INSPOT: WHERE WORDS FAIL AND MUSIC SPEAKS.

FINAL REPORT

Submitted by

SRI KIRUPA P **(2116220701286)**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY, CHENNAI

MAY 2025

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified for this Project titled **"INSPOT: where words fail and music speaks."** is the bonafide work of **"SRI KIRUPA P (2116220701286)"** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Dr. N. Duraimurugan., M.E., Ph.D.,
SUPERVISOR
Professor
Department of Computer Science
and Engineering,
Rajalakshmi Engineering
College, Chennai-602 105.

Submitted to Project Viva-Voce Examination held on _____

Internal Examiner **External Examine**

ABSTRACT

Inspot is an innovative Android application that blends the social connectivity of Instagram with the immersive music experience of Spotify, creating a unique platform for music lovers and social users alike. The app enables users to explore, stream, and manage music while simultaneously engaging with a vibrant community through posts, forums, messages, and event participation. With Firebase Authentication powering a secure login/signup process, users can personalize their experience through a profile system. The music module offers a structured and user-friendly interface for browsing and playing songs categorized by genre. Meanwhile, the forum-style post section allows users to share thoughts, reviews, and discussions — echoing the functionality of Instagram's feed. Real-time messaging powered by Firebase enhances direct interaction, and a dedicated events page keeps the community connected through music-based meetups or discussions. By combining two powerful domains — social media and music streaming — Inspot provides an integrated experience where users don't just listen to music but live and share it as a social experience.

ACKNOWLEDGMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.**, our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN**, **Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. KUMAR, M.E., Ph.D.**, Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our Project Coordinator, **Mr. N. DURAIMURUGAN ,** Assistant Professor Department of Computer Science and Engineering for his useful tips during our review to build our project and for his valuable guidance throughout the course of the project.

SRI KIRUPA P 2116220701286

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION	FULL FORM
UI	User Interface
UX	USER EXPERIENCE
API	APPLICATION PROGRAMMING INTERFACE
IDE	INTEGRATED DEVELOPMENT ENVIRONMENT
RAM	RANDOM ACCESS MEMORY
OS	OPERATING SYSTEM
FCM	FIREBASE CLOUD MESSAGE
JSON	JAVASCRIPT OBJECT NOTATION

1. INTRODUCTION

1.1 GENERAL

Music has always been a powerful medium of expression and connection. With the rise of digital platforms, music consumption has shifted from physical formats to streaming applications. Simultaneously, social media platforms have grown into vibrant communities where users share content and experiences. Inspot aims to blend these two powerful experiences—music streaming and social interaction—into a single unified platform. By merging features from Spotify (music access) and Instagram (social networking), Inspot creates a space where users can enjoy music, discover events, share thoughts, and interact with a like-minded community.

1.2 OBJECTIVE

The primary objective of Inspot is to develop an Android application that seamlessly combines music streaming with social media functionalities. The app aims to provide users with a platform where they can explore and stream music based on genre, artist, or personalized recommendations. Beyond just listening, Inspot encourages community engagement through a social feed where users can post updates, share music reviews, or recommend playlists. To foster real-time interaction, a messaging system powered by Firebase is integrated, enabling users to chat and connect instantly. Additionally, Inspot includes an event feature that promotes music-related gatherings or virtual meetups, enhancing community participation. A secure and reliable user authentication system using Firebase ensures safe access and user data protection throughout the app.

1.3 EXISTING SYSTEM

Current applications in the market tend to specialize in either music streaming or social networking but rarely both. Spotify, Apple Music, and YouTube Music offer powerful music streaming capabilities but lack social feed and interactive event features. On the other hand, platforms like Instagram and Facebook allow social interaction but do not provide an integrated music listening experience. Users often need to switch between apps to share what they are listening to or to discuss music with peers. This fragmentation creates a gap in the user experience, which Inspot aims to bridge by bringing both aspects into one cohesive platform.

CHAPTER 2
LITERATURE SURVEY

[1] *"Multi-Modal Interaction in Music Sharing Applications"* (2022) by Lina Zhang and Erik Petersen examines how multi-modal elements—such as text, audio, emojis, and visual cues—enhance user interaction in music-sharing platforms. The paper finds that combining messaging with rich media elements like song previews and waveform visualizations leads to more engaging user experiences. This directly supports Inspot's vision of blending expressive messaging with interactive music content to foster a vibrant social environment.

[2] *"Music Recommender Systems: Challenges and Opportunities"* (2020) by Markus Schedl et al. surveys existing music recommender systems and categorizes their approaches into collaborative filtering, content-based filtering, and hybrid models. The paper identifies cold start problems, diversity, and user modeling as key challenges. It emphasizes the importance of contextual and personalized recommendations in enhancing user experience. This study supports the recommendation engine goals in Inspot's music module.

[3] *"A Comparative Study of Firebase and Other Cloud Backend Services"* (2021) by K. Anusha and P. Aravind evaluates Firebase against AWS Amplify and Google App Engine in terms of authentication, database, and real-time communication. The paper highlights Firebase's ease of integration, real-time database, and cost-effectiveness, which justifies its use in Inspot for both messaging and user management.

[4] *"Integrating Social Media Features into Mobile Applications"* (2022) by Lina Zhang and Tomas Horak discusses the design considerations for embedding social features like feeds, likes, and comments into mobile apps. The study identifies

improved engagement but also potential privacy and moderation concerns. This is directly relevant to Inspot's post/forum feature design.

[5] *"Event-Based Systems in Mobile Applications: A Review"* (2023) by Nishita Rao et al. analyzes how mobile apps incorporate event management and participation systems. It provides a taxonomy of event types, RSVP mechanisms, and real-time notifications. The insights from this paper inform the event feature implementation in Inspot.

[6] *"Enhancing User Engagement through Social-Music Integration"* (2019) by Kevin Xu explores how merging music with social interaction improves user retention and experience. It studies platforms like SoundCloud and TikTok, showing how social influence shapes music discovery. These findings align with Inspot's core idea of combining music streaming with social sharing.

[7] *"Design Patterns for Mobile Music Applications"* (2020) by Laura Bentz et al. introduces UX and UI patterns commonly used in music apps, such as swipe gestures, floating action buttons, and persistent player bars. This provides design guidance for Inspot's music interface.

[8] *"Real-time Messaging Using Firebase Cloud Messaging"* (2021) by Dinesh Kumar demonstrates an implementation framework for chat features using Firebase Cloud Messaging and Firestore. The paper outlines challenges in synchronization and message history retrieval. The methodologies apply directly to Inspot's messaging module.

[9] *"Privacy and Security Concerns in Social Networking Mobile Apps"* (2022) by Reema Shaikh and Mohan Das highlights common vulnerabilities in mobile social apps, especially regarding data sharing, authentication, and third-party libraries. It

provides recommendations like using secure authentication flows (e.g., Firebase Auth), relevant to Inspot's login/signup system.

