CONFIDENTIAL AUDIT REPORT

Tech Due Diligence Report: AudioLoop AI Song Recommender

Micah 6 AI · January 17, 2026 · Critical Findings Identified

Product Claim

AI Song Recommender for Spotify

Claimed Tech Stack

Python, Spotify API, DeepFace

12-Month Vision

Apple Music, Deezer, Spotify, Shazam + Netflix movie recommendations

TECHNICAL HEALTH SCORECARD

Overall System Assessment

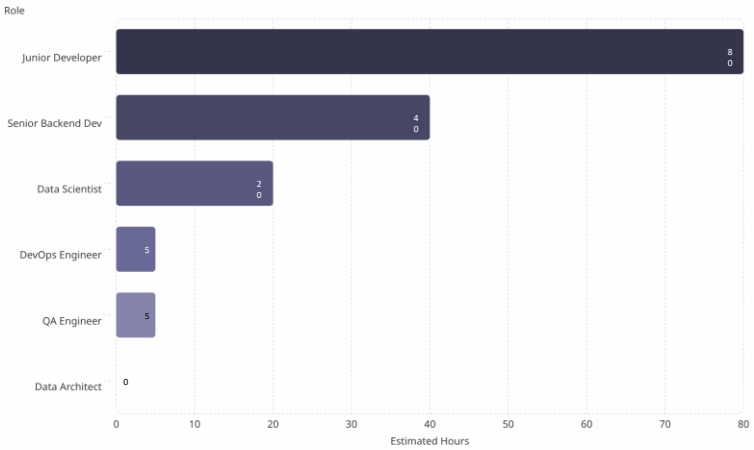
Eight critical dimensions were evaluated across the AudioLoop codebase. The results reveal a system in early, experimental form — with several areas requiring immediate remediation before any production deployment or investor consideration.

Category	Status	Description
Security	🔴 Critical	Hardcoded credentials, arbitrary code execution via `eval()`
Architecture	🔴 Critical	Monolithic, tightly coupled, poor separation of concerns
Code Quality	🟠 High	Magic paths, fixed parameters, inconsistent dependency management, lack of error handling, redundant API calls
Algorithmic Integrity	🔴 Critical	DeepFace bias, black-box VAD mapping, heuristics over true ML
Scalability	🔴 Critical	Not designed for the ambitious 12-month vision; hardcoded elements block growth
Maintainability	🟠 High	High technical debt due to coupling, hardcoding, and lack of modularity
Testability	🟠 High	Lack of clear component separation makes unit testing difficult
Observability	🟡 Medium	General lack of logging/monitoring practices inferred from architectural flaws

CONSTRUCTION EFFORT

Build Effort Estimate

The following breakdown reflects the estimated hours likely required to build the *current* uploaded codebase, based on its complexity, structure, and identified issues — not the effort required to build a production-ready system.



Template Check Result: The code appears to be an **original creation** developed in an early, experimental, or rapid prototyping phase. It does not exhibit the structured patterns, modularity, or best practices found in well-maintained boilerplate or purchased templates.

The near-zero Data Architect involvement is telling — the system lacks any intentional data layer design. The heavy Junior Developer hours reflect exploratory, unstructured coding rather than engineered architecture.

ALGORITHMIC INTEGRITY

The "AI" Claim: A Critical Examination

> It's like using a very old, unverified mood ring to pick songs — without truly understanding how the mood ring works or if it's accurate for everyone.

The core claim of "AI Song Recommender" hinges on DeepFace for emotion recognition and a custom Valence-Arousal-Dominance (VAD) mapping. This architecture presents severe challenges to algorithmic integrity and introduces substantial bias risks.

Step 1: DeepFace.analyze

External pre-trained neural network detects emotion from a user image. This is the only true ML component in the pipeline.

Step 2: Hardcoded VAD Mapping

Raw emotion probabilities are converted to a Valence-Arousal-Dominance vector using fixed, empirically undocumented rules — pure heuristics.

Step 3: Similarity Search

Songs in `muse_v3.csv` "closest" to the VAD vector are returned. A simple distance-based search — not a learning model.

The system is **not a true AI/ML recommendation model**. It is a proof-of-concept that uses an external AI for input processing, then applies deterministic, heuristic-based logic to produce recommendations.

BIAS ANALYSIS

Statistical & Societal Bias Risks

Three distinct categories of bias threaten the integrity and fairness of AudioLoop's recommendations. Each represents a systemic risk that cannot be patched — only re-architected.

Selection Bias — DeepFace

Pre-trained models like DeepFace are notorious for biases in their training data. If skewed toward certain demographics (Western, younger, specific lighting, gender), it will perform poorly for diverse users. Cultural differences in expressing emotion are often not captured, leading to systematic misinterpretations and unfair recommendations.

