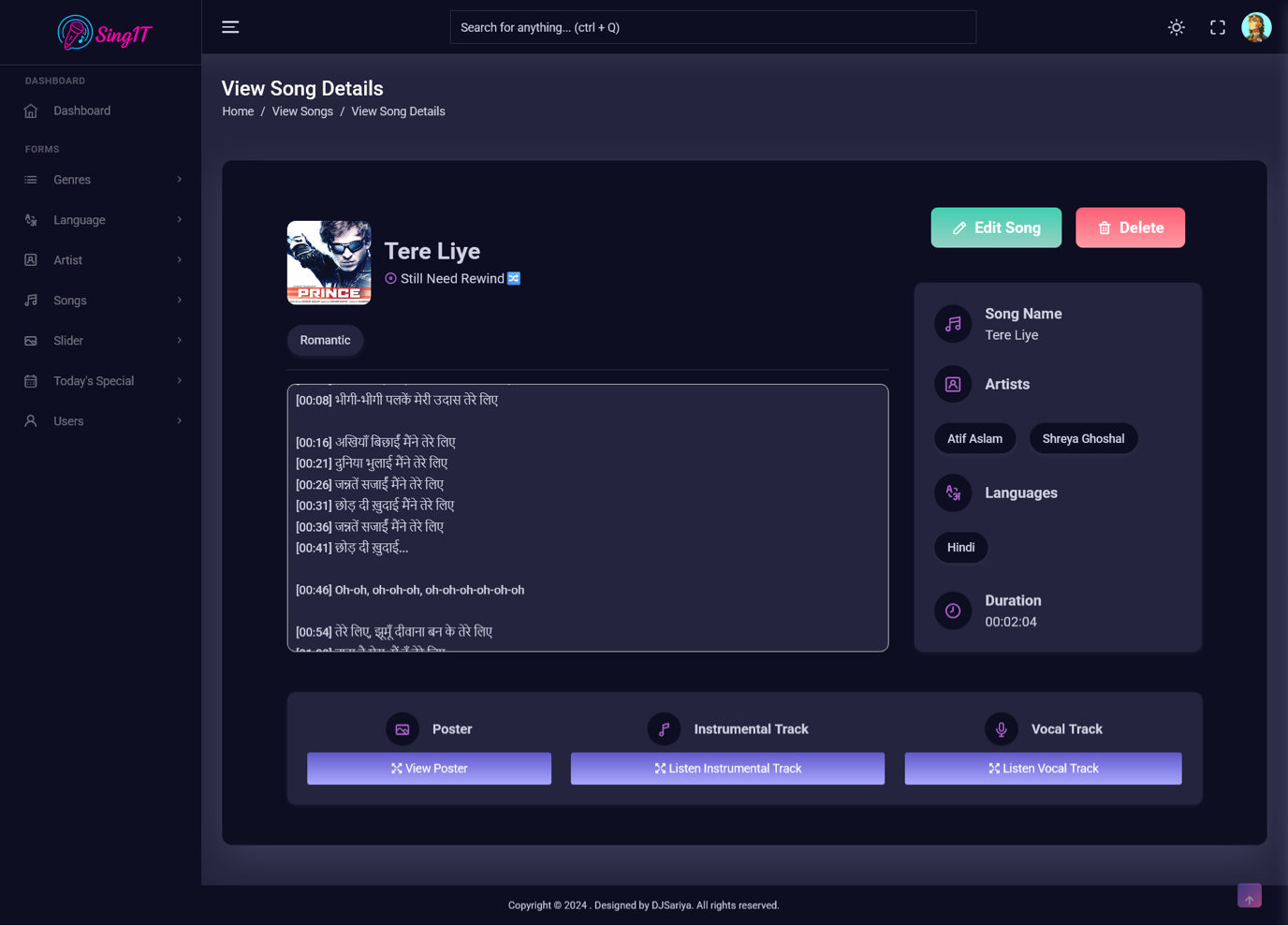
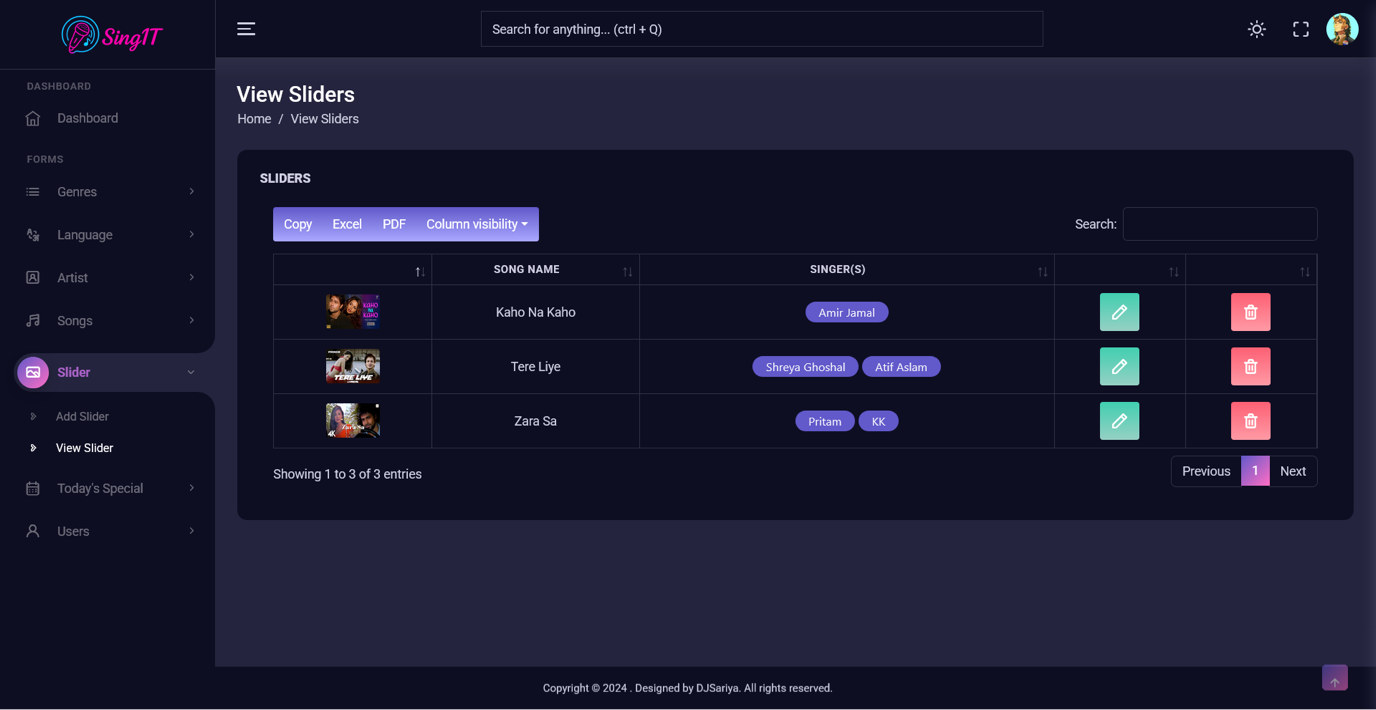
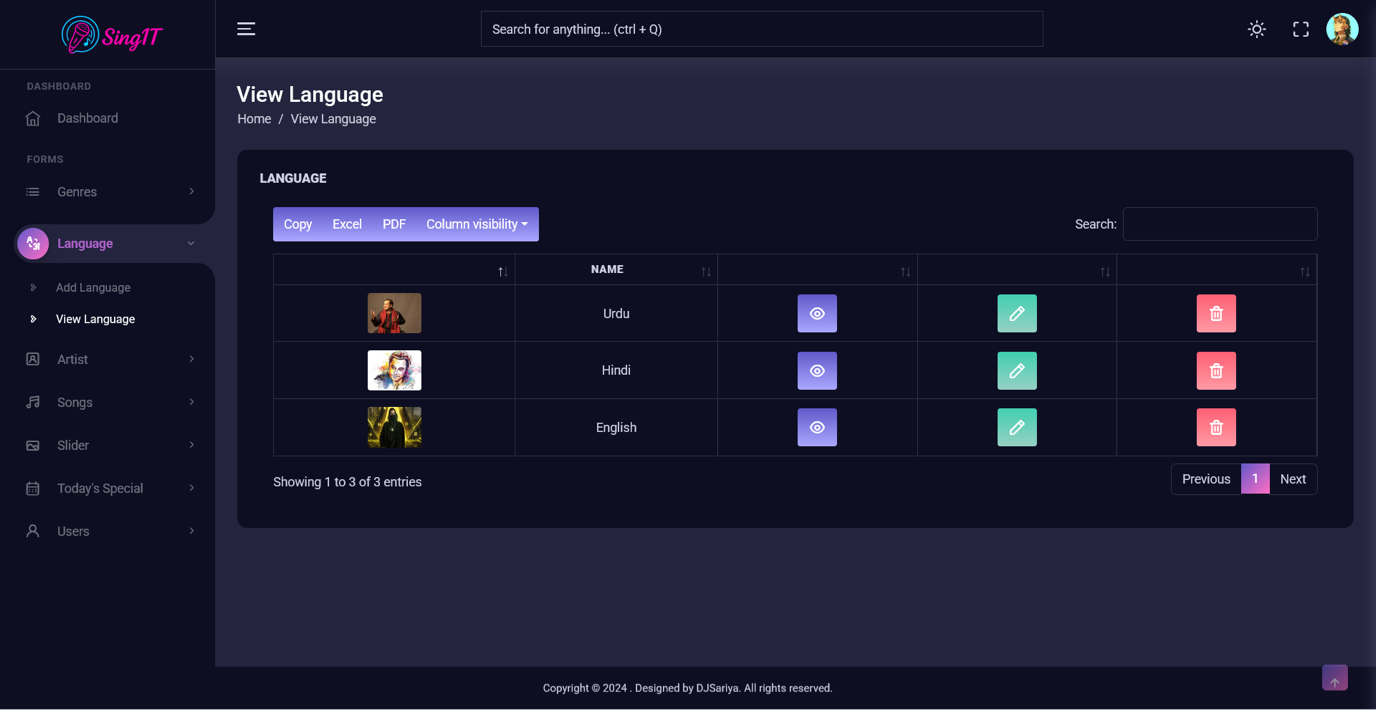