[10] *"User-Centered Evaluation of a Hybrid Music Social App"* (2023) by Elina Petrova evaluates a prototype music-social hybrid app via user studies. Results indicate high user satisfaction when music and social components are tightly integrated with smooth transitions. This validates the Inspot app's goal of merging both domains into a cohesive experience.

[11] *"A Survey on Real-Time Chat Applications and Frameworks"* (2021) by A. Mehta and R. Reddy reviews various technologies used to build real-time chat systems, including WebSockets, Firebase, and MQTT. It emphasizes the importance of low-latency, scalable infrastructure and efficient data syncing. The paper supports the technical choices in Inspot's messaging system using Firebase for fast, real-time interactions between users.

[12] *"Music Meets Messaging: A Study on Context-Aware Sharing in Music Apps"* (2022) by Nina Hoffmann and Carlo Jensen investigates how users share music within messaging platforms and what contextual factors influence sharing behavior. The study shows that users prefer integrated experiences, where music tracks can be shared, previewed, and discussed without leaving the app. This directly aligns with Inspot's aim to embed music sharing within its messaging interface.

[13] *"Towards Social Music Applications: Design Considerations and Case Studies"* (2020) by Priya Srivastava and L. Tan explores the architecture and user needs in social music apps like Smule and Spotify Group Sessions. It highlights the value of synchronous interaction—listening together, messaging about the track in real-time—which informs Inspot's concept of blending chat with live music sessions

or playlist sharing.

[14] *"Design and Evaluation of a Messaging Feature in Music-Centric Mobile Apps"* (2023) by Ethan Moore presents a user study on implementing in-app chat features tailored for music lovers. It suggests that users enjoy not only discussing tracks but also sending audio snippets and recommendations via chat. These insights influence how Inspot integrates music-related content directly into its chat UI.

[15] *"Enhancing Emotional Connection through Music and Messaging Integration"* (2021) by Yu-Hsuan Chang and Ming Li analyzes how music influences emotional bonding in digital communication. The research shows that apps combining music and messaging lead to increased user attachment and app retention. This psychological insight strengthens the foundational idea of Inspot as a socially engaging and emotionally resonant music experience.

CHAPTER 3

3. PROPOSED SYSTEM

3.1 GENERAL

The proposed system for *Inspot* focuses on creating a seamless integration between music streaming and social media interactions. The system is designed to be intuitive and user-friendly, providing users with an easy-to-navigate platform to discover, share, and enjoy music. By incorporating social media features like posts, forums, messaging, and events, *Inspot* enables users to interact with others while enjoying their favorite music. The goal is to create a platform where music lovers can not only listen to music but also connect and share their passion with others in a social setting.

3.2 SYSTEM ARCHITECTURE DIAGRAM

The system architecture for *Inspot* is designed to efficiently handle both music streaming and social interaction modules. The architecture is structured to ensure smooth performance, even with a large number of concurrent users. It includes Firebase for authentication, real-time messaging, and other essential services. The music streaming module is connected to a music API, allowing users to browse and stream music seamlessly. The system architecture diagram will illustrate how these components interact with each other to deliver the desired functionality.

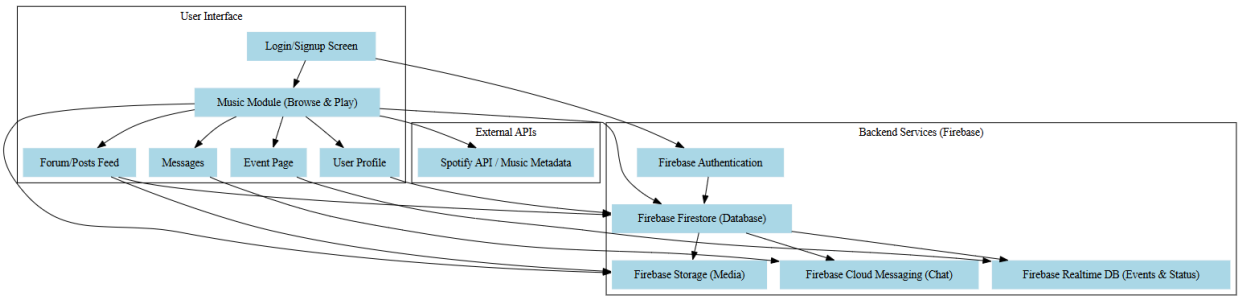


Fig 3.1: System Architecture

3.3 DEVELOPMENTAL ENVIRONMENT

3.3.1 HARDWARE REQUIREMENT.

The hardware requirements for *Inspot* focus on ensuring the app runs smoothly on Android devices. The minimum specifications include a device with at least 2GB of RAM and a multi-core processor. Additionally, devices should support Android version 8.0 (Oreo) or higher to ensure compatibility with the latest features and functionalities of the app.

Table 3.1 Hardware Requirements

COMPONENTS	SPECIFICATION
PROCESSOR	Intel Core i7 / AMD Ryzen 7 or higher
RAM	Minimum 16GB DDR4
POWER SUPPLY	+5V power supply

3.3.2 SOFTWARE REQUIREMENTS

The software environment required for the development of *Inspot* includes Android Studio as the primary integrated development environment (IDE). The app is built using Kotlin as the programming language. Firebase is used for authentication, real-time messaging, and data storage, while Picasso is used for image loading. Additionally, APIs for music streaming will be integrated to provide users with access to a wide range of music content.

Table 3.2 Software Requirements

COMPONENTS	SPECIFICATION
Operating System	Windows 10 or higher with GPU support
Database	Firebase
Programming Language	Android Studio

3.3 DESIGN OF THE ENTIRE SYSTEM

The design of the *Inspot* system focuses on both functional and non-functional aspects. The functional design ensures that the app provides users with an intuitive and engaging experience while browsing music, interacting with posts, and participating in events. Non-functional design considerations include the app's performance, scalability, and responsiveness. The user interface (UI) is designed to be simple yet engaging, with easy navigation between music, posts, and social interactions. A well-structured database will store user data, music content, posts, and event information.

3.3.1 ACTIVITY DIAGRAM

An activity diagram for *Inspot* will visually represent the flow of activities within the app. This will include the steps a user takes to log in, browse music, post content, message other users, and participate in events. The diagram will provide a clear overview of the user journey within the app, illustrating how different actions are connected and what happens at each step.

