# Software Requirements Specification for

# MeloFi

| | | |
|---|---|---|
| Submitted by | : | Rahul Krishna R |
| Submitted to | : | amFOSS Anveshan 2025 |
| Submission Date | : | 14-11-2025 |
| Base Version | : | 1.0 |

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to describe the basic requirements and functionalities of MeloFi, a simple application designed for music exploration and manging (mobile and web). This SRS clearly points out about the app's goals, core features, and behaviour so that development can proceed smoothly. It serves as a reference for the contributors and reviewers who are involved in designing and evaluating this project.

## 1.2 Overview of the App

MeloFi will be a lightweight music app in which the users could explore new tracks, listen to music, and organize their favourite songs into playlists. Users can also create an account to explore, save playlists and follow other users. MeloFi focuses on being a simple audio experience console without advanced features that requires heavy programming. MeloFi aims to be simple, easy to use, and accessible on both web and mobile platforms.

## 1.3 Intended Audience

1. My classmates/seniors willing to contribute or evaluate my project.
2. Anyone willing to experience simple and enhanced music experience through MeloFi.

# 2. Description of the System

## 2.1 Core Idea

The core idea of MeloFi is to provide a minimal music streaming platform where users can discover tracks, listen music, and create personal playlists. A simple user system supports the users to follow each others to maintain basic connectivity between users, ultimately gifting a rich audio experience.

## 2.2 Main Features

1. Basic user account system (login/signup)
2. Discover or browse music on the home screen
3. Search for songs
4. Play audio tracks
5. Like and share the link of songs
6. Create and manage playlists
7. Follow and get followed by other users

# 3. Functional Requirements

1. The system must have the feature of user sign up and log in.
2. The system should allow users to play, pause, and seek within a track.
3. The application must display trending or recommended tracks in the home page.
4. The users should be able to search for songs, and play them.
5. The users should be able to create playlists and manage them.
6. The system should allow users to like or share songs, add them to playlists.
7. All users should be able to follow other users.
8. Profiles should display the user's playlists, followers, and following count.
9. The system should allow playback on both mobile and web versions.

# 4. Non-Functional Requirements

1. Usability: The interface must be minimal, clean, premium and easy to navigate for beginners.

2. Performance: Pages should load quickly, and audio playback should start with minimal delay.

3. Reliability: The app should function smoothly without frequent crashes or bugs.

4. Compatibility: Must adapt to display on different screen sizes for both web and mobile.

5. Security: User credentials should be stored securely, and access to playlists must be account-restricted.

6. Scalability: The system should be able to handle the data of various playlists and users safely and without crashes.

# 5. Basic Tech Stack

## 5.1. Wireframing & UI Design

Figma: Used for creating the structure or wireframes and high-quality UI designs for all screens for MeloFi. (Welcome, Home, Search, Player, Playlist, Profile).

## 5.2. Frontend

React.js: a JavaScript library for building the web application.

Kotlin: - for building the Android APK version of MeloFi.

Android Studio – Which is the main IDE for Android development.

## 5.3. Backend

Flask (Python): - for the backend framework.

REST APIs: - for handling the data exchange between frontend and backend.

# 6. Conclusion

MeloFi is designed as a beginner-friendly music platform focusing on simplicity and basic features. This SRS provides a clear and minimal structure for developing the app without unnecessary complexity. By following this document, contributors can create a functional and smooth music experience that works across both web and mobile platforms. This project also leaves room for future enhancements and scalability if required.