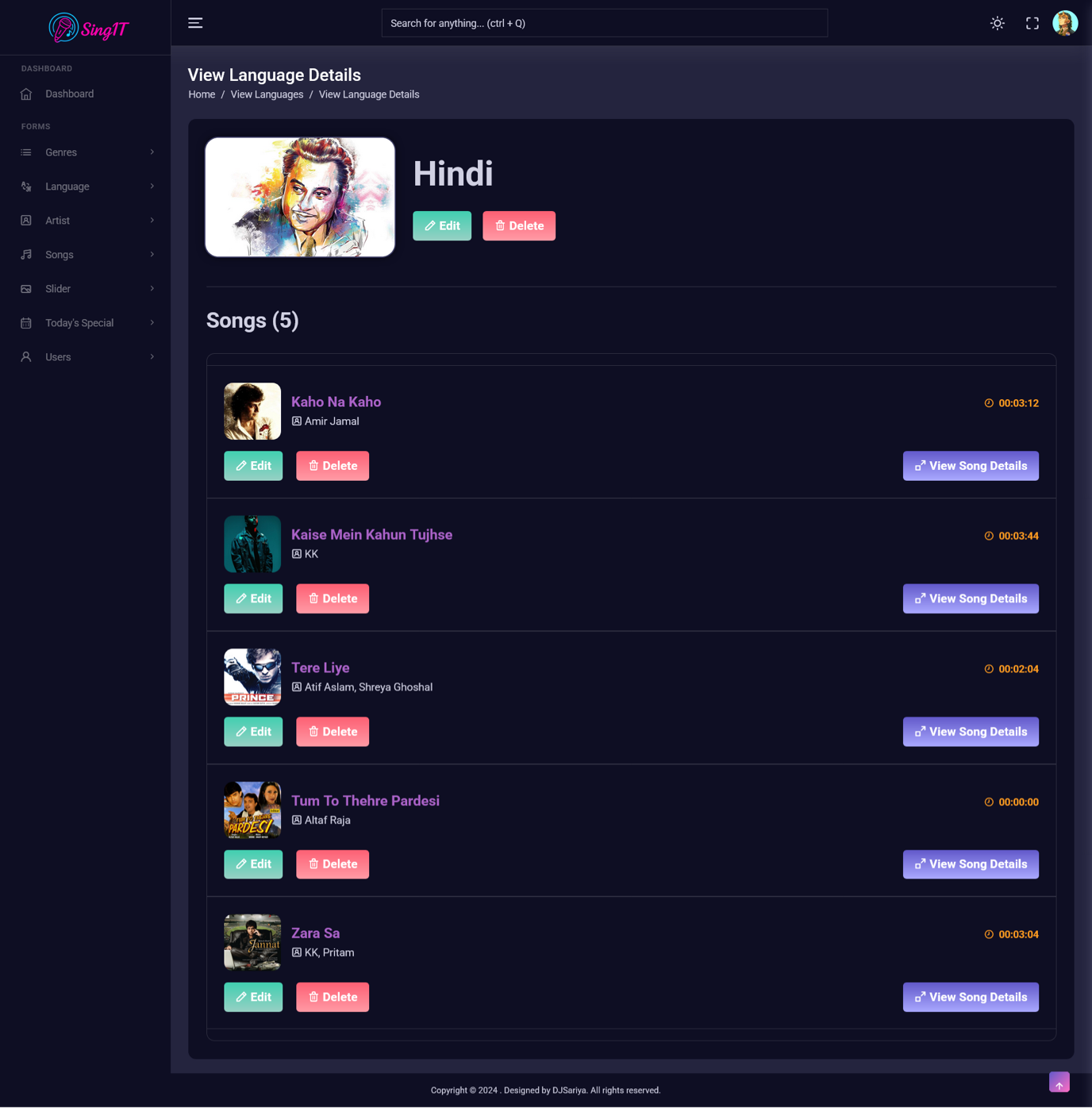
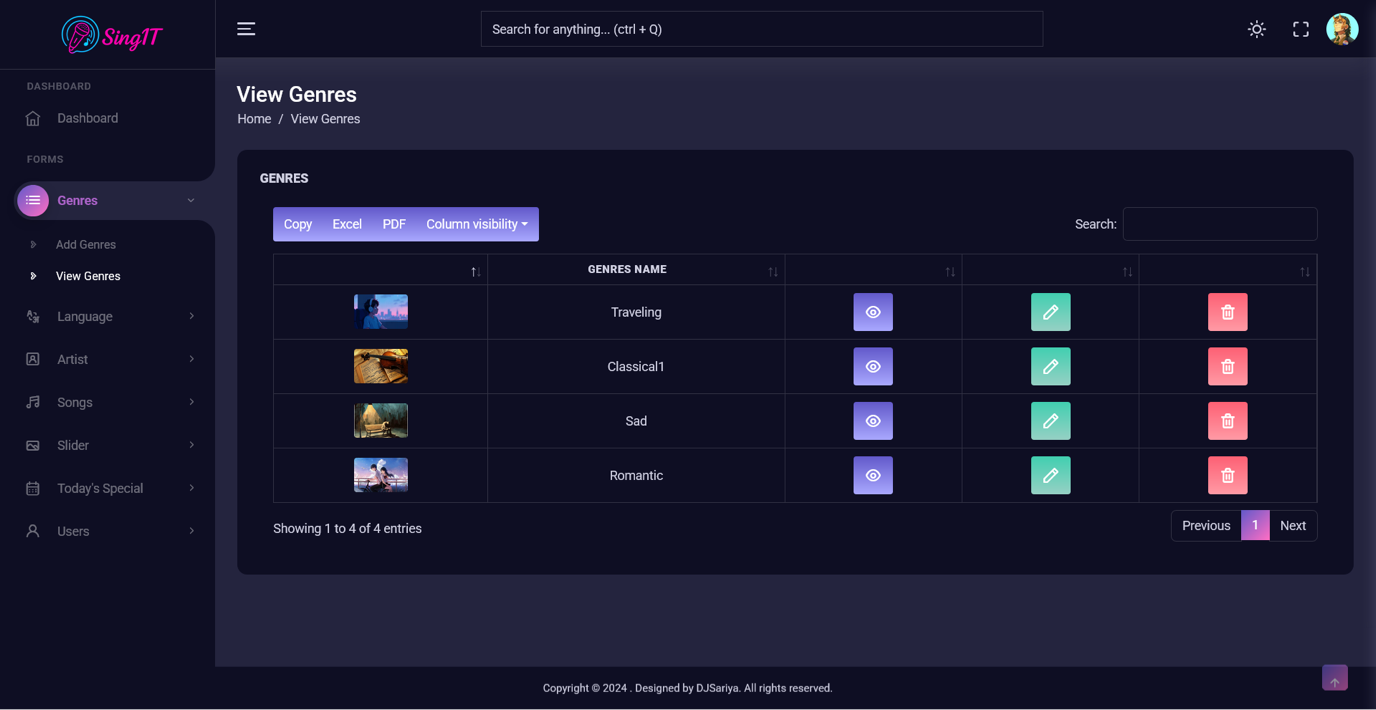
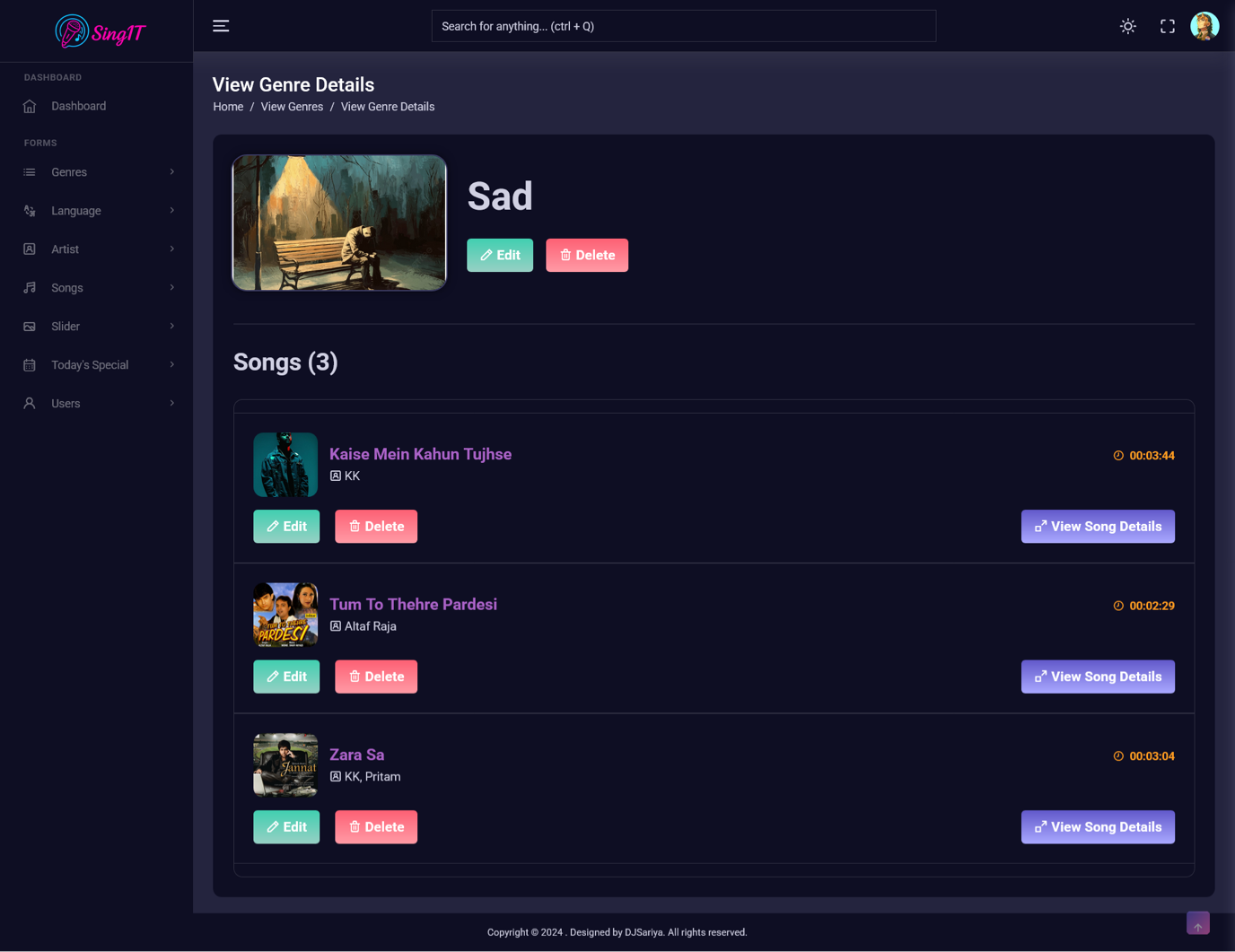
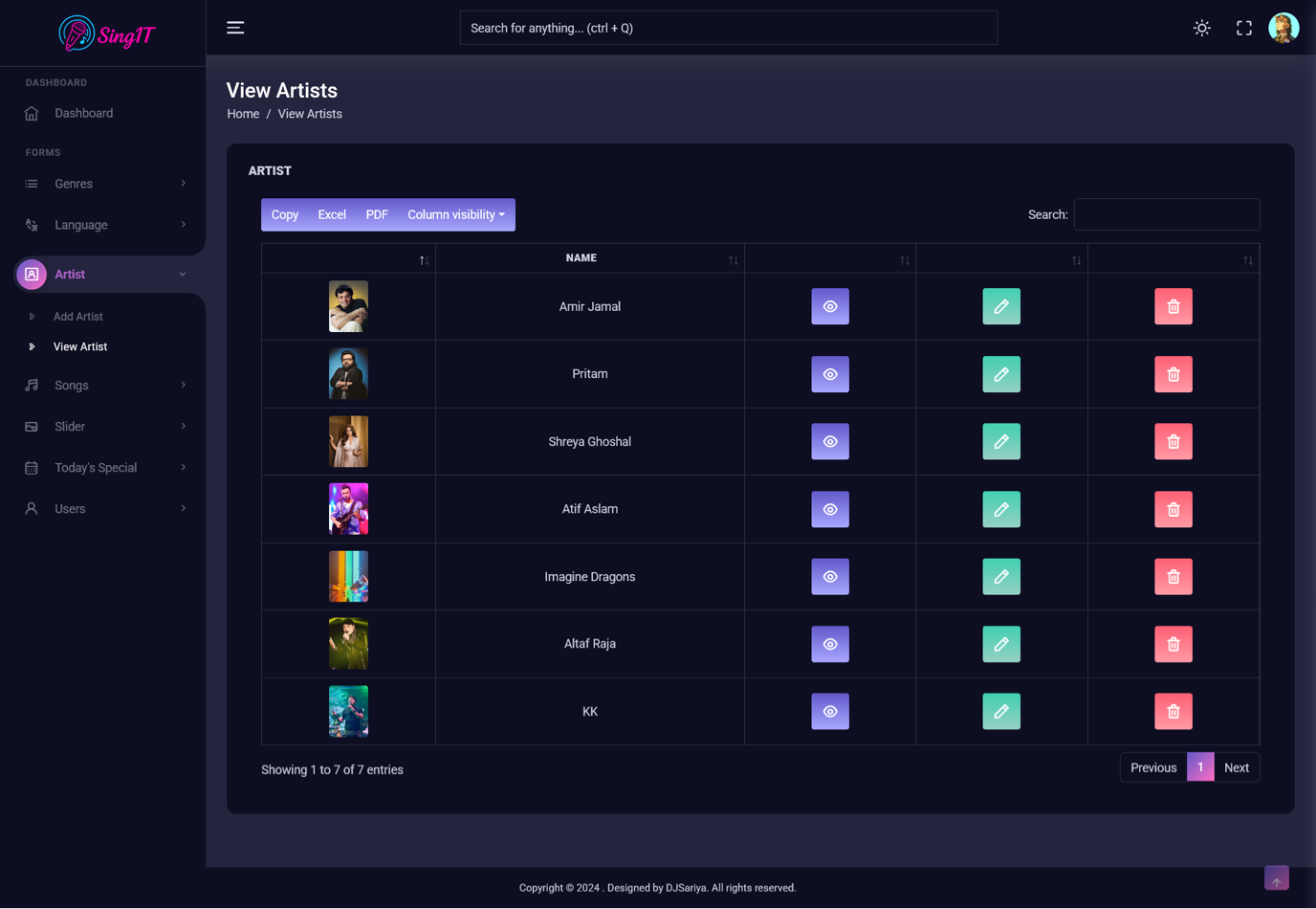