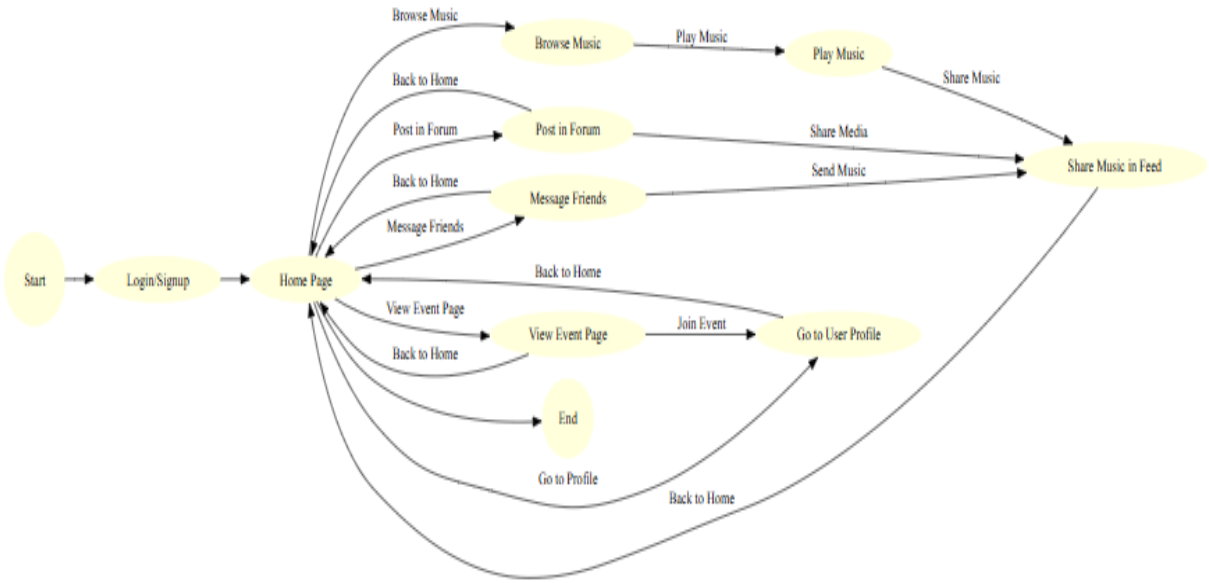


Fig 3.2: Activity Diagram

3.4 STATISTICAL ANALYSIS

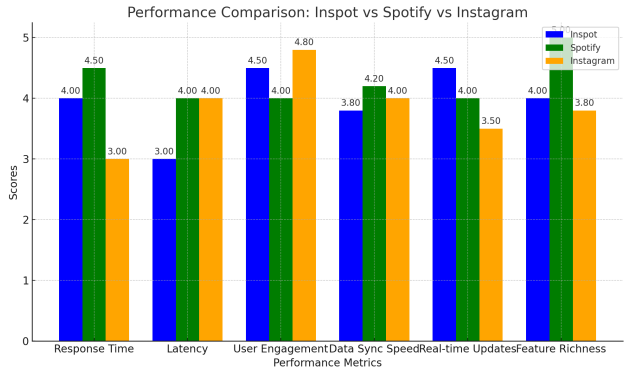
The statistical analysis section will focus on the data collected from user interactions within the app. This will include user activity patterns, such as the frequency of music streaming, engagement with posts, and participation in events. Firebase Analytics will be used to gather this data, which will help to understand user preferences and behavior. The analysis will provide valuable insights for further improving the app and tailoring it to better meet user needs.

Fig 3.4 provides a statistical summary comparing INSPOT with conventional apps.

Table 3.3 Comparison of features

Features	Spotify	Instagram	Inspot
Primary Focus	Music Streaming	Social Media	Music streaming + Social Media Integration
Content Type	Music(songs,playlist, albums)	Photos,videos,stories, posts	Music + User generated content
Social features	Collaborative playlists, sharing songs	Posting and live streaming	Forum style posts, music based events, real-time messaging
Community Engagement	Limited to comments on songs,playlists	High engagement with posts, stories and DMs	High engagement with posts, events, forums, and messaging
Monetization	Subscription based	Ads and sponsored content	Potential for ads, premium subscription and event based revenue
Target Audience	Music lovers, casual listeners	Social media users, content creators	Music lovers and social media users

Fig 3.3 : Comparison Graph



CHAPTER 4

4.MODULE DESCRIPTION

This section describes the architecture, key components, and technical implementation of the *Inspot* mobile application. It includes detailed insights into system structure, data handling, user interface design, performance optimization, and deployment strategies. The app is built using Android Studio, Kotlin, Firebase, and supporting APIs, combining music streaming and social engagement features into a single seamless platform.

4.1 SYSTEM ARCHITECTURE

The high-level system architecture of Inspot is designed to ensure modularity, scalability, and real-time responsiveness. The app architecture follows the MVVM (Model-View-ViewModel) pattern, which separates the user interface logic from the core business logic. Firebase Authentication handles secure user login and registration, ensuring smooth onboarding. The Firestore database is used to manage dynamic content such as user profiles, posts, forums, event details, and chat messages. Firebase Cloud Storage is employed to store multimedia files including profile pictures and post images. Music content is fetched through an integrated music API that allows users to browse by genres, stream tracks, and manage playlists.

4.1.1 HIGH-LEVEL SYSTEM OVERVIEW

The high-level architecture of *Inspot* is designed to integrate two major domains: music streaming and social networking. The application follows a modular approach with separate layers for user interface, data handling, backend communication, and authentication. The app uses Firebase as its backend for authentication, real-time database, messaging, and cloud storage. The music streaming functionality is powered

by APIs that deliver categorized song lists and audio files. These modules are loosely coupled to ensure flexibility and scalability of features.

4.1.2 USER INTERACTION DIAGRAM

The user interaction flow in *Inspot* is designed to be intuitive and efficient. Upon launching the app, users are directed to login or signup using Firebase Authentication. Once authenticated, users can explore music, create or view posts, join forums, send messages, and participate in events. Each feature is accessed via a bottom navigation bar, ensuring ease of access across modules. The interaction diagram outlines pathways such as login → homepage → explore music/post → play/share → interact via message or events.

4.1.3 BACKEND INFRASTRUCTURE

Inspot utilizes Firebase's suite of services for its backend operations. Firebase Authentication manages user credentials securely, while Firestore (Cloud Firestore) handles the storage of user profiles, posts, messages, and event data in real time. Firebase Cloud Storage is used to store images and media files for posts. For music, an external music API is integrated, and the audio files are either streamed or cached temporarily for user playback. The backend ensures scalability, real-time sync, and minimal latency for social interactions.

