Melody: A Flutter-Based Music Recommendation App with Firebase and MongoDB Integration

For

MAD Lab mini Project

Third Year (Semester-VI) Bachelors in Information Technology, Autonomous Program

By

Student Name - Anshi Tiwari   Class_Roll no - 56

Supervisor

Mrs. Dipti Karani

Department of Information Technology
Vivekanand Education Society's Institute of Technology
An Autonomous Institute Affiliated to University of Mumbai
2024-2025

Certificate

This is to certify that Anshi Tiwari has completed the project report on the **Melody: A Flutter-Based Music Recommendation App with Firebase and MongoDB Integration** topic satisfactorily in partial fulfilment of the requirements for the award of Mini Project in MAD lab of third Year, (Semester-VI) in Information Technology under the guidance of Mrs. Dipti Karani during the year 2024-2025 as prescribed by An Autonomous Institute Affiliated to University of Mumbai

Supervisor/ Examiner

Declaration

We declare that this written submission represents our ideas in our words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misinterpreted or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Anshi Tiwari-56

Date:

References

Table of Contents

Abstract

Melody is a personalized music recommendation application developed using the Flutter framework, aiming to enhance user experience through real-time suggestions based on individual preferences. By integrating Firebase for secure authentication and MongoDB for flexible data storage, the app ensures smooth performance and scalability. The app supports both email/password and Google sign-in methods, enabling seamless access while maintaining user data security.

The recommendation engine leverages user input such as genre preferences and listening habits to provide tailored song suggestions. A clean and intuitive interface, designed with Material Design principles, allows users to explore trending music, search for specific tracks, and manage their profile details effortlessly. Features like favorite song tracking and genre insights help create a more engaging and interactive musical journey.

In addition to its current capabilities, Melody is built with future scalability in mind. Planned features include AI-based mood detection for smarter recommendations, playlist creation and sharing, and a dark mode for better user comfort. This project highlights the potential of combining modern cross-platform development tools with real-time databases to deliver dynamic, user-centric digital experiences.

List of Figures

Chapter 1: Introduction

1.1 Introduction

Music is deeply personal, and in the digital age, users expect their music apps to understand their preferences and offer recommendations that resonate with their tastes. Melody is a cross-platform mobile application developed using Flutter that addresses this need by delivering personalized music suggestions. It combines a clean user interface with real-time functionality, ensuring users not only find music they enjoy but also engage with an app that feels responsive and intuitive.

At the core of Melody lies Firebase Authentication, which secures user data and streamlines the sign-up/login process with support for email/password and Google authentication. MongoDB, used for backend storage, allows dynamic handling of user data such as favorite genres, top songs, and listening history. Together, these technologies ensure that each user's experience is customized and updated in real time, laying the foundation for a smart recommendation system.

Melody emphasizes both functionality and user experience. The app features a minimalistic UI, a pastel-themed aesthetic, and interactive elements like icons and animations. Users can browse song suggestions, search by title or artist, and manage their profiles effortlessly. Designed with future enhancements in mind, Melody stands as a scalable, modern solution in the space of music recommendation apps.

1.2 Objectives

- ○ **Develop a personalized music recommendation system:**
  Design an app that understands user preferences based on genres and listening history to provide relevant song suggestions. The aim is to make the discovery process intuitive, enjoyable, and genuinely tailored to each user.
- ○ **Implement secure user authentication:**
  Use Firebase Authentication to manage user sign-up and login processes securely. This includes support for both email/password and Google login, ensuring ease of access without compromising data privacy.
- ○ **Store and manage user data effectively:**
  Integrate MongoDB as the backend database to handle dynamic user preferences, favorite songs, and genre selections. The goal is to achieve efficient data retrieval and real-time updates with minimal latency.
- ○ **Design an interactive and user-friendly interface:**
  Build a clean, minimal UI using Flutter and Material Design principles. Focus on

providing a visually appealing and easy-to-navigate experience that encourages users to explore and engage with the app.

- **Ensure scalability and future extensibility:**
  Structure the application to accommodate future enhancements like mood-based recommendations, playlist sharing, and dark mode. This future-readiness is key to maintaining relevance and user satisfaction over time.