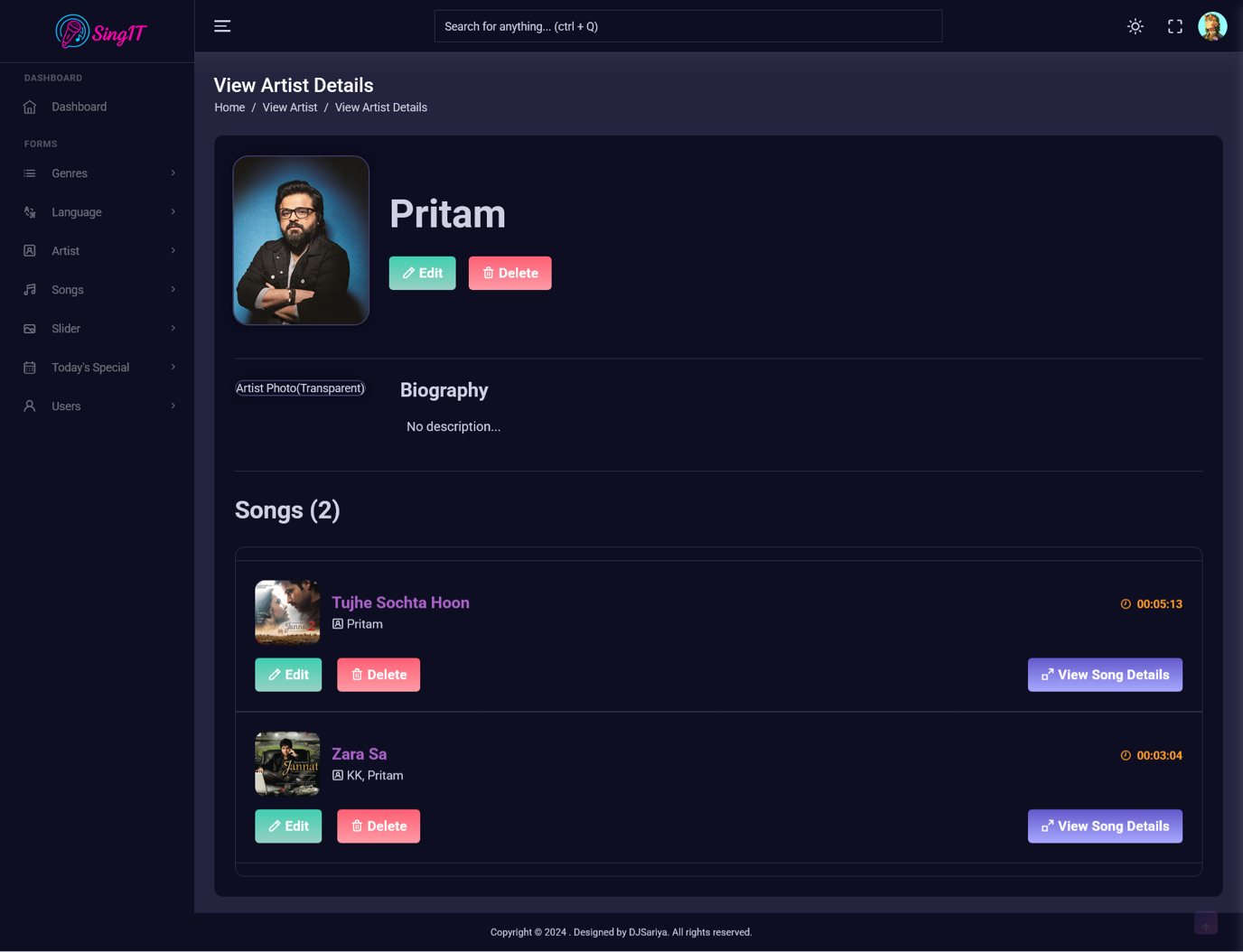
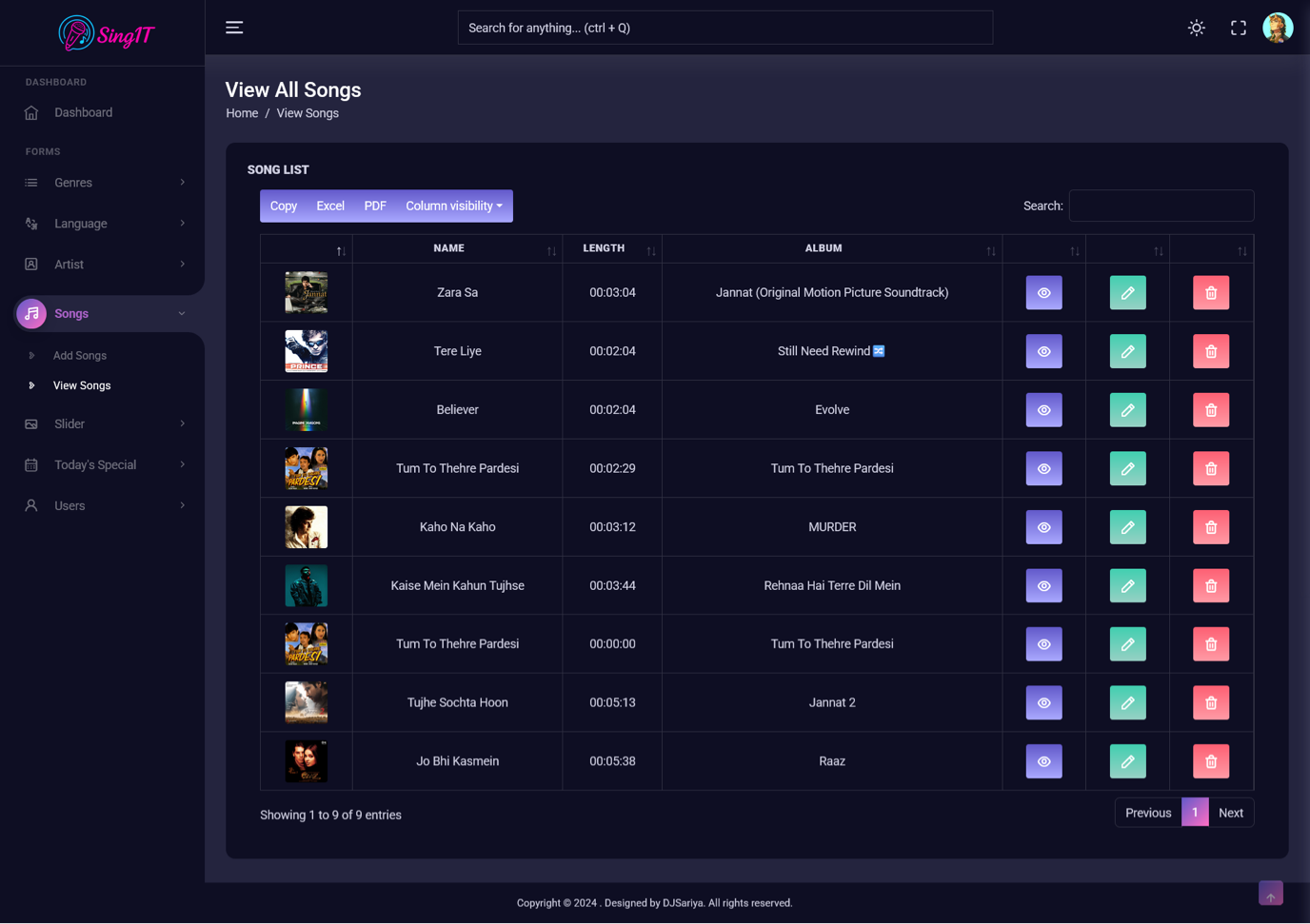
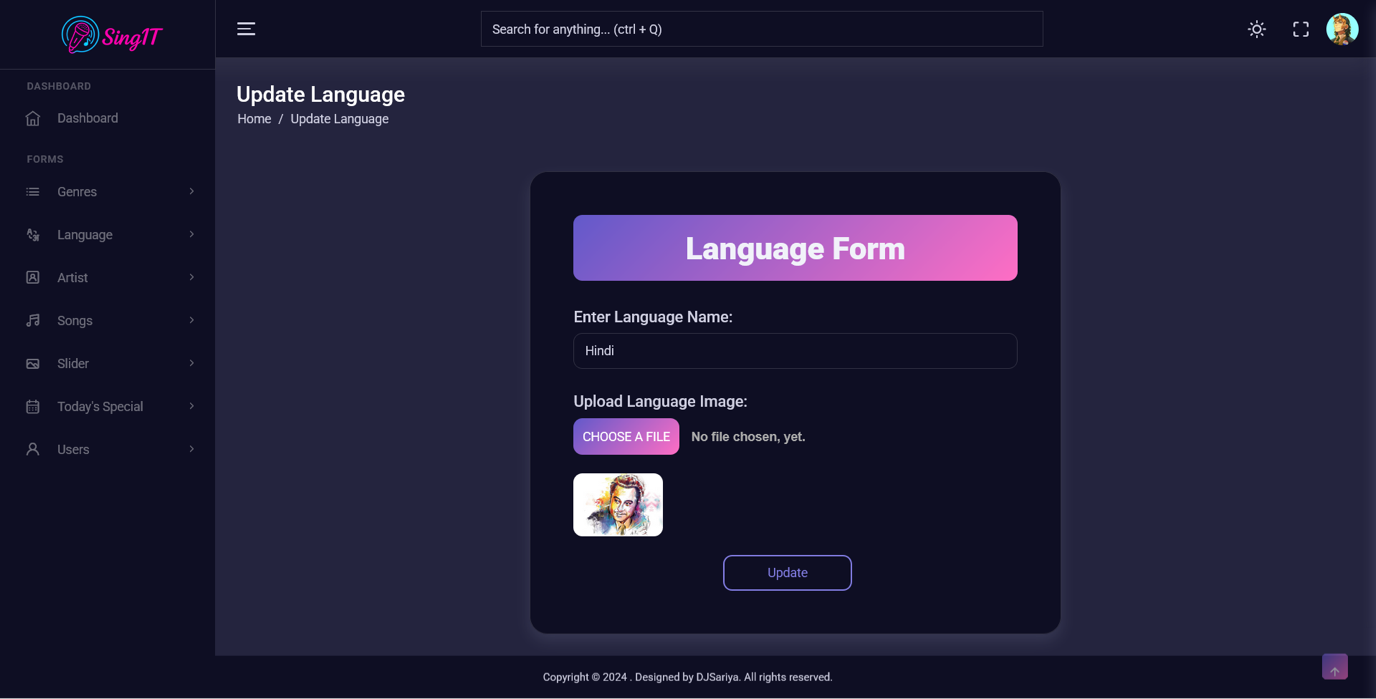