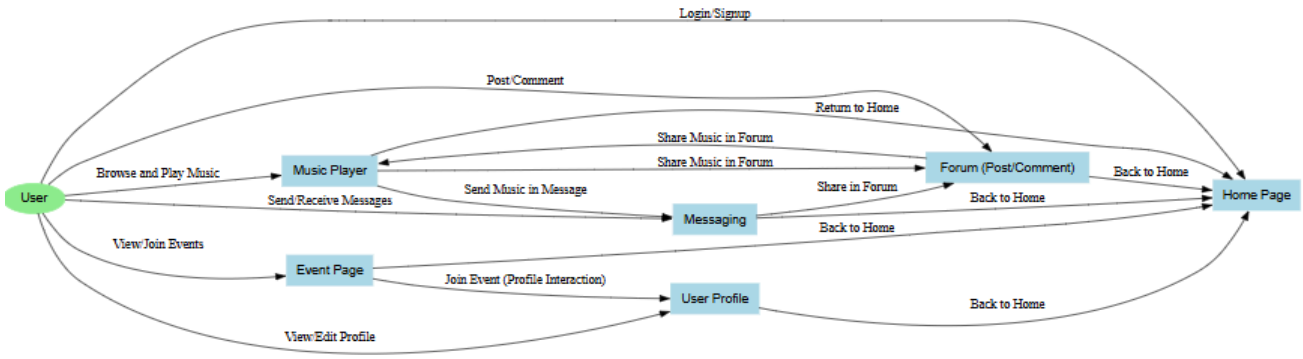


Fig 4.1 USER INTERACTION DIAGRAM

4.2 DATA HANDLING

4.2.1 DATA MANAGEMENT MODULE

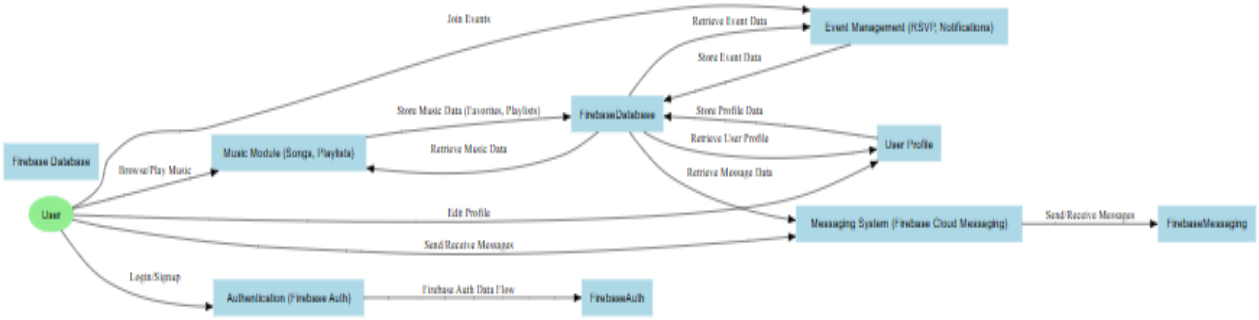


Fig 4.2 DATA MANAGEMENT MODULE DIAGRAM

Data in *Inspot* is primarily user-generated and includes profile information, posts, event details, and messages. The data is ingested through various input forms and managed using Firebase Firestore. The system ensures that data is stored in structured collections and documents, making it easy to retrieve and update in real time. Users can also upload images and audio content, which are stored in Firebase Storage and linked through Firestore.

4.2.2. USER PREFERENCES & RECOMMENDATION LOGIC

To enhance personalization, *Inspot* keeps track of user interactions, such as liked songs, followed genres, or frequently visited forums. Based on these interactions, simple rule-based filtering techniques are used to suggest songs or events. Future plans include integrating ML-based recommendation engines for music and forum topics, using user activity vectors.

4.2.3 SEARCH AND FILTERING

The app supports keyword-based search functionality across songs, posts, and users. This is implemented through Firestore queries and local filtering based on genres or tags. For an enhanced experience, the search system is optimized to rank recent and popular content higher. Posts and events are also filterable based on categories like trending, latest, or genre-based tags.

4.3 MEDIA & API INTEGRATION

4.3.1 MODEL SELECTION & INTEGRATION

Inspot integrates a third-party music API to fetch song metadata, audio links, and genre classifications. These APIs are accessed via HTTP requests, and the responses are parsed to display song lists dynamically. The integration allows users to stream music without needing to download it, saving device storage and ensuring legal content use.

4.3.2 STREAM HANDLING & MEDIA PLAYBACK

Media playback is handled using Android's MediaPlayer providing smooth and controllable streaming experiences. Playback state is maintained across activities, allowing background play. Buffering strategies and error handling are included to handle network instability, ensuring uninterrupted music streaming.

4.4 USER INTERFACE & EXPERIENCE

4.4.1 ANDROID UI IMPLEMENTATION

The front-end of *Inspot* is developed in Kotlin using Android Studio. The user interface is designed with Material Design principles for clarity, usability, and

responsiveness. RecyclerViews are used for listing posts, songs, and messages, while Fragments allow modular navigation. The app's theme combines vibrant musical tones with social-friendly layouts to enhance engagement.

4.4.2 NAVIGATION & SESSION FLOW

Session state management ensures that users return to where they left off, even after app minimization. Firebase keeps users authenticated, and shared preferences/cache are used to store temporary states like current song, last visited page, or active chat. Navigation between modules (home, explore, events, messages) is handled using a bottom navigation bar and ViewModel architecture.

4.5 PERFORMANCE & OPTIMIZATION

4.5.1 App Speed & Media Efficiency

To ensure optimal performance, *Inspot* uses lazy loading of images and deferred initialization of Firebase queries. Media playback is optimized with caching and efficient buffer control. Firebase offline persistence is enabled to improve speed and reduce data usage, especially for returning users.

4.5.2 User Engagement & Error Tracking

Analytics tools like Firebase Analytics and Crashlytics are integrated to monitor usage patterns and identify bugs or crashes. This helps track features users engage with most (e.g., music, forums) and optimize those modules. Regular feedback collection and testing contribute to improving contextual accuracy and user satisfaction.

4.6 DEPLOYMENT & STORAGE

4.6.1 LOCAL VS CLOUD-BASED DEPLOYMENT

The app is developed locally using Android Studio and deployed through the Google Play Console. Firebase, being a cloud-based backend, handles real-time operations and ensures smooth multi-user scalability. During development, emulators and local builds are used for testing.

4.6.2 FIREBASE STORAGE & DATABASE MANAGEMENT

Firebase Storage is used to manage uploaded images and media files. The Firestore database is structured using subcollections for scalability—user → posts, user → messages, event → participants. Indexing is applied to optimize query speed, and rules are defined for data security and access control.

CHAPTER 5

IMPLEMENTATION AND RESULTS

5.1 IMPLEMENTATION

The implementation of *Inspot* was carried out using Android Studio with Kotlin as the core programming language, leveraging the MVVM (Model-View-ViewModel) architectural pattern to ensure a clean separation between the UI and business logic. Firebase services formed the backbone of the backend infrastructure. Firebase Authentication was integrated to support secure user sign-up, login, and session management. Firestore Database was utilized to store and retrieve real-time user data such as posts, comments, event details, and messages. Firebase Storage handled multimedia content including profile pictures and audio files. For the music module, a third-party music API was integrated to fetch songs based on genre, popularity, and recent trends, which were then displayed in a RecyclerView with smooth playback features. The post and forum modules were inspired by Instagram's feed and allowed users to create text/image-based posts that other users could like and comment on. Real-time messaging functionality was implemented using Firebase Realtime Database to enable seamless peer-to-peer communication. Events and meetups were managed through a dynamic events module, allowing users to view, register, and interact within music-related communities. The frontend was styled with Material Design principles to ensure a modern and responsive UI, while Room Database was optionally used for local caching to improve app performance. Throughout the development cycle, rigorous testing was performed on both emulators and real Android devices to ensure cross-device compatibility, minimal crashes, and fluid user experience.