1.3 Scope of the Project

The Melody app is designed to function as a lightweight yet powerful music recommendation platform with a focus on personalization, usability, and future scalability. It currently includes core features like secure login, profile management, and song recommendations based on user preferences. The app is built using technologies that allow cross-platform compatibility and efficient data handling, laying the groundwork for more advanced features to be added in the future.

Project Scope Includes:

- Personalized song recommendations based on selected genres and listening behavior.
- Secure user authentication using Firebase (email/password and Google Sign-In).
- MongoDB integration for storing user preferences and song data.
- Profile management features allowing updates to user info and preferences.
- A clean, responsive UI developed with Flutter and Material Design.

Chapter 2: Review of Literature

2.1 Literature Survey

Music recommendation systems have evolved significantly in recent years, employing various techniques to understand user preferences and deliver tailored content. Two notable papers offer insights into both traditional and advanced approaches in this domain.

Yading Song, Simon Dixon, and Marcus Pearce conducted a comprehensive survey covering different types of music recommendation models, including collaborative filtering, content-based, context-aware, and emotion-based systems. The paper discusses six prominent models and dives into the challenges of user experience, modeling techniques, and matching algorithms. It introduces a motivation-based framework grounded in human behavior and music psychology, offering a fresh direction for enhancing recommendation systems [1].

Markus Schedl's work explores the application of deep learning in music recommendation systems. The paper highlights how latent factor models and sequence learning improve recommendation quality, particularly in playlist generation and user engagement. It presents a review of various deep learning architectures, input types, recommendation strategies, and tasks, while also addressing implementation challenges in real-world systems [2].

2.2 Research Gap

- **Limited Personalization in Existing Apps:**
  Many mainstream music platforms offer recommendations, but they often rely heavily on generic trends or collaborative filtering without deeply analyzing individual user behavior and preferences.
- **Lack of Real-Time Adaptability:**
  Current systems may not update recommendations dynamically based on immediate user interactions, leading to static or delayed personalization that fails to reflect changing tastes.
- **Insufficient Integration of Emotional Context:**
  Few recommendation engines consider user mood or emotional state, which plays a significant role in music choice. Emotion-based recommendation is still an underexplored area in many apps.
- **Complexity and Scalability Challenges:**
  Traditional recommendation systems often struggle to balance recommendation quality with system performance, especially as user data scales and preferences diversify.

Chapter 3: Project Description

3.1 Problem Definition

In today's digital landscape, users are often overwhelmed by the abundance of music content available on streaming platforms, making it challenging to discover songs that truly match their preferences. Most existing systems rely on either generic popularity metrics or collaborative filtering, which may not fully reflect individual taste, mood, or listening habits. There is a clear need for a lightweight, cross-platform solution that delivers intelligent, real-time, and deeply personalized music recommendations while maintaining a smooth and secure user experience.

Key Problems Addressed:

- **Overwhelming Volume of Music Choices:**
  Users often struggle to find music that resonates with their current mood or taste due to the vast and unfiltered libraries of content on most platforms.
- **Inadequate Personalization Mechanisms:**
  Existing recommendation systems commonly lack depth in personalization, relying too much on surface-level data like trending charts or average user behavior, rather than individual listening patterns.
- **Poor Cross-Platform Experience:**
  Many music apps lack consistent performance across platforms, leading to broken UI, slower load times, or different features on Android and iOS. This inconsistency affects user satisfaction and accessibility.
- **Security and Privacy Concerns in Authentication:**
  Handling sensitive user data like login credentials and listening history requires secure, reliable authentication methods—something not all applications prioritize or implement effectively.
- **Limited User Control and Feedback Integration:**
  Users often cannot influence or fine-tune their recommendations manually, nor do most apps adapt quickly to new feedback or changes in taste, resulting in stagnant or repetitive suggestions.

3.2 Steps Involved

Building an app like Melody requires a clear roadmap that ensures each feature is implemented effectively, resulting in a seamless user experience. The process involves both frontend and backend development, integrating various services for secure authentication and real-time data management.