Confirmation Bias — VAD Mapping & Data

Hardcoded `VALUES` for VAD mapping introduce confirmation bias. The system is "designed" to interpret emotions in a fixed way rather than learning from user feedback. If `muse_v3.csv` is dominated by certain genres or emotional ranges, recommendations will only confirm these biases, perpetuating a narrow musical scope.

Construct Validity — Emotion-to-VAD

There is no documented scientific or empirical basis for the `VALUES` and `VAD_max` constants. Is 'fear' truly mapped effectively to a specific VAD range? Is this mapping universally valid? Without validation, the system may not be measuring what it *claims* to measure — user emotion for song recommendation.

BLACK BOX PROBLEM

Opacity vs. Explainability

What the System Cannot Explain

- Why DeepFace assigns a certain emotion probability
- How 'anger' transforms into a specific VAD score
- Why a particular song was recommended over another
- Whether the VAD mapping is accurate for any given user

What True AI/ML Would Provide

- Continuous learning from liked/disliked songs
- Personalization beyond the initial emotion input
- Adaptation to evolving musical tastes and trends
- Custom-trained recommendation engine with feature engineering
- Adaptive learning mechanisms and robust model architectures

⚠ **Black Box Within a Black Box:** DeepFace is an external black box. The VAD mapping is an internal opaque heuristic. The `VALUES` and `VAD_max` are "magic numbers" without explanation — making the crucial translation step a black box *within the company's own code*.

CRITICAL VULNERABILITY

Vulnerability #1: Hardcoded Credentials

Non-Tech Explanation

Leaving your business's "keys to the kingdom" openly visible in the code is like writing your bank account and PIN on a sticky note and leaving it on your computer. Anyone who sees the code can access your Spotify developer account, potentially incurring costs, making unauthorized API calls, or disrupting your service.

Location: `CLIENT_ID` and `CLIENT_SECRET` are directly embedded in `AudioLoop-main/authorization.py` (lines 3–4).

Risks & Consequences

- **Security Breach:** Immediate compromise of Spotify API credentials
- **Financial Loss:** API abuse leading to unexpected charges
- **Service Disruption:** Spotify could revoke access if credentials are leaked
- **Reputation Damage:** Loss of user trust due to security negligence

Solution

Store sensitive credentials in environment variables or a secure secrets management service (AWS Secrets Manager, Google Secret Manager, Azure Key Vault, HashiCorp Vault). Access them dynamically at runtime.

FIX EFFORT: 4 HOURS

CRITICAL VULNERABILITY

Vulnerability #2: Arbitrary Code Execution via eval()

Non-Tech Explanation

Using `eval()` on data from the internet is akin to letting a stranger write instructions directly into your computer's operating system. If their input is malicious, they could take full control of your server, steal all your data, or install malware.

Location: `eval()` on an external API response in `AudioLoop-main/crawl.py` (line 39).

Risks & Consequences

- **Complete System Compromise:** Attacker executes arbitrary code on the server
- **Data Exfiltration/Modification:** Sensitive data stolen or corrupted
- **Denial of Service:** Server crashed or rendered inoperable
- **Legal/Compliance Issues:** Severe breach implications

Solution

Replace `eval()` with `json.loads()` for parsing JSON API responses. Ensure proper error handling around JSON parsing.

FIX EFFORT: 2 HOURS

HIGH-RISK VULNERABILITIES

Vulnerabilities #3–#5: Architecture, Paths & Parameters

1

Monolithic Architecture

Components are tightly coupled through direct imports and global script execution. Flask is listed but unused. Cannot scale individual components independently. The 12-month vision of integrating multiple platforms will be **impossible** to implement efficiently in this architecture.

Fix: Implement microservices or well-defined service layers. Utilize Flask/FastAPI correctly for API endpoints.

120 HOURS · DATA ARCHITECT + SENIOR BACKEND DEV

2

Magic Paths

`sys.path.append("../spotify_api_web_app")` uses a fragile, hardcoded relative path. Application will fail in Docker containers or cloud instances where the relative path doesn't exist.

Fix: Use Python's package management features correctly. Configure `PYTHONPATH` appropriately in deployment environments.

8 HOURS · SENIOR BACKEND DEV + DEVOPS

3

Fixed Parameters

Critical settings baked into code: `n_recs=100`, `time.sleep(0.2)`, `VAD_VALUES`, `VAD_max`, test image paths, CSV filenames, Spotify base URL. Cannot adjust without code changes and redeployment.

Fix: Externalize into configuration files (`.ini`, `YAML`, `.env`) or a database. VAD values need scientific validation.