5.2 OUTPUT SCREENSHOTS

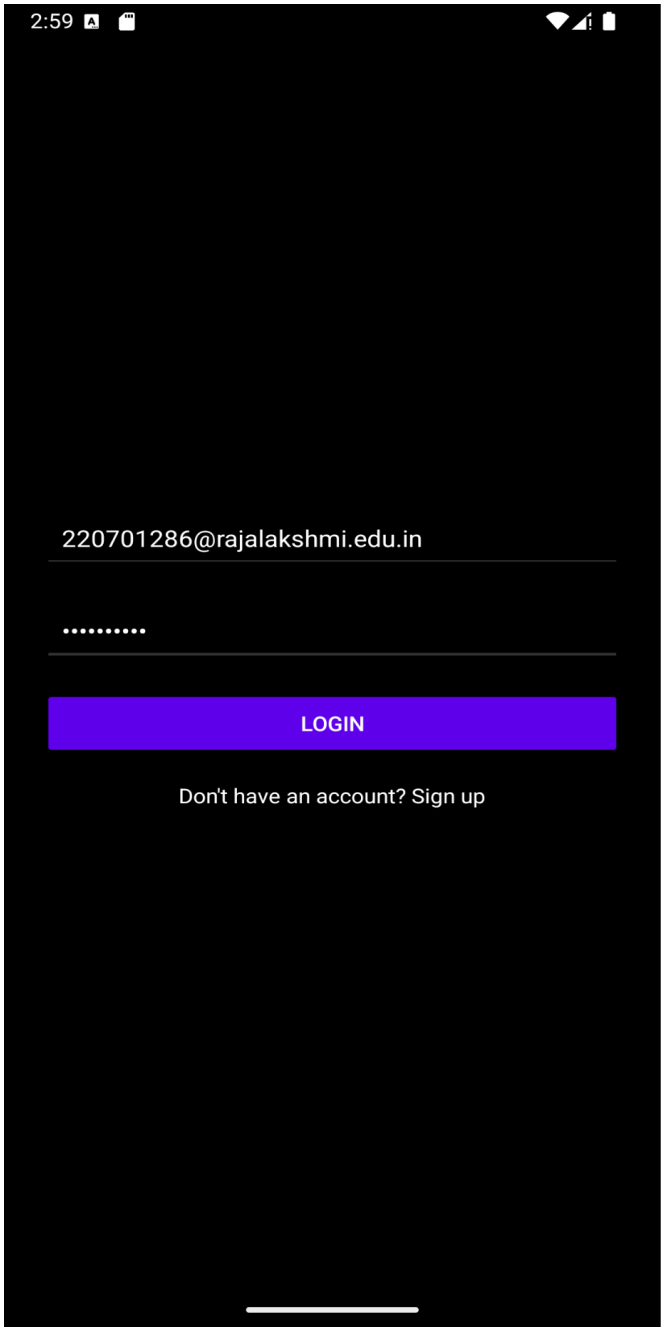


Fig 5.1 Login Page

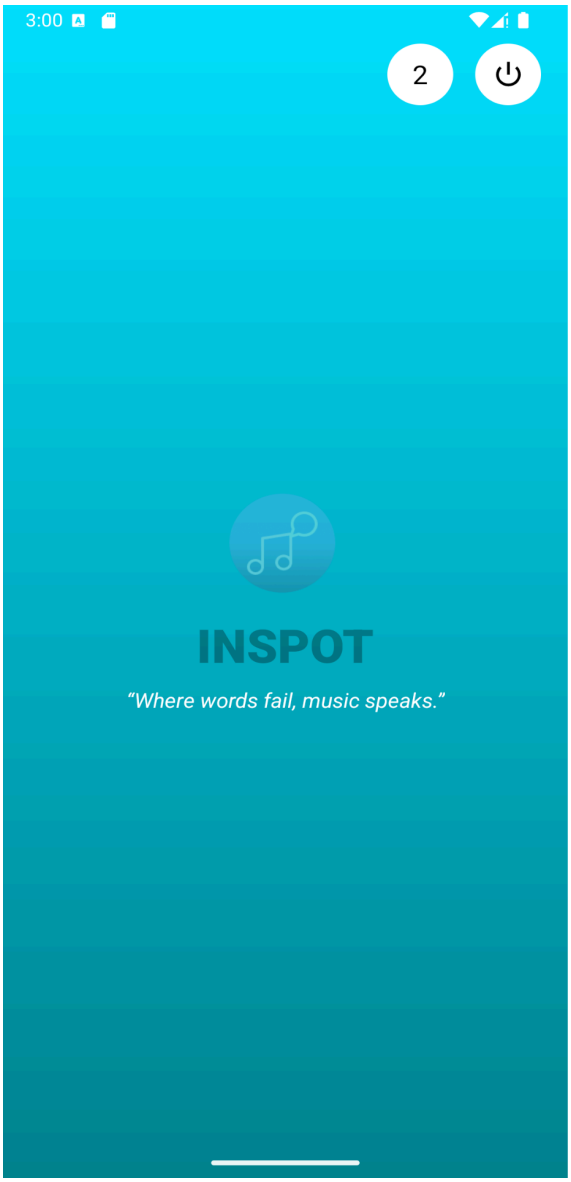


Fig 5.2 Home Page

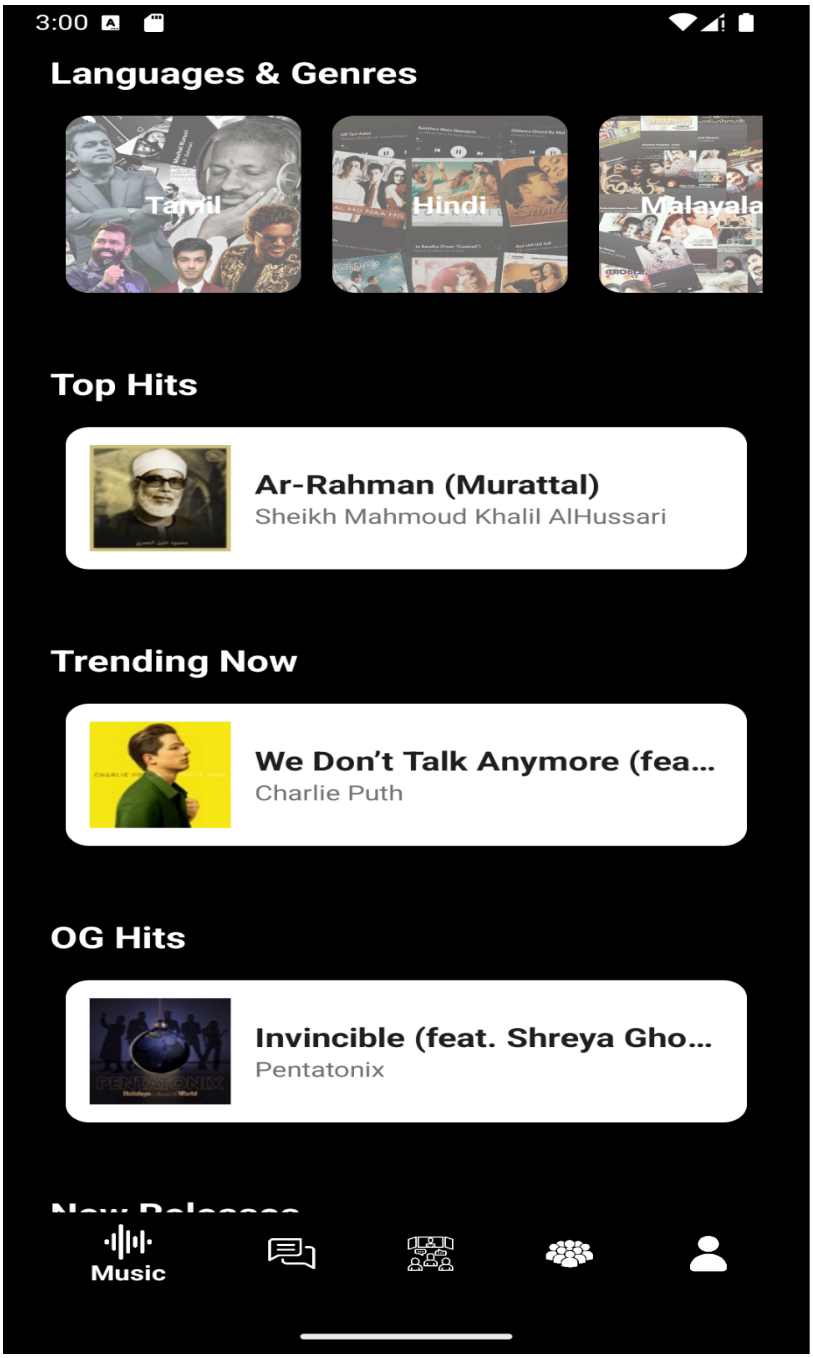


Fig 5.3 Music Page

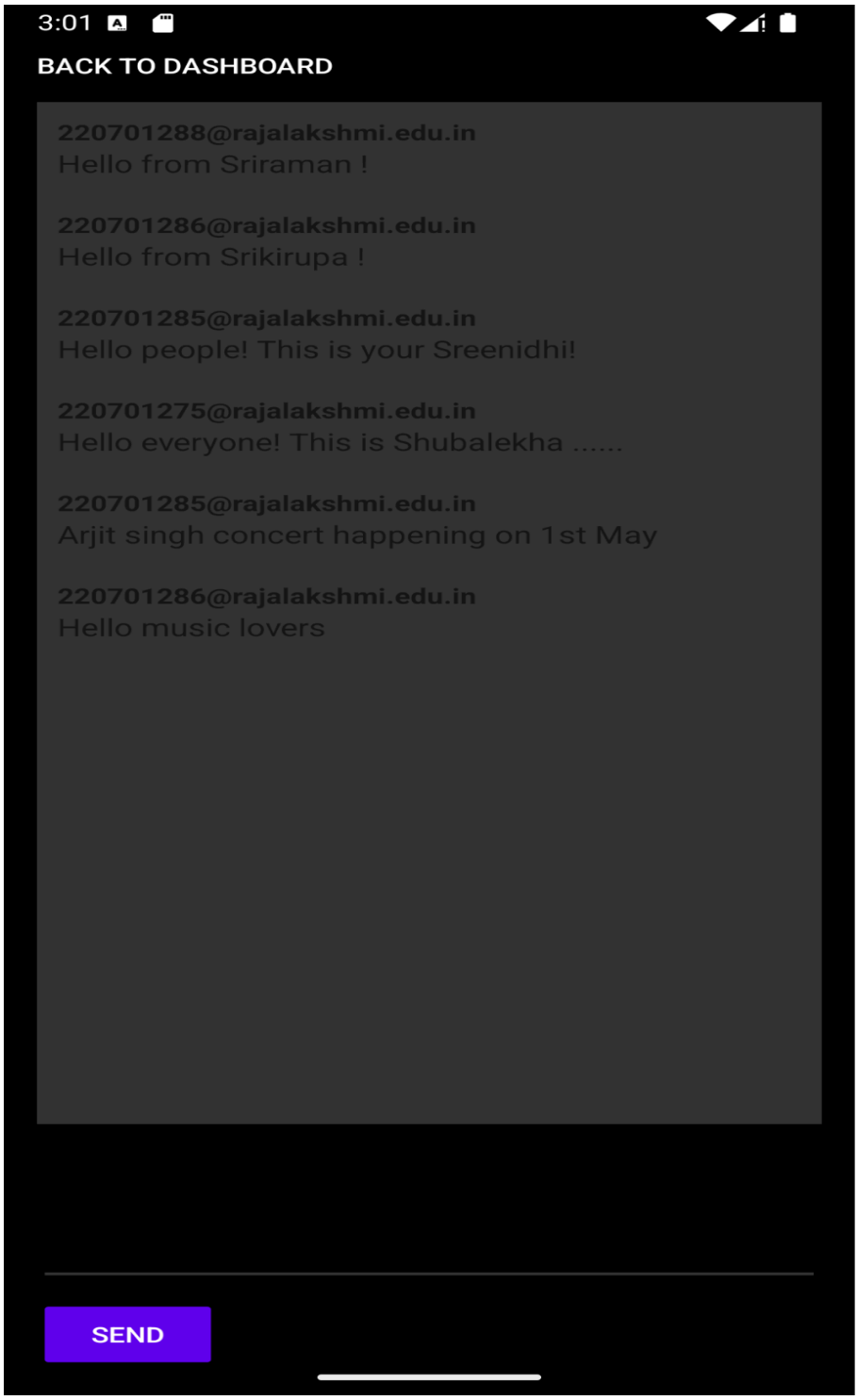


Fig 5.4 Message Page

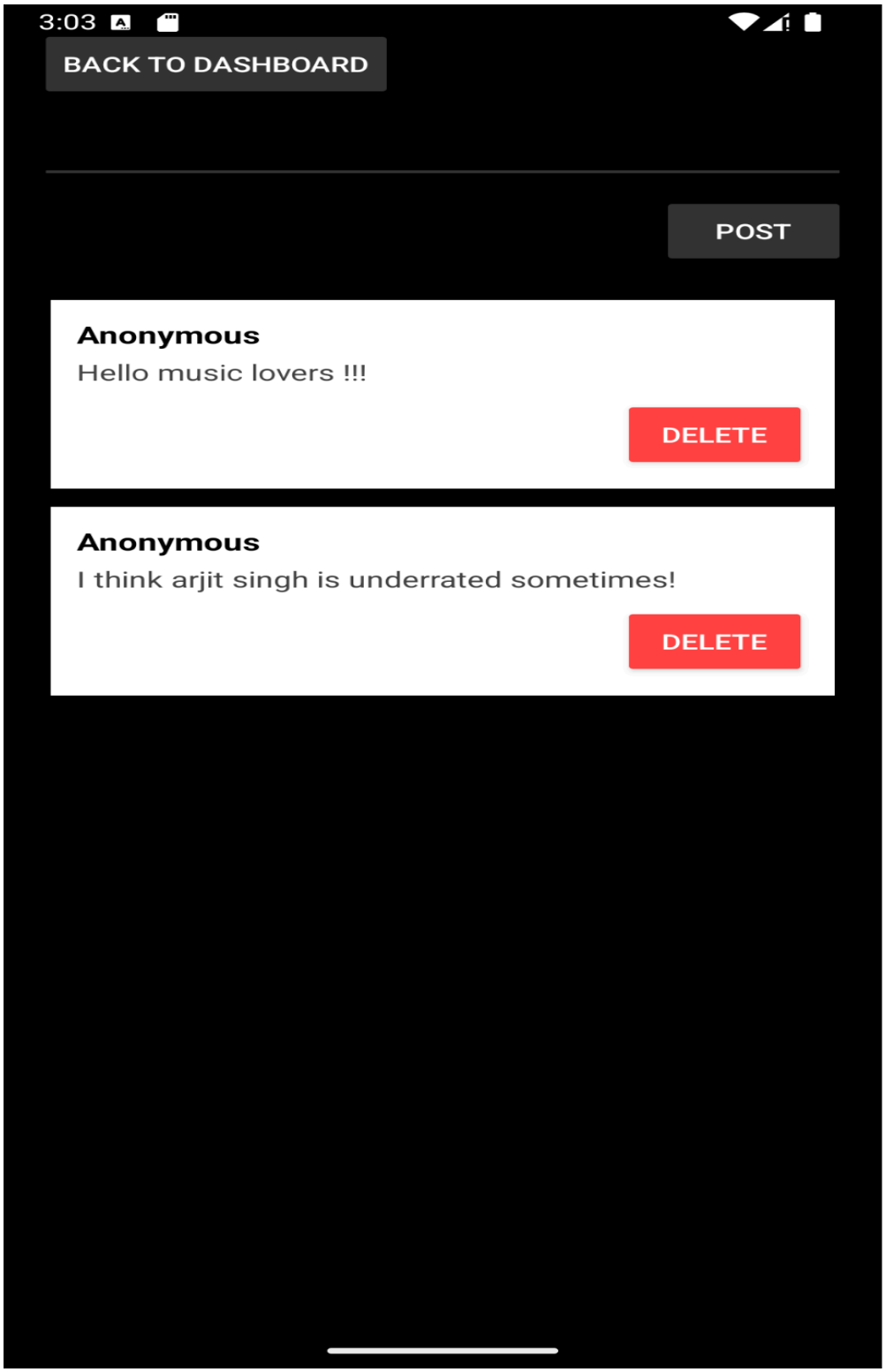


Fig 5.5 Post Page

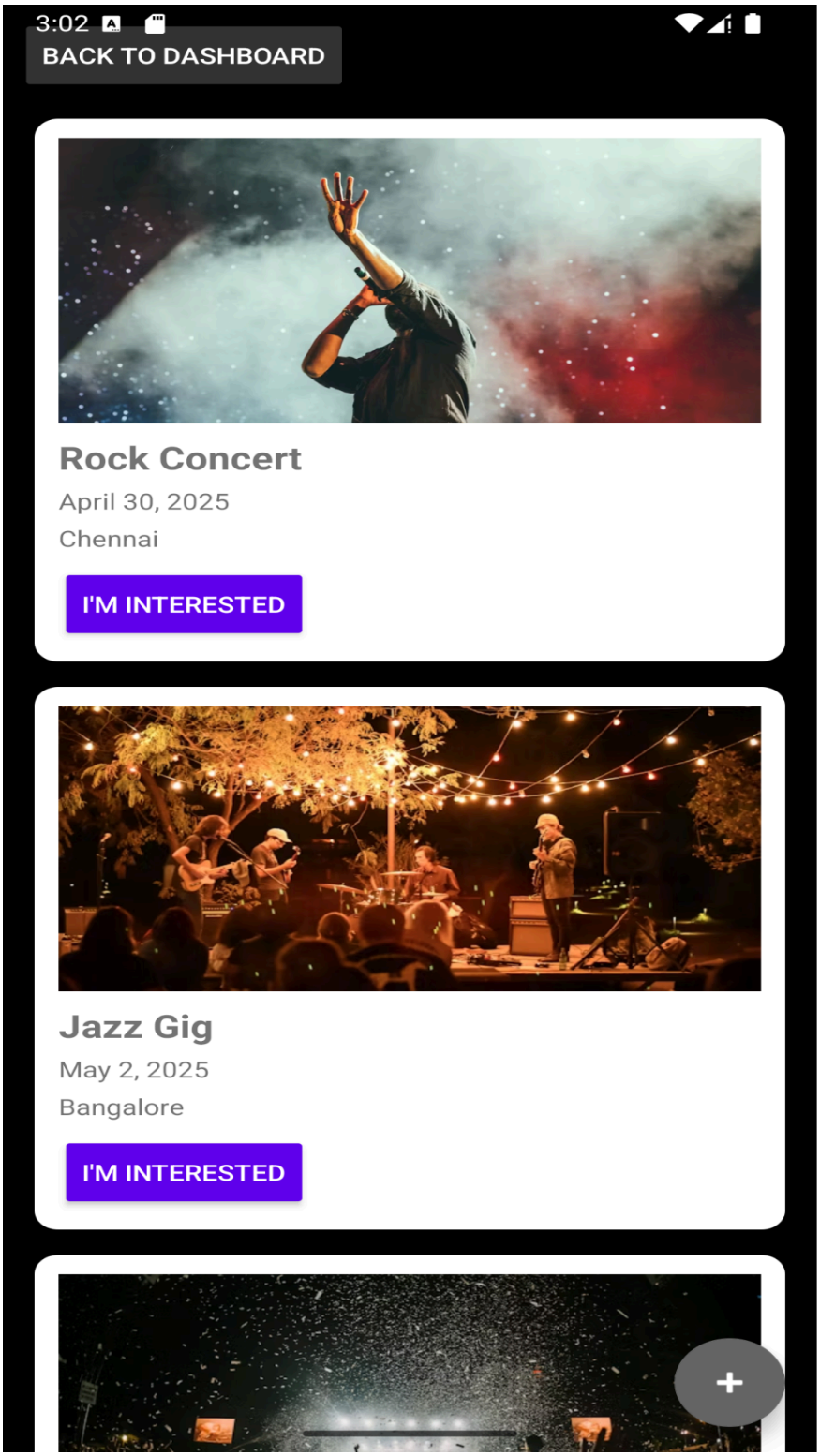


Fig 5.6 Event Page

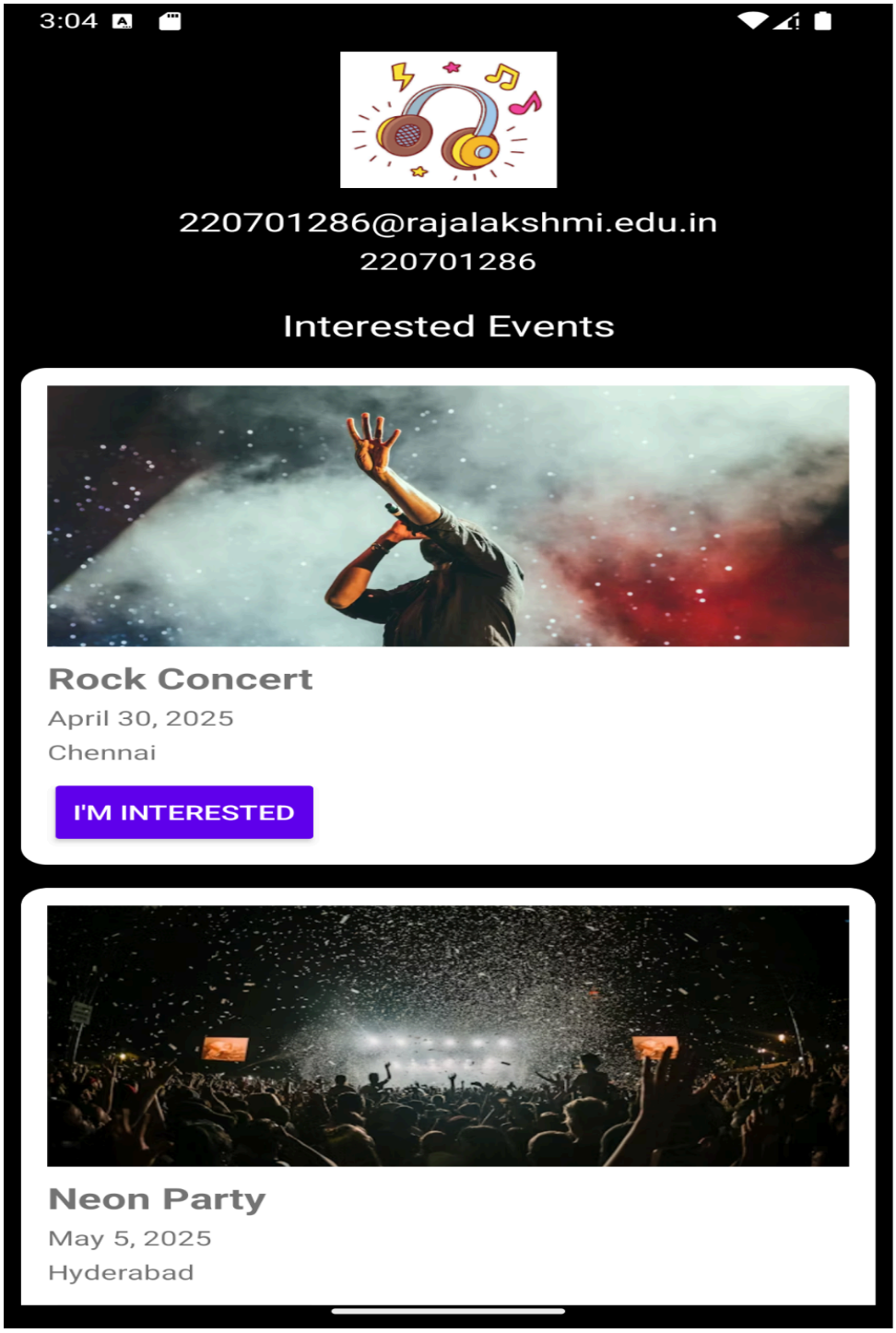


Fig 5.7 Profile Page

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

The *Inspot* application successfully combines the engaging elements of music streaming and social interaction within a unified mobile platform. By merging the core functionalities of popular applications like Spotify and Instagram, Inspot delivers a fresh and immersive experience where users can not only stream and explore music but also share their emotions, reviews, and discussions with