Steps Involved:

- **User Authentication Setup:**
  Integrate Firebase Authentication for managing secure sign-ups and log-ins, supporting both email/password and Google authentication methods.
- **Database Integration:**
  Set up MongoDB to store user data, including preferences, listening history, and favorite genres, ensuring dynamic and efficient data management.
- **Song Recommendation System Development:**
  Implement the recommendation engine to suggest songs based on user preferences, ensuring that it can learn and adapt to new input over time.
- **Frontend Design and Implementation:**
  Develop the mobile interface using Flutter, focusing on clean UI/UX design principles, and ensuring smooth functionality on both iOS and Android platforms.
- **Testing and Deployment:**
  Conduct thorough testing of the application to ensure that it works seamlessly across different devices and that all features are functioning as expected before deployment.

3.3 ER Diagram of Proposed Project

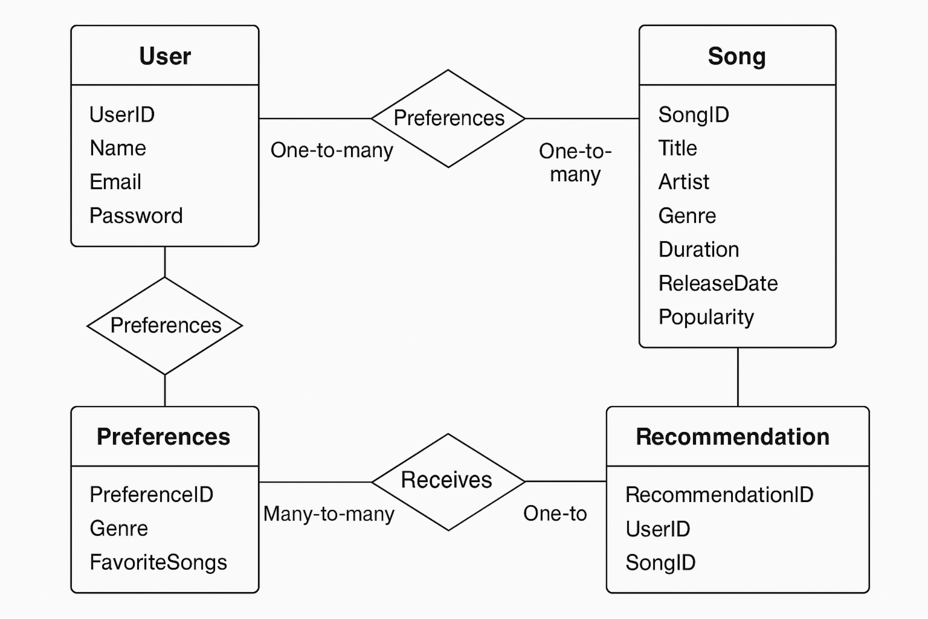


Fig 1: ER Diagram of Proposed Project

3.4 Technology Description

To create an engaging and dynamic music recommendation app, the selection of technologies is crucial for both performance and scalability. Melody leverages powerful tools and frameworks that enable smooth, real-time interactions between the user interface and backend, while also ensuring data security and ease of management.

Technologies Used:

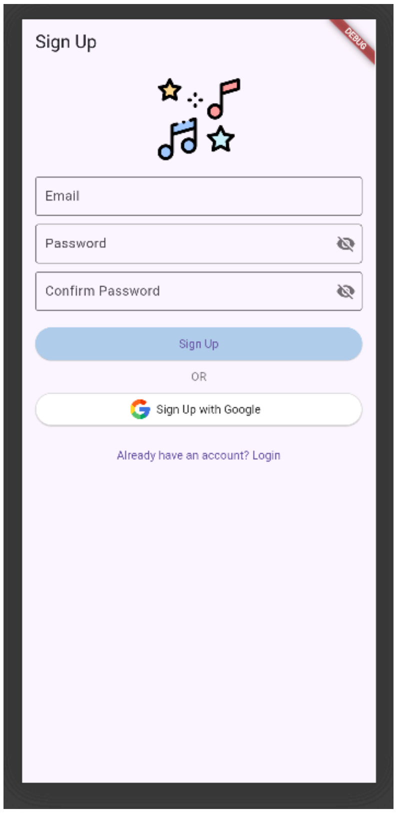
- **Flutter (Dart):**
  The cross-platform mobile development framework used to build the app's frontend. It allows for fast and efficient development with a consistent experience across both iOS and Android devices.
- **Firebase Authentication:**
  A reliable solution for managing user authentication, enabling secure logins via email/password or Google Sign-In, which enhances user experience while protecting sensitive data.
- **MongoDB:**
  A NoSQL database that stores and manages user data such as preferences, song history, and genre selections. Its flexible document-based structure ensures fast and scalable data retrieval.

