

Open Brief Proposal

Mwape Kurete, 23115

Open Window, School of Fundamentals

DV200

Lecturer: Tsungai Katsuro

2024/09/12

Table of Contents

Project Proposal Overview.....	3
Target Audience.....	3
Technology Stack.....	4
Application Features.....	4
Database Design.....	5
UI and UX Design.....	5
Security Considerations.....	6
Project Timeline.....	6
Week 1: Project Planning and Requirements Gathering.....	6
Week 2: Development Environment Setup and Backend (MongoDB, Express, Node.js).....	6
Week 3: Frontend Setup and Core Features (React).....	6
Week 4: Complete Core Features and Basic UI.....	7
Week 5: Spotify API Integration & Friend Activity.....	7
Week 6: AI Integration for Music Recommendations.....	7
Week 7: UI/UX Refinements, Security, and Testing.....	7
Week 7.5: Final Testing and Deployment.....	7
Challenges and Risks.....	8
Conclusion.....	8

Project Proposal Overview

The vast expanse of music genres and albums makes it difficult for users to discover, categorise, and share their preferences efficiently. While services like Spotify provide music streaming, they often lack the depth of interaction and personalised recommendation systems beyond algorithm-driven suggestions. There is a need for a platform that enables users to share reviews, discover new albums, track friends' listening habits, and build personalised, curated lists of albums and artists.

This application aims to solve the issue of fragmented music discovery by creating a community-driven platform for music lovers. Users will be able to create, rate, review, and categorise music albums, as well as follow friends to track their top albums and recent listens. This interactive experience will connect like-minded individuals and enhance the social aspect of music discovery, all within a comprehensive user experience.

Target Audience

The primary users of this application are music enthusiasts, from casual listeners to audiophiles, who enjoy discovering new music, sharing their opinions through reviews, and engaging with a community of like-minded individuals. The target audience also includes users who already enjoy similar services like Letterboxd (for movies), RateYourMusic (for albums), and Spotify's social features but desire a platform that focuses exclusively on music discovery and album curation.

The application will benefit these users by offering a social space for interaction and discovery, allowing them to share their musical tastes and follow friends' music activities, which is currently lacking in most music apps.

Technology Stack

The application will use the **MERN stack** (MongoDB, Express, React, Node.js).

- **MongoDB** will be used for the database to store user profiles, album information, reviews, and ratings in a non-relational structure, which is ideal for flexible data structures like user reviews and music tags.
- **Express** will serve as the backend framework, handling routing and API requests for CRUD operations related to albums, users, and reviews.
- **React** will power the front-end, creating a dynamic and interactive user interface that offers a smooth experience when browsing albums, reviewing, and following friends.
- **Node.js** will be used to manage the server-side operations and handle asynchronous requests, making it suitable for real-time features like viewing friends' currently playing music.

The MERN stack is chosen for its scalability, fast development process, and ability to build highly dynamic, responsive web applications.

Application Features

- **User Profiles:** Users can create profiles, display their top 5 favourite albums and view or edit their reviews.
- **Album Reviews:** A recommendation system similar to musicroamer.com and everynoise.com, using content-based filtering to suggest albums based on users' search queries.
-

- **Music Discovery Tool:** A recommendation system similar to musicroamer.com and everynoise.com, using content-based filtering to suggest albums based on users' search queries.
- **Friend Activity:** Display what music the platform's users have reviewed and rated. Users will be shown albums of that have the most reviews and albums that have the highest average rating. Users can also read and like reviews posted by others.

Database Design

- **Users Collection:** Stores user profile data, liked albums, reviews (links to reviews collection), and lists of favourite albums.
-
- **Albums Collection:** Contains album details (name, artist, release date, genre) and links to user reviews.
- **Reviews Collection:** Holds user-generated reviews, ratings, and metadata like timestamps and associated albums.

The database will be structured to optimise for speed and flexibility, ensuring that user reviews and ratings can be quickly fetched and displayed.

UI and UX Design

The UI will resemble the minimalist structure of Letterboxd, focusing on ease of navigation, aesthetic album displays, and seamless integration of social elements. The application will feature clear, easy-to-use filters for music discovery and prominent space for user reviews and friend activity. The UX will be tailored to create a community-driven

experience where users feel engaged with both music content and their friends' musical journeys.

Security Considerations

- **Data breaches:** To protect user data, all sensitive information, including user credentials, will be encrypted. The app will use secure communication protocols (HTTPS).
- **Input validation:** Strict input validation rules will be applied to prevent SQL injection or cross-site scripting (XSS) attacks.
- **Authentication and authorization:** User authentication will be handled with JSON Web Tokens (JWT) to ensure that only authorised users can access specific parts of the app, such as personal profiles and review features.

Project Timeline

Week 1: Project Planning and Requirements Gathering

- Define the problem statement, target audience, and core features.
- Finalise the technology stack and database design.

Week 2: Development Environment Setup and Backend (MongoDB, Express, Node.js)

- Set up MongoDB, Express, and Node.js for CRUD operations.
- Implement backend API routes for user profiles, reviews, and ratings.

Week 3: Frontend Setup and Core Features (React)

- Build React components for user profiles, album reviews, and CRUD operations.
- Start connecting the frontend to the backend.

Week 4: Complete Core Features and Basic UI

- Finalise CRUD operations across frontend and backend.
- Create basic UI for user interaction (reviews, rating system, user profiles).

Week 5: Spotify API Integration & Friend Activity

- Implement Spotify API to pull friend activity data and integrate it into user profiles.
- Test real-time friend activity feed.

Week 6: AI Integration for Music Recommendations

- Implement GPT or AI-based recommendations for personalised music discovery.
- Integrate AI into the music discovery feature.

Week 7: UI/UX Refinements, Security, and Testing

- Refine UI for a smoother user experience.
- Implement security measures (JWT, input validation, etc.).
- Begin user testing.

Week 7.5: Final Testing and Deployment

- Finalise any remaining issues and conduct a final round of testing.
- Deploy the application and ensure post-launch support plans are in place.

Challenges and Risks

- **Spotify API Limitations:** Integrating real-time data from the Spotify API could pose a challenge due to potential rate limits or access restrictions. This will be addressed by optimising API calls and considering fallback options for when data is unavailable.

- **AI Integration:** Integrating a GPT-powered recommendation system may require careful fine-tuning to ensure that suggestions are relevant and valuable.
- **User Privacy:** Handling and storing user activity (like music history) must comply with privacy regulations to avoid legal issues.

Conclusion

This project aims to create an innovative and interactive music discovery platform with a rich set of features tailored to music enthusiasts. The application will not only solve the issue of fragmented music discovery but also foster a social community around music sharing and reviews. With its combination of AI integration and real-time friend activity, the app will offer a unique user experience that stands out in today's music ecosystem.