Music Analytics Platform Project Meeting Minutes

Date: March 7, 2025
Topic: Project Methodology and Requirements
Participants: Team Members (3)

1. Methodology Discussion

Decision: Agile Methodology

The team evaluated both Waterfall and Agile methodologies and decided to adopt **Agile** for the following reasons:

- **Flexibility for user feedback integration**: Allows for incorporating user feedback throughout development
- **Iterative development approach**: Enables building the project incrementally and making improvements
- **Adaptability to changing requirements**: Supports modifying features based on user needs
- **Project-specific suitability**: Better aligned with the interactive and evolving nature of an analytics platform

2. Requirements Specification

2.1 Business Requirements

1. Create a platform that leverages Spotify API to provide music data analysis
2. Offer both free (visitor) and premium (registered user) features
3. Deliver insights valuable to music enthusiasts, playlist curators, and industry professionals
4. Implement monetization options through premium features

2.2 User Requirements

1. Access music trends without login (visitor access)
2. Authenticate via Spotify account for personalized analytics

3. Analyze personal music preferences and listening patterns
4. Compare music across different time periods (e.g., 90s music vs. current)
5. Discover new music based on analytics insights
6. Share music insights with friends and family
7. Export analytics data in various formats

2.3 Functional Requirements

1. **Authentication System**
    - User login/registration via Spotify OAuth
    - Session management
    - Profile management
2. **Analytics Engine**
    - Process music data for trend analysis
    - Generate personalized insights
    - Perform comparative analysis
3. **Visualization Module**
    - Create interactive charts and graphs
    - Display music trends visually
    - Present user-specific analytics
4. **Export and Sharing**
    - Generate downloadable reports
    - Implement social sharing functionality

2.4 Non-Functional Requirements

1. **Security**
    - Data encryption
    - Secure authentication
    - Password protection
2. **Performance**
    - Fast response times
    - Efficient data processing
    - Handle multiple concurrent users
3. **Scalability**
    - Support growing user base
    - Optimize database performance
4. **Reliability**

    - Implement data backups
    - Error logging and monitoring
    - Password recovery mechanisms

2.5 System Requirements

1. **Technology Stack**
    - Backend: Django
    - Database: PostgreSQL
    - Frontend: HTML, CSS, JavaScript
    - Version Control: Git for team collaboration
2. **External Integrations**
    - Spotify API integration
    - OAuth implementation
3. **Deployment**
    - Deployment platform: Render or Wagtail
    - Free hosting options for project demonstration

3. Next Steps

- Continue detailed discussions in the team group
- Begin development planning based on the Agile methodology
- Set up project repository and collaborative development environment