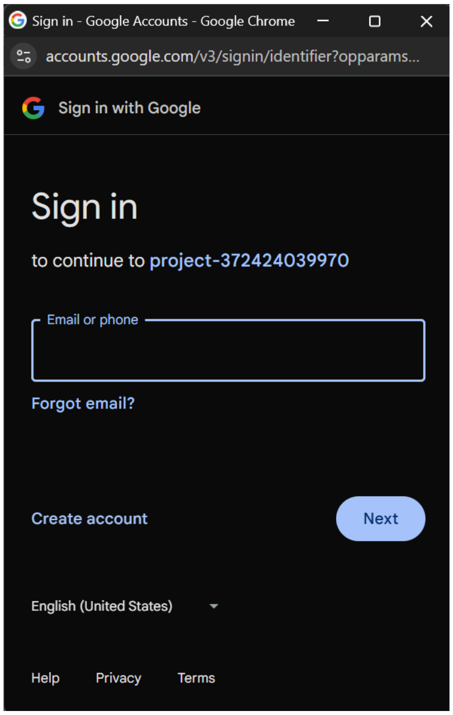
Chapter 4: Implementation

4.1 Application Implementation

The application implementation of Melody involves integrating the various technologies and tools to create a seamless experience for the user. The Flutter framework forms the foundation for the mobile interface, offering a responsive design for both iOS and Android platforms. Firebase Authentication ensures secure user login and registration, while MongoDB is utilized to store user preferences, songs, and genres in a scalable and efficient manner. The recommendation system leverages the stored user data to generate personalized music suggestions, ensuring a tailored experience. The app also includes features such as search functionality and profile management, all working in real-time to provide users with an interactive and user-friendly platform.

1. Account Creation and verification




Sign in - Google Accounts - Google Chrome

accounts.google.com/v3/signin/identifier?opparams...

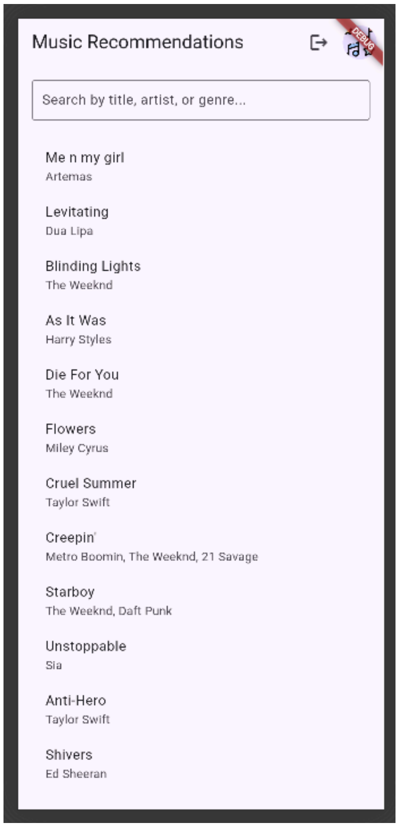
Sign in with Google

Sign in

to continue to project-372424039970

Fig 2: Signup Page **Fig 3: Signup using google account**

2. Music Recommendation and Profile Page


Music Recommendations

Search by title, artist, or genre...

