Business Requirements Document: Modern Royalty Management System for Artists

# Introduction:

Royalty Management System (RMS) is tailored specifically for artists. The RMS aims to streamline the management of royalties, providing artists with greater transparency, efficiency, and control over their earnings from various sources within the music industry. In the music industry, managing royalties for artists is a multifaceted task due to the diverse cultural landscape and many revenue streams. A modern Royalty Management System is crucial to effectively track and manage royalty information, ensuring exact calculations and timely payments to artists and their stakeholders.

# Functional Requirements:

FR1. Admin, Artists and Managers can log in and access the system with proper permissions. FR2. Based on the role of the user, they will be redirected to respective dashboard.

FR3. If a user is logging in for the first time, they will be redirected to a page to change their password.

FR4. Login will have forgotten password component.

FR5. Admin can create accounts for new Artist and Manager. (Artist and manger cannot register themselves to the Portal)

FR6. Admin can view a dashboard with visualizations of top performing Artists and Managers based on Streams and Royalty.

FR7. Admin can soft delete the account of a user.

FR8. Music usage data is ingested and stored daily for Artists and Mangers to access. FR9. Royalty calculation based on streams. Calculation is given by slab wise

* **streams × 0.002** (if streams≤10,000)
* **10,000 × 0.002+(streams−10,000) × 0.005** (if 10,000<streams ≤50,000)
* **10,000×0.002+(50,000−10,000) × 0.005 + (streams−50,000) × 0.01** (if streams>50,000)

FR10. Artists can view a dashboard with visualizations of top performing songs based on Streams and Royalty.

FR11. Arist can request Manager to form a partnership. When accepted the Artist can view the Manager and Vise versa.

FR12. Artists can add their songs and get them listed on their contents page with data like song title, release date with collaborators if any.

FR13. Managers have access to a dashboard displaying visualizations of the top-performing songs by artists who have collaborated with them, ranked by streams and royalties.

FR14. Artists and Mangers can download or export their royalty transactions as PDFs. FR15. Artists and Mangers can manage their accounts, including updating their profiles

FR16. Artists and Mangers will receive notifications for royalty payments and account updates and request notifications.

FR17. Artists and Mangers can view the royalty payments and receive transaction confirmations.

# Non-Functional Requirements:

* + User credentials will be stored in the database and encrypted.
  + The RMS must be capable of scaling seamlessly to accommodate growing numbers of artists and increasing data volumes without compromising performance.
  + The RMS must deliver fast response times and high availability to ensure a seamless user experience.
  + The RMS should prioritize usability and user experience to ensure ease of use and adoption by artists and other stakeholders.
  + The RMS must demonstrate high reliability and robustness to minimize the risk of system failures or data loss.

# Architecture Constraints:

* + The architecture must follow a microservices pattern to allow independent deployment and scaling of components. Such as Royality module, Dashboard module, Alert module, User management module.
  + Password will be stored after encryption and the data will be stored.
  + There must be automated testing in place to ensure at least 80% coverage in test cases and must pass all the gateways in SonarQube.
  + Proper variable names must be used to enhance code readability and maintainability. Static data must be stored in external configuration files (e.g., JSON, YAML, properties) and not hardcoded within the application code. Consistent code formatting and styling should be enforced across the codebase. Comprehensive code documentation and comments should be provided for complex logic and algorithms.

# Scope:

* 1. **In Scope:**
     + Ensure access control to the system and authenticate users.
     + Provide comprehensive dashboards and reporting tools for royalty information.
     + Calculate royalties owed to artists based on contractual agreements and usage data.
     + Manage user accounts and control access to royalty data based on roles.
     + Notify the users about the mapping requests and highest trending songs.
     + Make payment to the Artists on their royalty.

# Out Scope:

* + - Receiving daily feed files from various sources containing information about music usage cannot be implemented due to inaccessibility of external APIs.
    - The information going into 10 M records coming daily as feed files from various sources which are processed and put into the system not implementable due to limited data access.
    - Handling of violations of copyrights cannot be implemented due to complex legal structure.