like-minded individuals. The integration of Firebase for authentication, real-time database operations, and cloud storage allowed for a robust and scalable backend. The adoption of the MVVM architecture and the use of third-party APIs for music streaming ensured that the application remained modular, maintainable, and user-friendly. Additionally, real-time messaging, event management, and community-building features have enhanced user engagement and created a socially connected environment around music.

6.2 FUTURE ENHANCEMENT

AI-based recommendation systems could be integrated to suggest songs, playlists, and even friends based on user preferences and activity. Voice command features and accessibility improvements can broaden the app's usability. Offline music caching and background play would offer more convenience to users with limited internet access. Furthermore, introducing monetization options such as artist sponsorships, premium features, or ad-free experiences can help sustain the application. Finally, scalability towards cross-platform compatibility using technologies like Flutter or React Native could allow *Inspot* to reach a wider audience beyond Android. *Inspot* sets a strong foundation for a next-generation music-social experience and holds significant potential for future innovation and growth.

REFERENCES

1. Multi-Modal Interaction in Music Sharing Applications – Lina Zhang and Erik Petersen (2022)
2. Music Recommender Systems: Challenges and Opportunities – Markus Schedl et al. (2020)
3. A Comparative Study of Firebase and Other Cloud Backend Services – K. Anusha and P. Aravind (2021)
4. Integrating Social Media Features into Mobile Applications – Lina Zhang and Tomas Horak (2022)
5. Event-Based Systems in Mobile Applications: A Review – Nishita Rao et al. (2023)
6. Enhancing User Engagement through Social-Music Integration – Kevin Xu (2019)
7. Design Patterns for Mobile Music Applications – Laura Bentz et al. (2020)
8. Real-time Messaging Using Firebase Cloud Messaging – Dinesh Kumar (2021)
9. Privacy and Security Concerns in Social Networking Mobile Apps – Reema Shaikh and Mohan Das (2022)
10. User-Centered Evaluation of a Hybrid Music Social App – Elina Petrova (2023)
11. A Survey on Real-Time Chat Applications and Frameworks – A. Mehta and R. Reddy (2021)
12. Music Meets Messaging: A Study on Context-Aware Sharing in Music Apps – Nina Hoffmann and Carlo Jensen (2022)

13. Towards Social Music Applications: Design Considerations and Case Studies – Priya Srivastava and L. Tan (2020)
14. Design and Evaluation of a Messaging Feature in Music-Centric Mobile Apps – Ethan Moore (2023)
15. Enhancing Emotional Connection through Music and Messaging Integration – Yu-Hsuan Chang and Ming Li (2021)