AI Content Generation – Submission Report

1. Environment Setup Documentation

APIs configured included Google Gemini API (used successfully for Lyria music generation) and KlingAI API (attempted for video generation). Environment variables were managed through a .env file.

Major setup issues included uv not being recognized on Windows, environment variables not loading automatically, Google Veo quota exhaustion, KlingAI JWT authentication failures, and FFmpeg not being available in PATH after installation.

These were resolved by explicitly installing uv via pip, manually loading .env using python-dotenv, switching providers when quotas were exhausted, inspecting provider source code directly, and identifying FFmpeg binaries manually.

2. Codebase Understanding

The architecture follows a provider-registry pattern. Providers are organized by vendor (google, kling, aimlapi). Pipelines abstract orchestration for music and video generation, allowing interchangeable providers.

The pipelines directory coordinates configuration, execution, and result handling, while providers implement vendor-specific API logic. This separation made debugging and provider switching possible.

3. Generation Log

Commands executed included ai-content list-providers, ai-content list-presets, ai-content music generation using Lyria, and multiple attempts at video generation using Veo and Kling.

Prompts focused on cinematic visuals and ambient music. Music generation succeeded, producing 8s and 30s WAV files. Video generation failed via APIs but succeeded externally using Pika.

4. Challenges & Solutions

Initial failures included API errors, invalid arguments, quota exhaustion, and authentication issues. Troubleshooting involved reading error logs, inspecting source code, testing environment variables, and isolating failures.

A key workaround was generating video externally using Pika and manually saving outputs into the exports folder, allowing the pipeline to be completed conceptually.

5. Insights & Learnings

The provider abstraction was more robust than expected. API instability highlighted the importance of graceful fallbacks. Compared to other AI tools, this system emphasized engineering discipline over convenience.

6. Links

GitHub Repository: https://github.com/Amy12R/AI_Content_Generation
Pika Labs: https://pika.art
Runway ML: https://runwayml.com