Me n my girl
Artemas

Levitating
Dua Lipa

Blinding Lights
The Weeknd

As It Was
Harry Styles

Die For You
The Weeknd

Flowers
Miley Cyrus

Cruel Summer
Taylor Swift

Creepin'
Metro Boomin, The Weeknd, 21 Savage

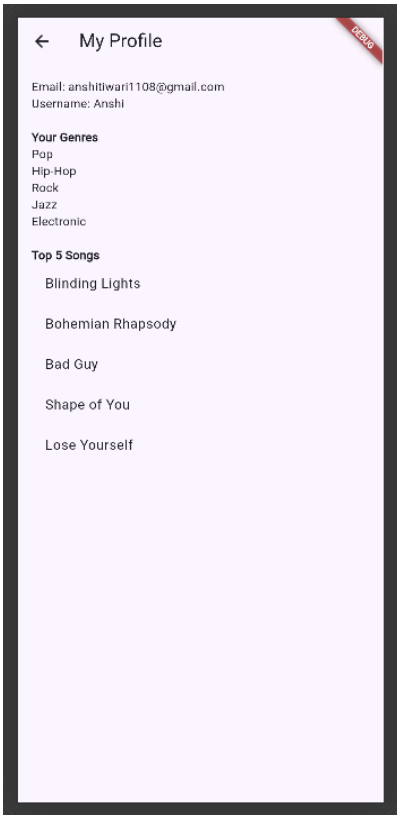
Starboy
The Weeknd, Daft Punk

Unstoppable
Sia

Anti-Hero
Taylor Swift

Shivers
Ed Sheeran

Fig 4: Music Recommendation Page

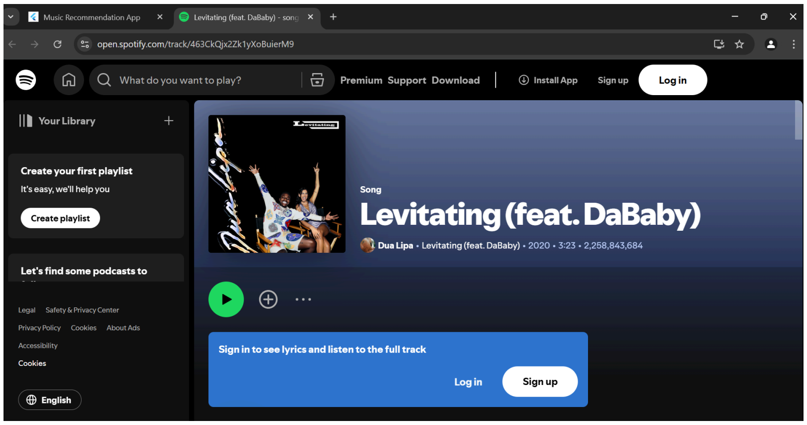

My Profile

Email: anshitiwari1108@gmail.com
Username: Anshi

Your Genres
Pop
Hip-Hop
Rock
Jazz
Electronic

Top 5 Songs
Blinding Lights
Bohemian Rhapsody
Bad Guy
Shape of You
Lose Yourself

Fig 5: Profile Page with Data



Create your first playlist
It's easy, we'll help you

Let's find some podcasts to

Song
Levitating (feat. DaBaby)
Dua Lipa • Levitating (feat. DaBaby) • 2020 • 3:23 • 2,258,843,684

Sign in to see lyrics and listen to the full track

Fig 6: Linking to Song’s Spotify page

4.2 Software Implementation (Flowchart/Algorithms)

To effectively depict the logic and flow of the application, a flowchart or algorithm is used to illustrate the sequence of operations. The software flowchart will visualize the user journey starting from authentication, through song recommendations, to profile management. It will demonstrate how the app processes user data, retrieves songs from the database, and presents tailored recommendations. Additionally, the algorithms driving the recommendation engine and data management will be detailed, showcasing how user inputs and preferences are incorporated to enhance the app's responsiveness. You can insert your **flowchart/algorithm diagram** here to visually represent the process.

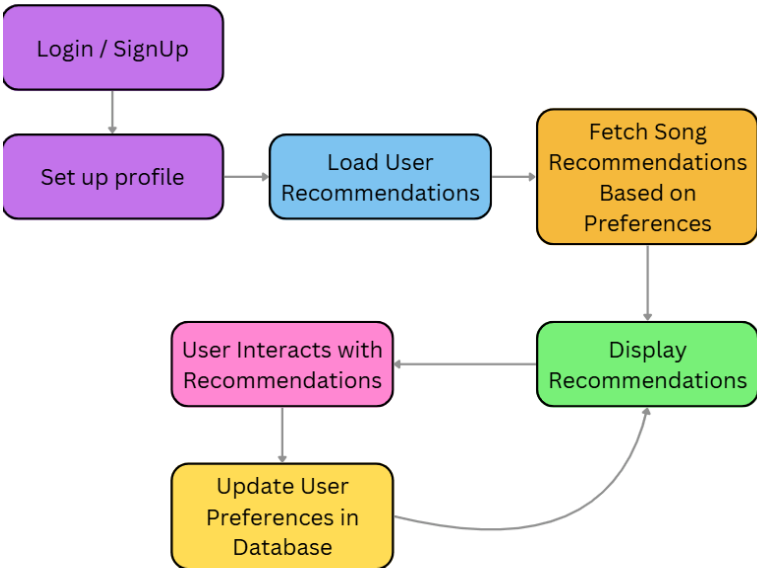

Login / SignUp
Set up profile
Load User Recommendations
Fetch Song Recommendations Based on Preferences
Display Recommendations
User Interacts with Recommendations
Update User Preferences in Database