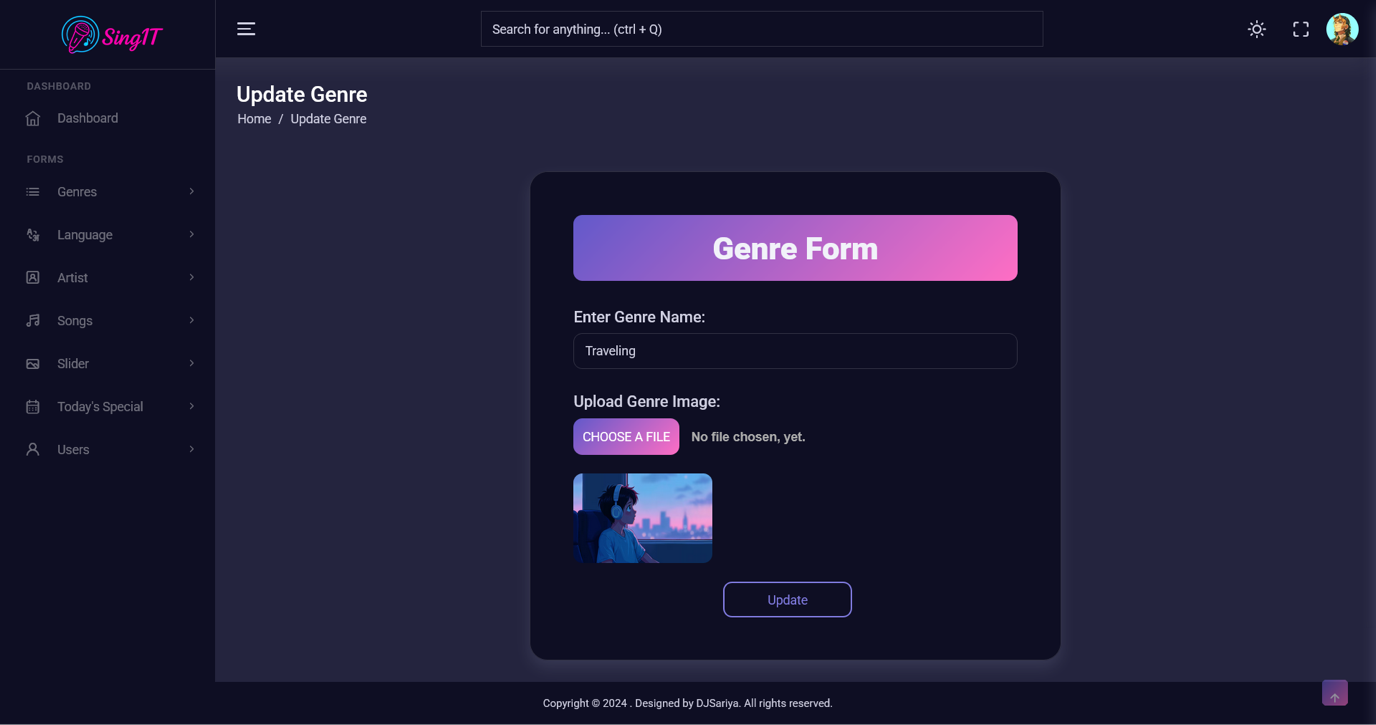
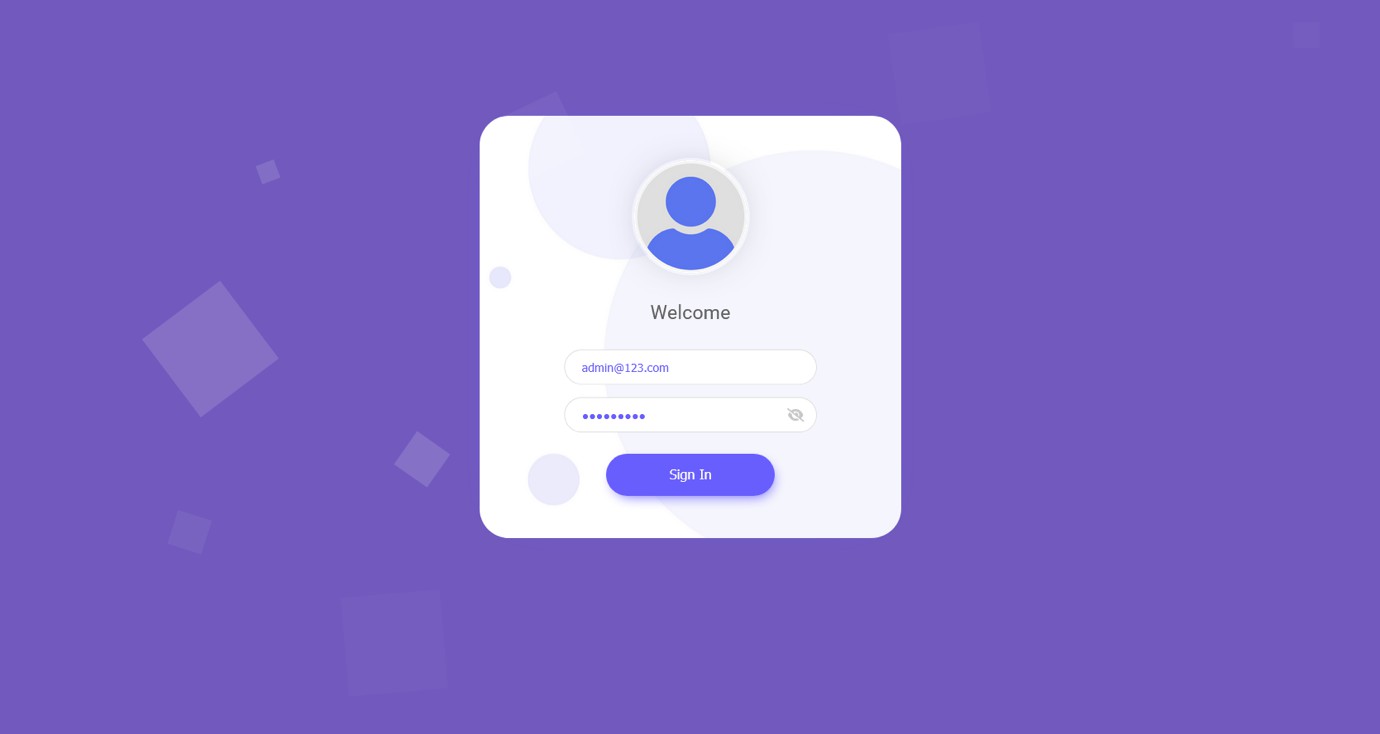
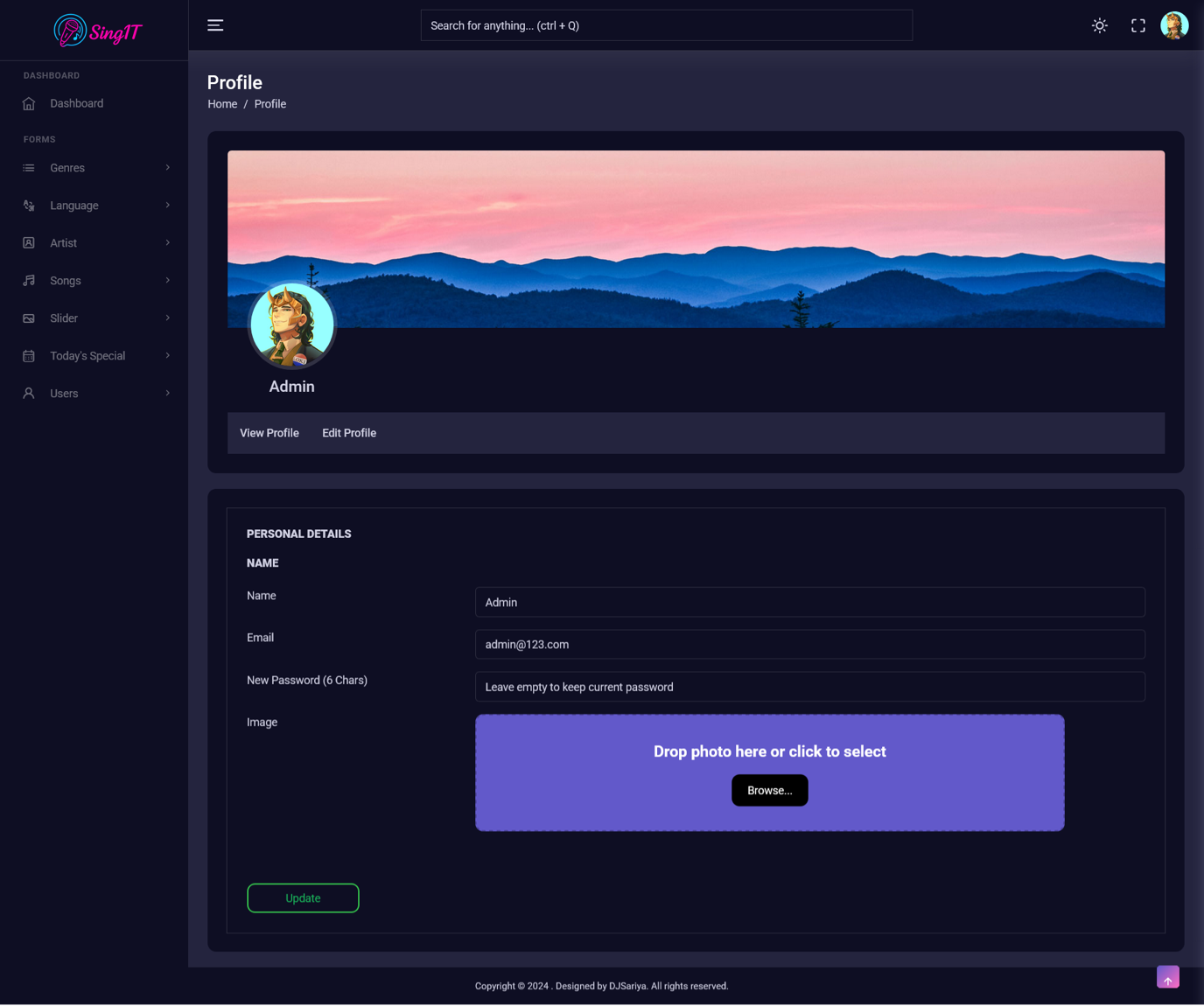
****

