

# Approach Document for AI-Powered Music Generation Application



The AI-powered music generation app aims to offer users a customized experience by allowing them to adjust energy levels, genre mix, and tempo. Supporting Progressive house, Psychedelic techno, and Deep house, the app generates tracks up to 9 minutes in duration, following genre-based rather than thematic analysis. Users can interact by previewing and downloading up to 5 tracks simultaneously at 320kbps quality. The development will begin with an Android version, inspired by Loudly.com's style and utilizing client reference songs. To enhance user experience, algorithm improvements will focus on avoiding repetitive structures. A feedback system will be implemented for users to request adjustments.

## 1. User Customization:

To implement user customization, we'll create an intuitive interface allowing users to input their preferences. Sliders or dropdowns for energy levels, genre mix, and tempo will be incorporated. The backend will process these inputs to guide the AI in music generation.

## 2. Genres:



We'll focus on three genres: Progressive House, Psychedelic Techno, and Deep House. Each genre will have distinct characteristics fed into the AI model for accurate music generation.

### **3. Duration:**

Ensuring tracks have a duration of up to 9 minutes requires adjusting the model's parameters. We'll fine-tune the model to produce compositions within the desired time frame while maintaining coherence.

### **4. Thematic Analysis:**

Since thematic analysis is not required, we'll concentrate on genre-based music generation, using the specified genres as the primary criteria for our AI model.

### **5. User Interaction:**

The user interface will allow users to generate and preview up to 5 tracks simultaneously. We'll implement a download option with quality up to 320kbps for user convenience.

### **6. Platform Support:**

Initially, the application will be developed for Android. Future plans include developing versions for iOS and web. Cross-platform compatibility will be a consideration during development.

### **7. Reference and Inspiration:**

We'll use the client's 500 songs as a reference for AI-generated music. Loudly.com's style approach will guide the aesthetics and feel of the generated music.

### **8. Algorithm Improvement:**

To enhance the algorithm, we'll implement measures to avoid repetitive structures and introduce more variation in the start and end of songs. This will provide users with diverse and engaging music tracks.

### **9. User Feedback and Adjustment:**

A "Contact Us" section will be incorporated with a provided email for users to provide feedback or request adjustments. Regularly monitoring this feedback will help us refine the model and enhance user satisfaction.

### **10. Python Library and Deployment Tools:**

For seamless development, we'll use Python with TensorFlow and PyTorch for genre-specific models. Leveraging Flask for backend API, we'll employ JavaScript, HTML, and CSS for a responsive frontend. Deployment will be on Google Cloud Platform with Kubernetes for scalability. Continuous integration via GitHub Actions will ensure code quality. For user customization, we'll utilize sliders with JavaScript event listeners. User feedback will be managed through Flask-Mail. Model improvement will involve continuous training with feedback integration. Android deployment will be achieved using tools like Android Studio. Future iOS and web versions will employ cross-platform frameworks like React Native.

### **Detailed Plan for Implementation:**

#### **User Customization:**

- Develop an intuitive interface with sliders/dropdowns.
- Backend processing to interpret user preferences.
- Input parameters into the AI model for personalized music generation.

#### **Technical Strategies for Genres and Durations:**

- Fine-tune the AI model with genre-specific data.
- Adjust model parameters to meet the 9-minute duration requirement.

#### **Design Considerations for UI:**

- Ensure a clean and intuitive interface.
- Implement a preview feature for generated tracks.
- Integrate a download option with quality selection.

#### **Algorithm Improvement:**

- Analyze and modify the existing algorithm to avoid repetitiveness.
- Introduce variations in the beginning and end of the generated tracks.

#### **Handling User Feedback:**

- Regularly check the "Contact Us" section for user feedback.
- Implement an iterative improvement process based on user suggestions.
- Provide timely updates addressing user concerns.

### **Roadmap for Development:**

#### **1. Phase 1 (Months 1-2): Develop the Android version.**

- Focus on core functionalities: user customization, genre-specific AI training, and duration control.
- Initial UI design and testing.

#### **2. Phase 2 (Months 3-4): Refinement and Testing.**

- Integrate user feedback system.
- Refine the AI model based on initial user interactions.
- Rigorous testing for stability.

#### **3. Phase 3 (Months 5-6): Android Release and Future Planning.**

- Launch the Android version.



- Gather user feedback for future improvements.
- Start planning for iOS and web versions.

#### **4. Future Phases:**

- Months 7-9: Develop iOS version, considering platform-specific requirements.
- Months 10-12: Begin web version development.
- Beyond Year 1: Continuous improvement based on user feedback, explore additional features, and expand platform support.

**This approach ensures a systematic and user-centric development process, delivering a high-quality AI-powered music generation application on Android, with a clear roadmap for future expansions.**