Fig 7: Flowchart of the Application

Chapter 5: Results and Conclusion

5.1 Results

The Melody application has been successfully developed and integrated with all key features, including user authentication, personalized music recommendations, and real-time data synchronization. Upon testing, the app has demonstrated the ability to recommend music effectively based on user preferences, ensuring that users receive suggestions tailored to their individual taste. The user interface, designed using Flutter and Material Design, has proven to be intuitive and easy to navigate, providing users with a seamless experience. Firebase Authentication has provided a secure and efficient login process, while MongoDB has proven to be reliable in managing user data and ensuring scalability.

Additionally, the real-time synchronization between user actions (such as song preferences and profile updates) and the app's backend has functioned smoothly. The application was also tested across both iOS and Android platforms, ensuring consistent performance and functionality. User feedback indicated a high level of satisfaction with the personalized recommendations, confirming the effectiveness of the system. In terms of performance, the app has shown minimal lag and quick response times, even with large datasets, thanks to the optimized database structure and cloud functions handling backend processes.

5.2 Conclusion

Melody successfully integrates cutting-edge technologies such as Flutter, Firebase, and MongoDB to create a comprehensive music recommendation system that provides users with personalized and secure music discovery. By focusing on a user-friendly interface, smooth cross-platform functionality, and robust backend architecture, Melody offers a seamless and engaging experience for users. The integration of Firebase Authentication ensures secure logins, while MongoDB allows for efficient and scalable storage of user data. The recommendation system, based on users' music preferences and history, delivers tailored song suggestions that enhance the overall user experience.

Despite the progress made, the app still has significant potential for growth. The combination of real-time updates, personalized recommendations, and a responsive interface makes Melody a competitive addition to the music recommendation space. The positive results from the initial testing phase confirm that the app fulfills its intended goals of providing an intuitive, secure, and engaging music discovery experience. Future updates will focus on refining the user experience and expanding the app's capabilities to accommodate more advanced features.

5.3 Future Improvements

While Melody currently offers core features like personalized song recommendations, secure authentication, and profile management, there are several areas for future enhancement that can further improve user experience and functionality. Some of the future implementation ideas include:

- **AI-Based Mood Detection:**
  An advanced feature could be the integration of mood-based recommendations, where the app analyzes the user's current emotional state (possibly via text input or even audio analysis) and suggests music that aligns with their mood. This would enhance the personalized experience by adapting recommendations in real-time.
- **Social Integration:**
  Melody could integrate social media sharing features that allow users to share their favorite songs or playlists on platforms like Instagram or Facebook, encouraging further engagement and expanding the app's reach.
- **Dark Mode Support:**
  As users increasingly prefer dark modes in apps, implementing this feature will enhance the app's versatility, offering a comfortable experience for night-time usage, reducing eye strain.
- **Voice Recognition for Hands-Free Control:**
  Integrating voice recognition features could allow users to control the app without having to touch their device. Users could request songs, change tracks, or get recommendations simply by speaking, improving usability, especially when users are on the go.
- **Enhanced Recommendation Algorithms:**
  Future versions of the recommendation engine can be improved by incorporating more sophisticated algorithms like collaborative filtering, deep learning, and contextual awareness. This will allow the app to provide even more accurate and diverse recommendations based on subtle patterns in user behavior.
- **Offline Mode:**
  Users could enjoy the app even without an active internet connection by enabling offline mode. This would allow them to listen to previously recommended or downloaded tracks when network access is unavailable.

References

1. [1] Y. Song, S. Dixon, and M. Pearce, “A Survey of Music Recommendation Systems and Future Perspectives,” *Citeseerx*, [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e0080299afae01ad796060abcf602abff6024754

2. [2] M. Schedl, “Deep Learning in Music Recommendation Systems,” *Frontiers in Applied Mathematics and Statistics*, vol. 5, pp. 1–14, 2019. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fams.2019.00044/full