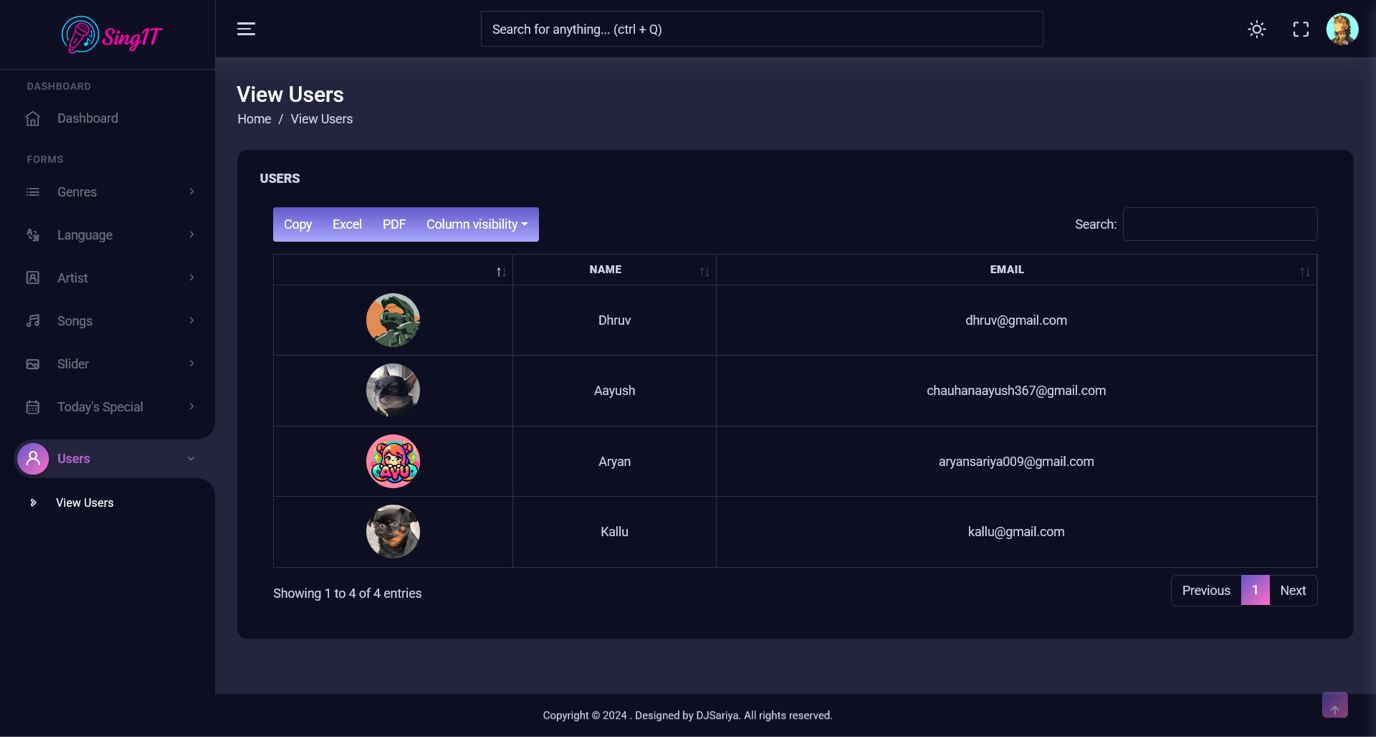
# SCREENSHOTS

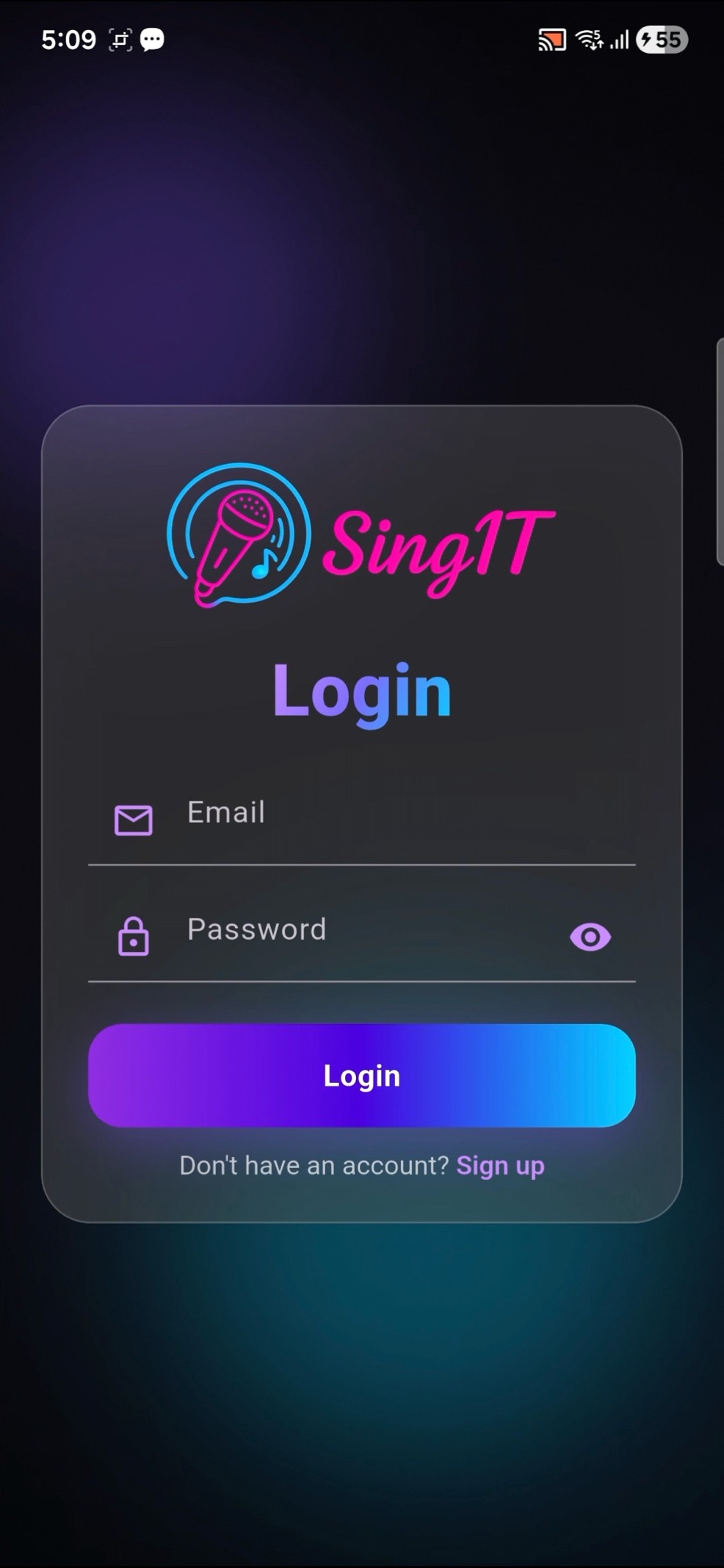
****

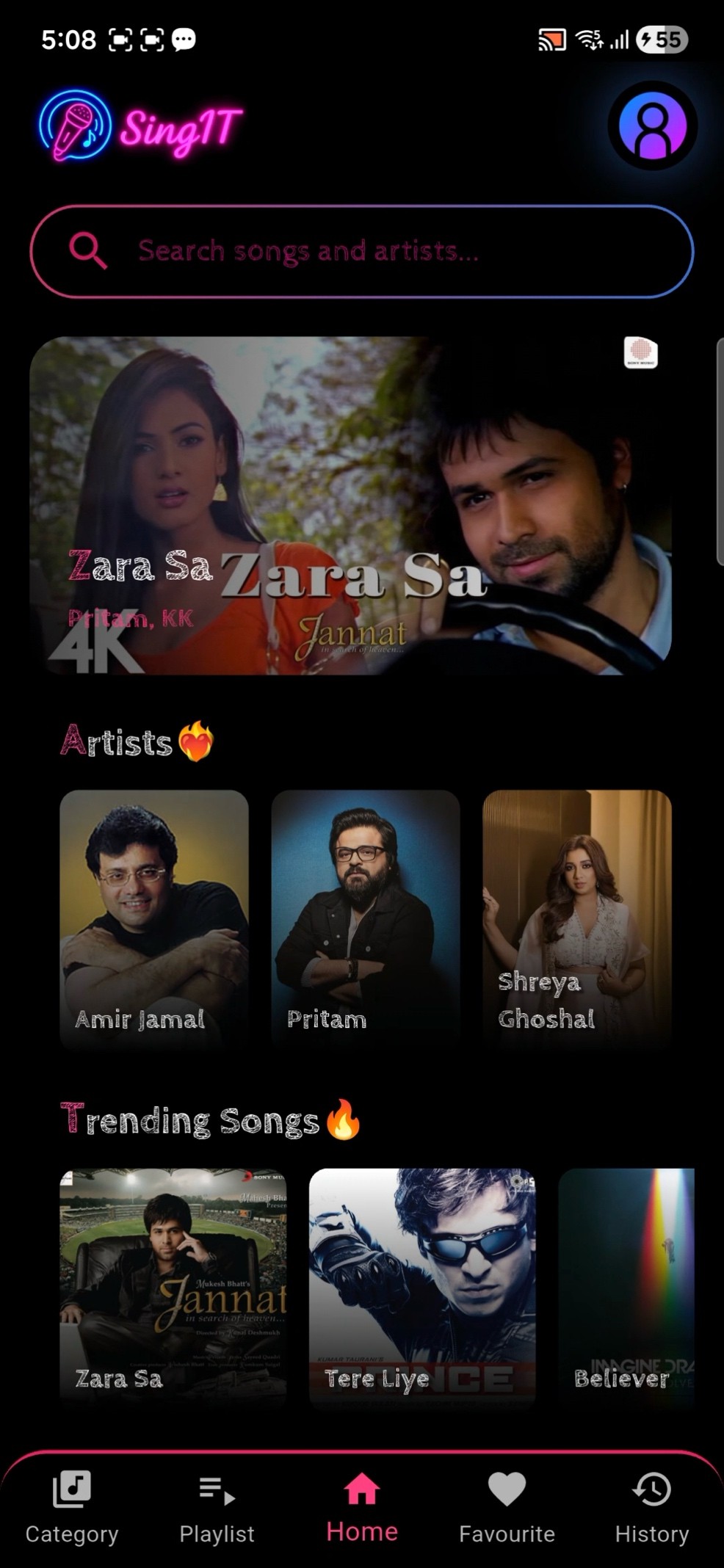


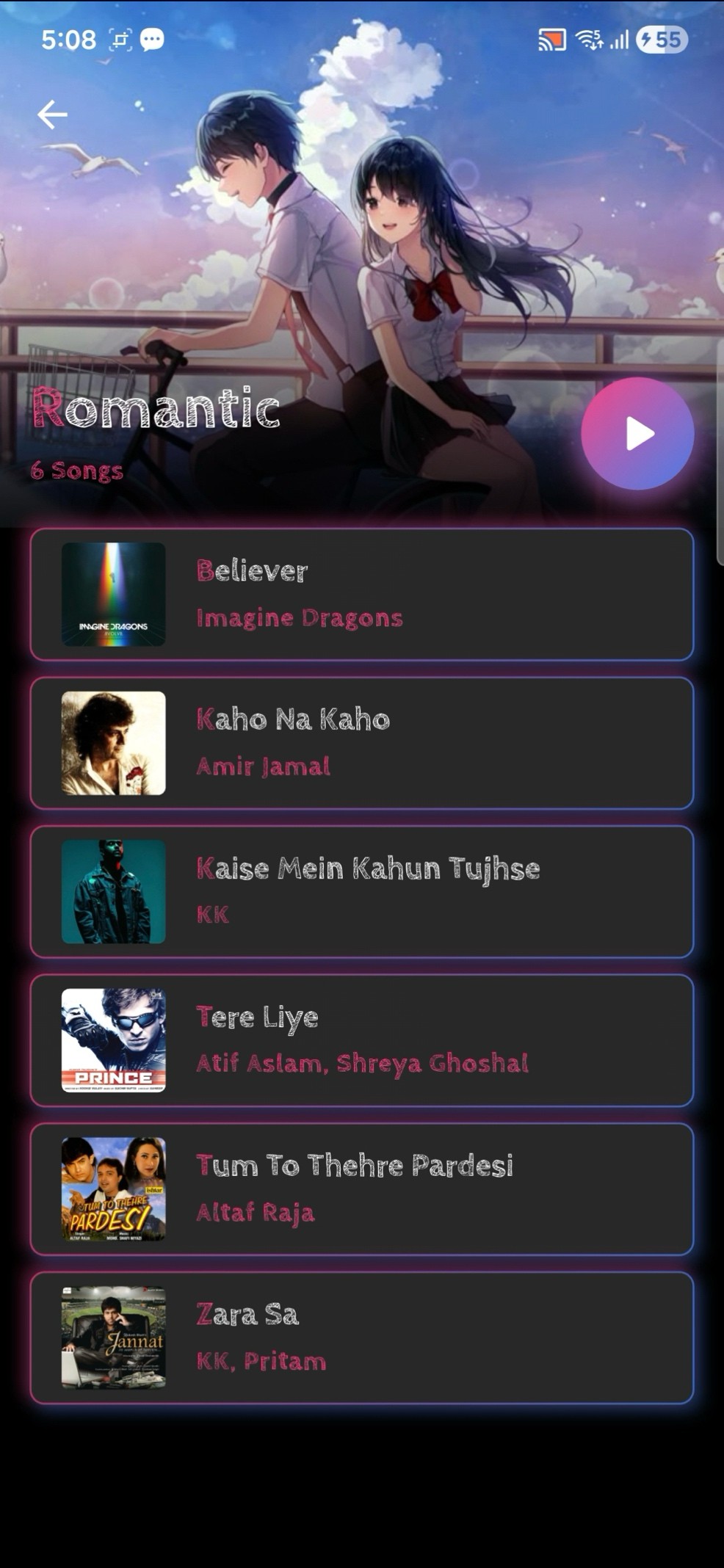


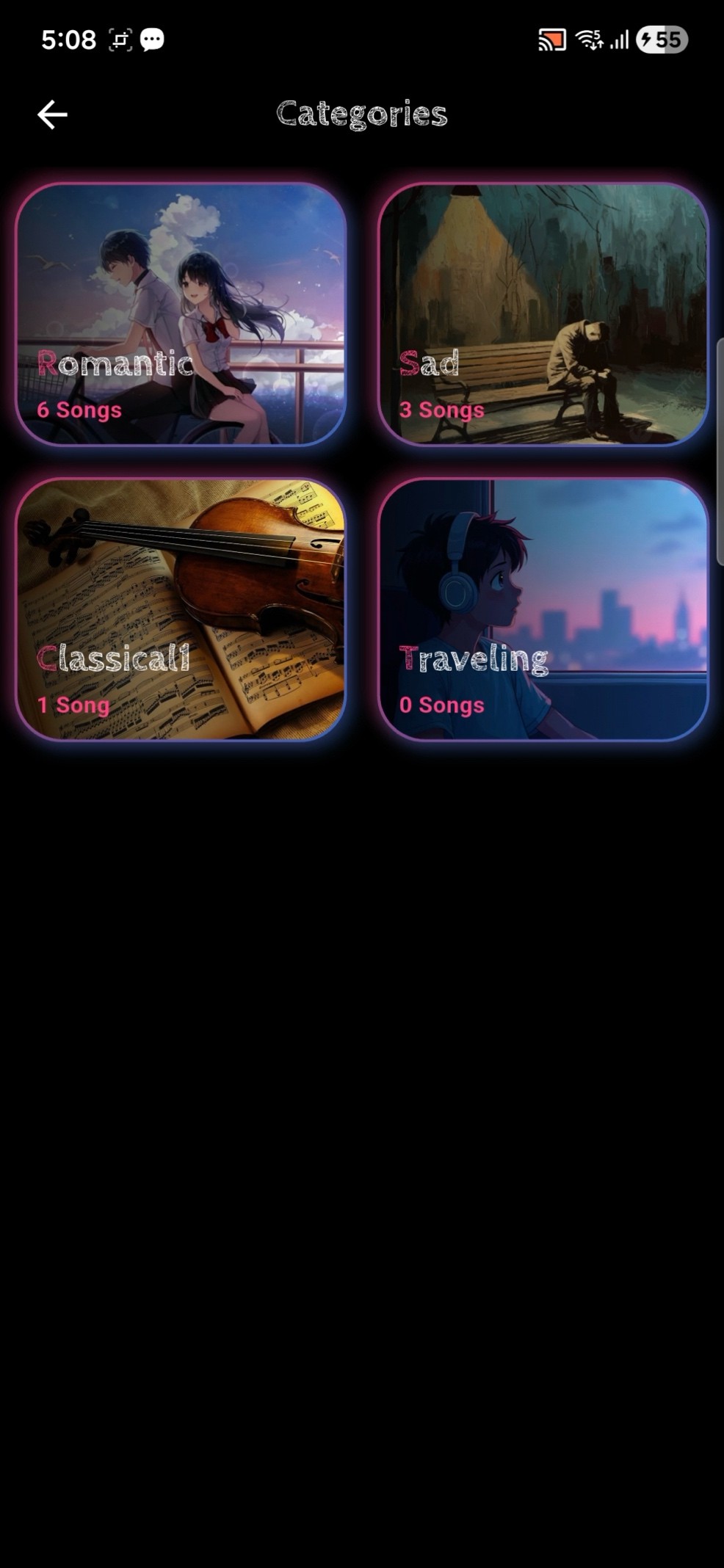


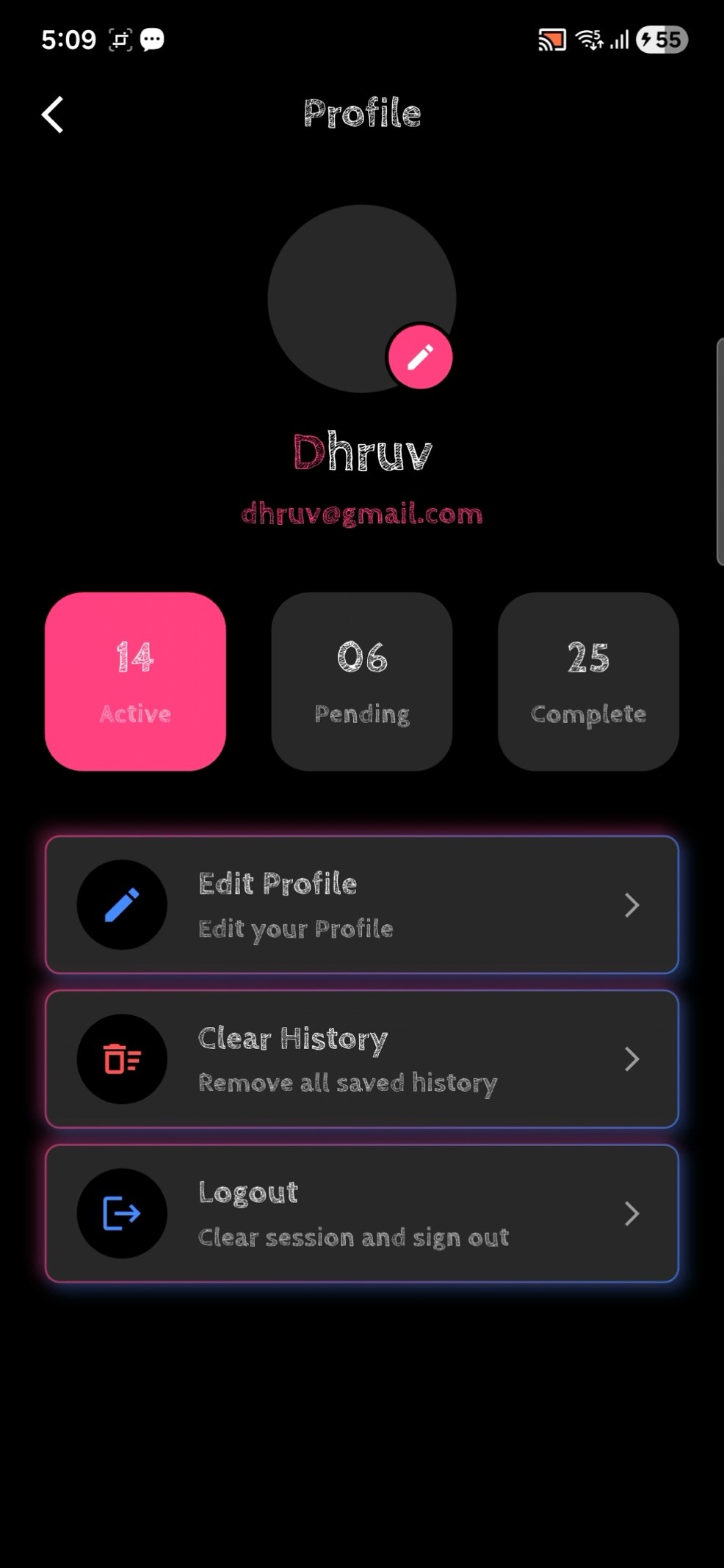




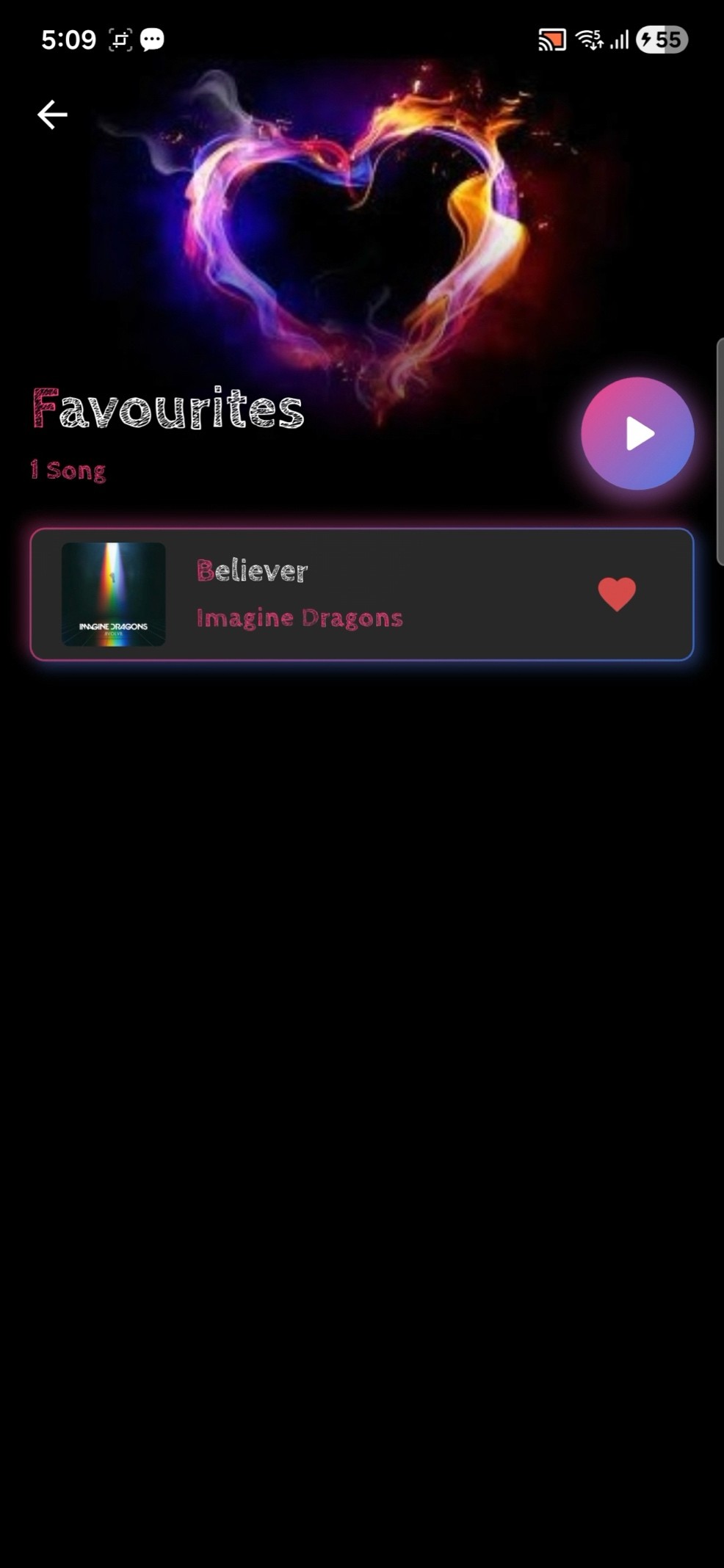


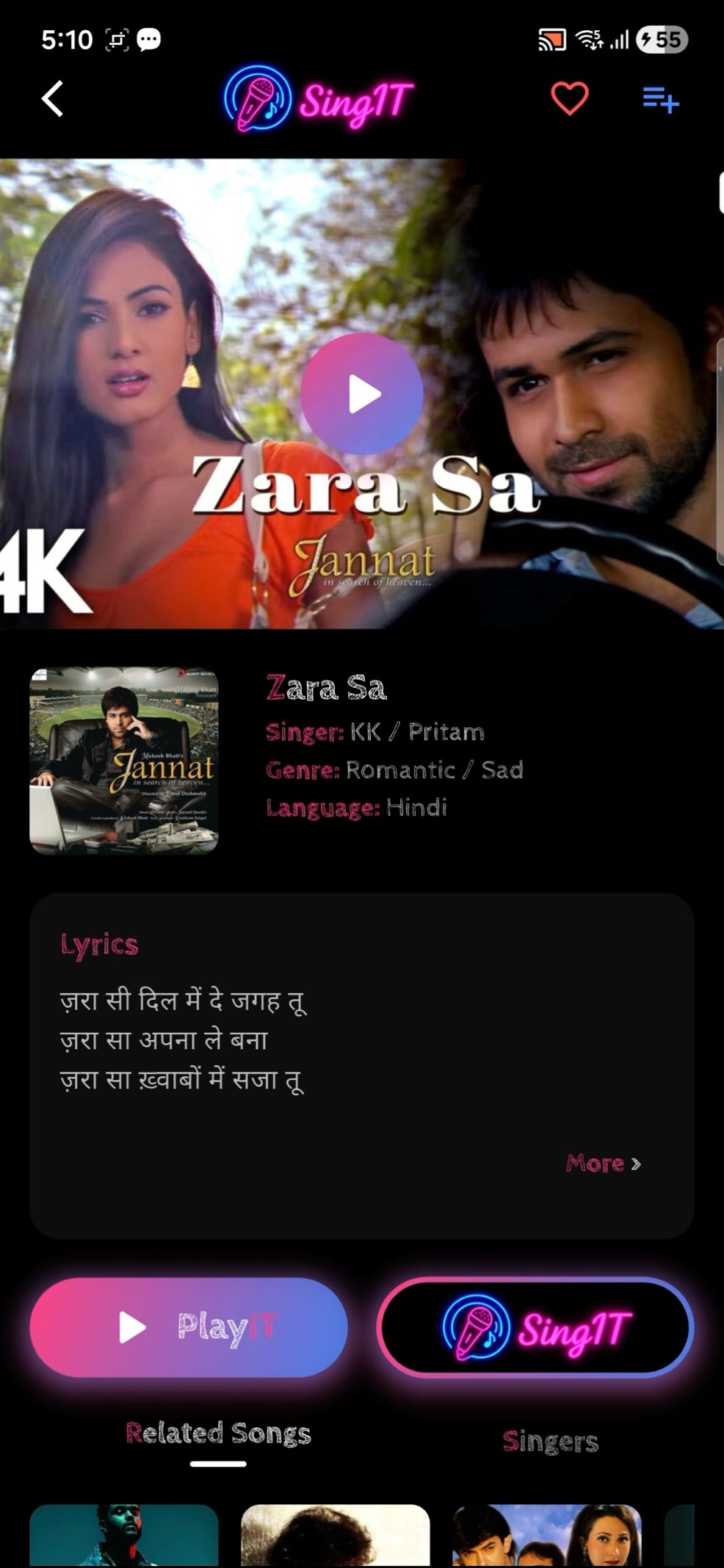


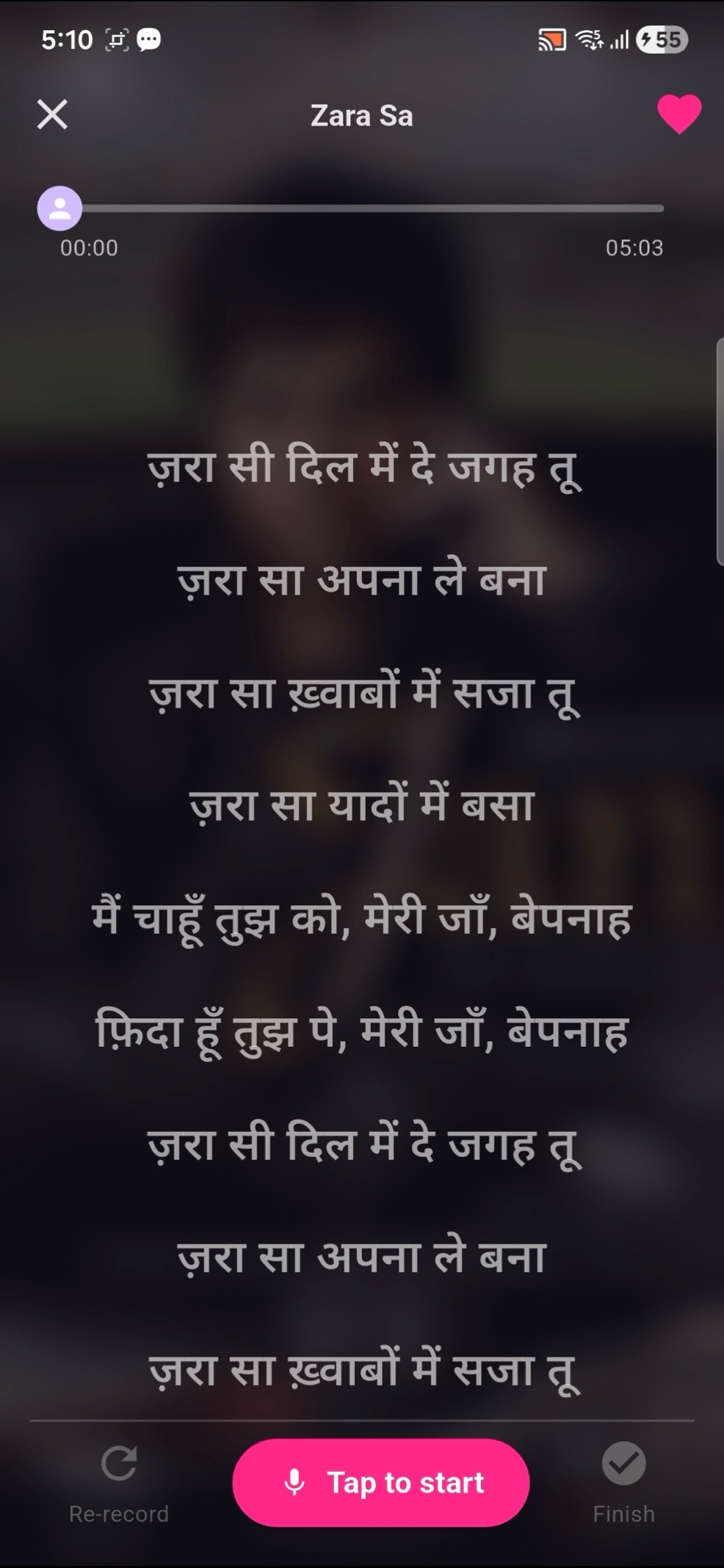


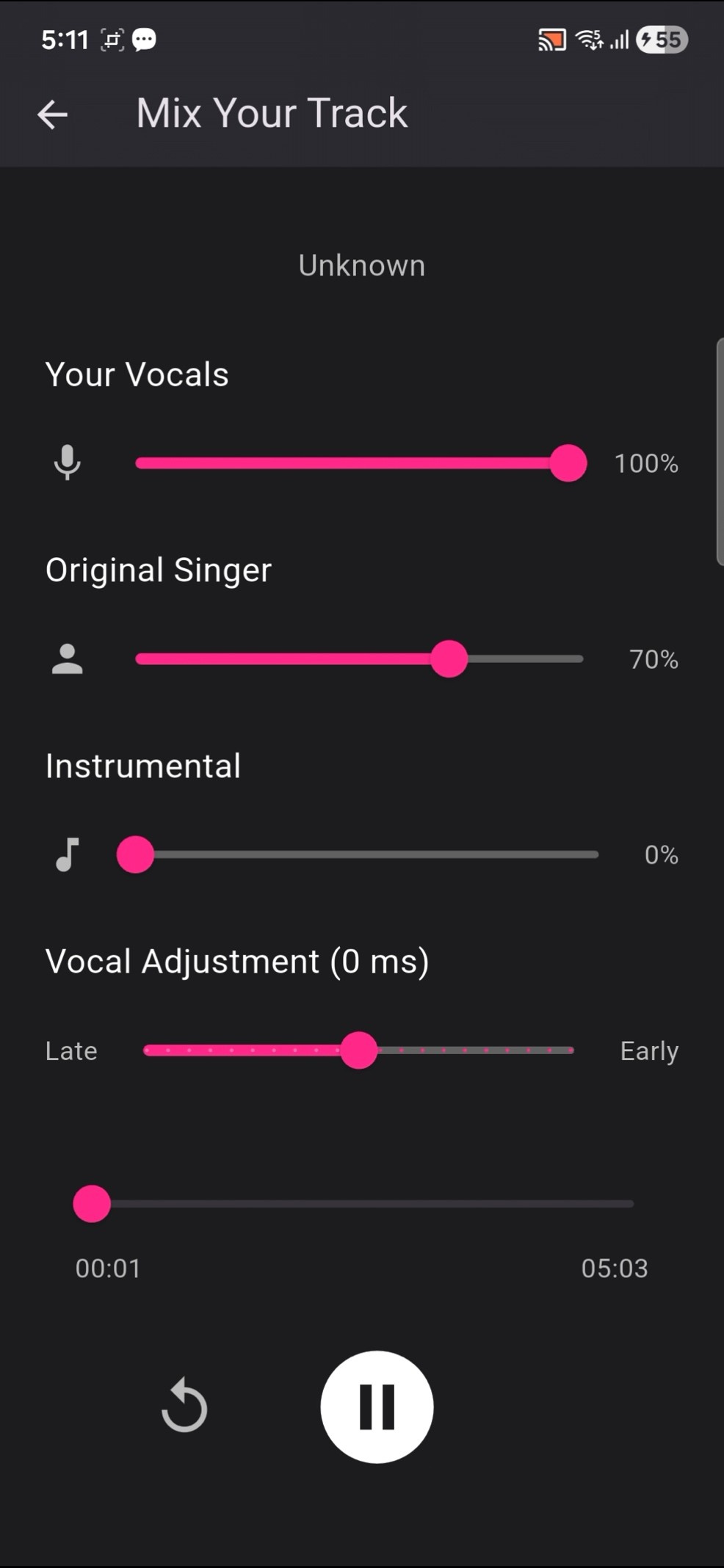


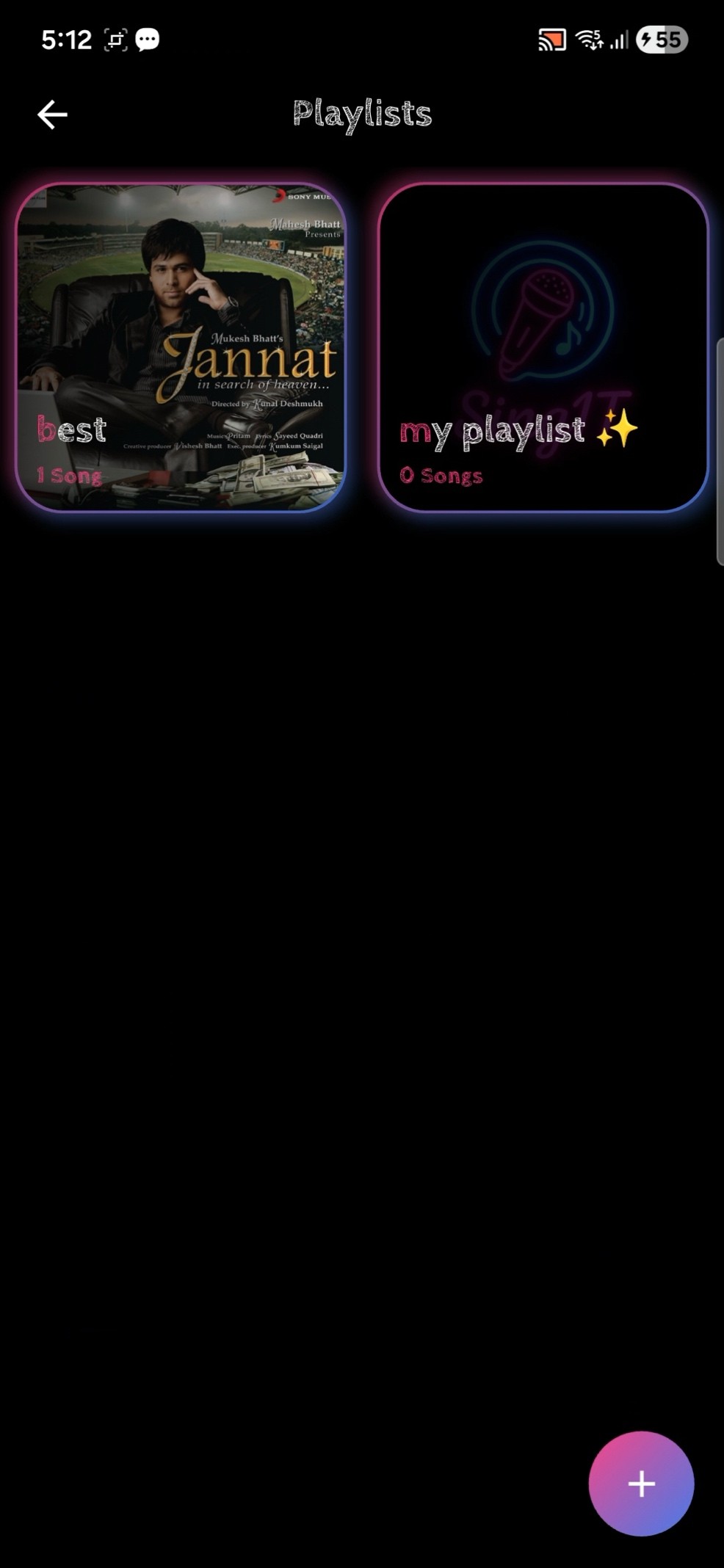


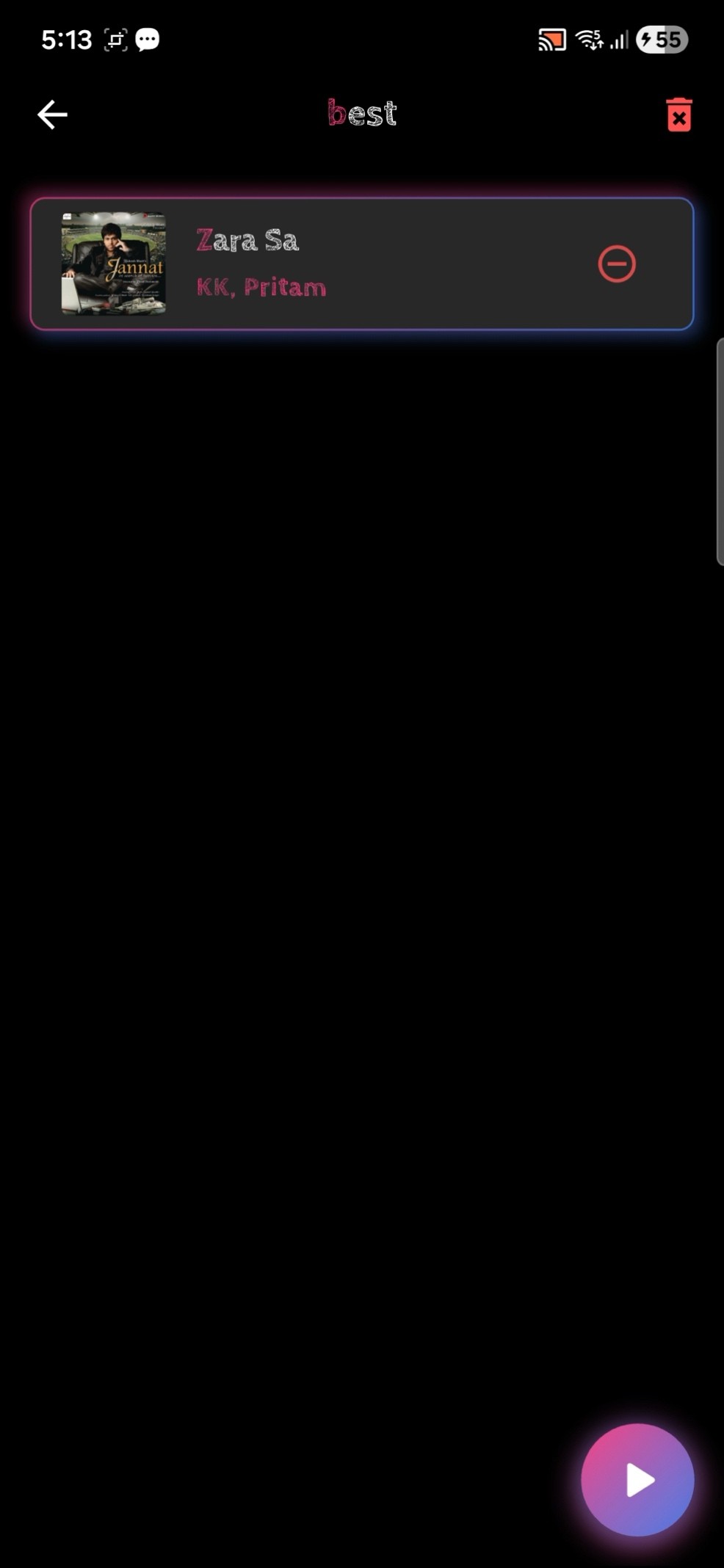


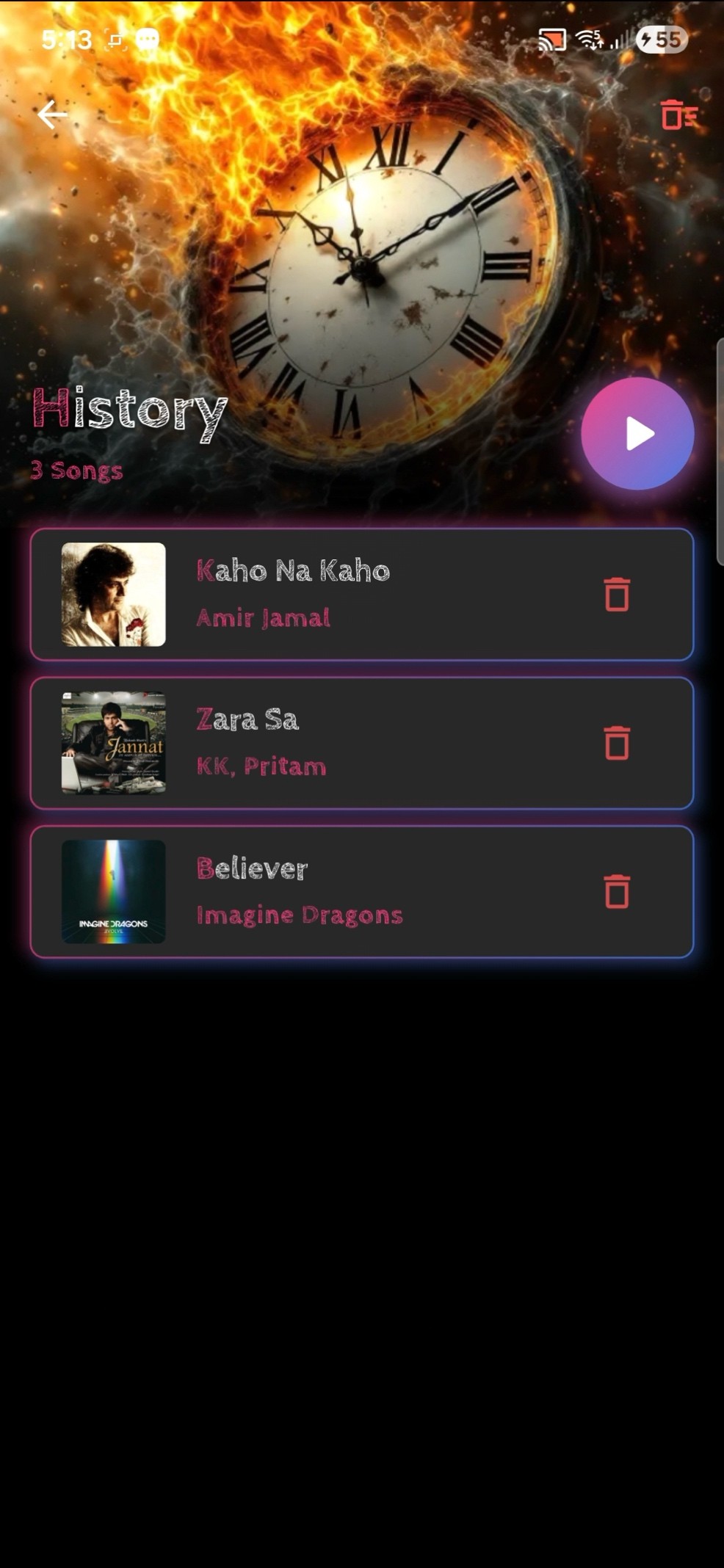












1. **PROPOSED ENHANCEMENT**

* **AI-based Audio Improvement:**
  + Add noise reduction, vocal enhancement, and auto-tuning features using machine learning models.
* **Advanced Playlist Automation:**
  + Introduce smart playlists that auto-group songs by user mood, history, or listening patterns.
* **Real-Time Lyrics Translation:**
  + Integrate multilingual translation so lyrics can be viewed in any preferred language.
* **Collaboration Mode:**
  + Allow two or more users to collaborate on audio/recordings using shared sessions.
* **Custom Audio Effects:**
  + Provide filters like echo, reverb, bass boost, and equalizer presets for enhanced audio mixing.
* **Performance Analytics Dashboard:**
  + Show user-specific stats like recording count, playback trends, time spent, and improvement charts.

# CONCLUSION

The proposed system, **SingIT**, successfully delivers an innovative and interactive karaoke experience optimized for Android devices. By intelligently separating each song into its core components—original track, instrumental version, and user- recorded vocals—the application offers a highly flexible and customizable singing environment. The integration of a dynamic sound-bar mechanism allows users to seamlessly switch or blend between audio modes in real time, while precise lyric synchronization ensures a smooth and guided performance.

https://flutter.dev/learn

# BIBLIOGRAPHY

https://[www.codeofaninja.com/create-simple-rest-api-in-php/](http://www.codeofaninja.com/create-simple-rest-api-in-php/) https://[www.w3schools.com/graphics/google\_maps\_intro.asp](http://www.w3schools.com/graphics/google_maps_intro.asp) https://stackoverflow.com/?gad\_source=1&gclid=CjwKCAiA6aW6BhBqEiwA6KzDc8dlD2 5tlurLWp8ZRkkVar6jQ6s3HHf4I5bIV6GvaEetMzECBNembxoC8DIQAvD\_BwE https://[www.geeksforgeeks.org/c-programming-language/](http://www.geeksforgeeks.org/c-programming-language/)