16 HOURS · SENIOR BACKEND DEV + DATA SCIENTIST

MEDIUM-RISK VULNERABILITIES

Vulnerabilities #6–#8: Dependencies, Errors & API Calls

Inconsistent Dependency Management

Two separate requirements files (`req.txt` and `requirements.txt`) create ambiguity. Different versions can cause runtime errors, broken CI/CD pipelines, and unclear authoritative source for dependencies.

Fix: Consolidate into a single `requirements.txt`. Use pip-tools, Poetry, or conda for version control.

4 HOURS · JUNIOR DEVELOPER

Lack of Error Handling

No `try-except` blocks for API calls, network issues, or data parsing. Potential `IndexError` in `recommend.py` if `result.dropna()` yields an empty DataFrame. Application crashes instead of recovering gracefully.

Fix: Implement robust try-except blocks for all external interactions. Log errors. Consider circuit breakers for external services.

20 HOURS · SENIOR BACKEND DEV

Redundant API Calls

`sp.track()` and `sp.audio_features()` are called separately for each track ID when batching is available. Doubles network overhead, accelerates rate limit consumption, and slows data retrieval.

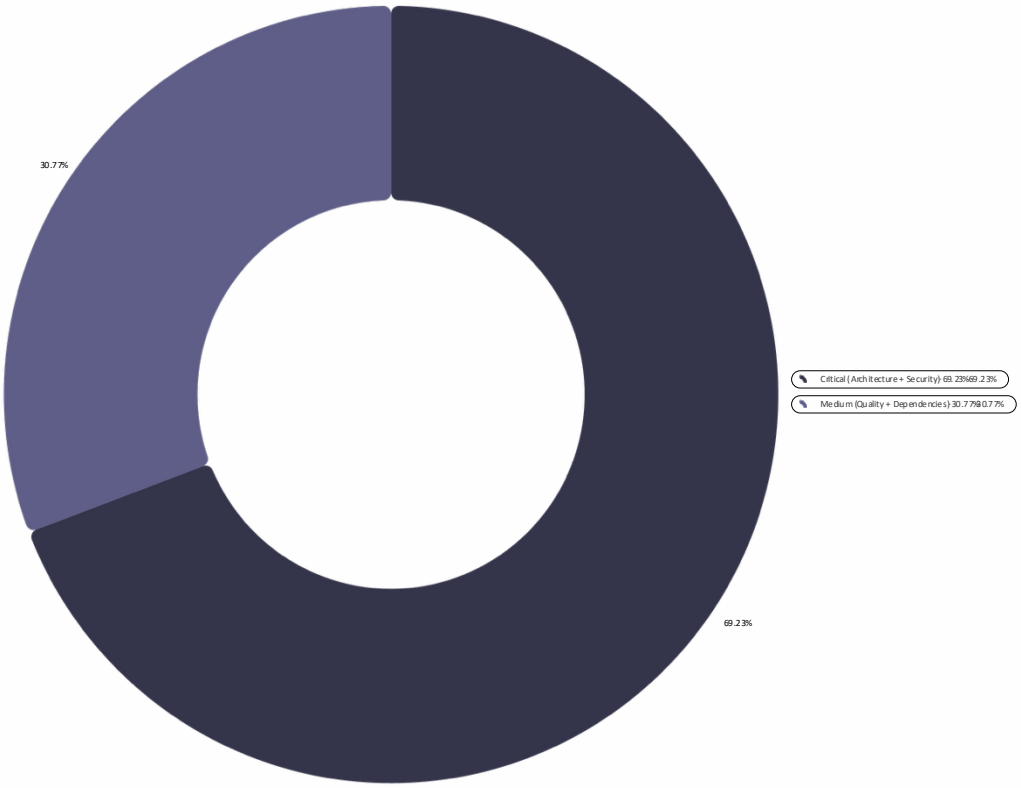
Fix: Leverage Spotify API batching: `sp.tracks()` and `sp.audio_features()` accept multiple IDs simultaneously.

8 HOURS · SENIOR BACKEND DEV

REMEDIATION SUMMARY

Total Remediation Effort: 182 Hours

This table summarizes the estimated hours required to address all Critical, High, and Medium risks identified in this report. Note: this is the cost to *fix* the current code — not to build the ambitious 12-month roadmap.



Risk Severity	Total Hours	Required Expertise
Medium	56 hours	Senior Backend Dev, Data Scientist, Junior Developer, DevOps Engineer
Critical	126 hours	Data Architect, Senior Backend Dev, DevOps Engineer
Total Sum	**182 hours**	Full cross-functional engineering team

SMB / FOUNDER VERDICT

Final Recommendation for SMBs & Founders

REFUSE WITH REVIEW OPPORTUNITY

The current codebase, while demonstrating an initial proof-of-concept, is burdened with severe security vulnerabilities, fundamental architectural flaws, and critical issues regarding its core algorithmic integrity. The claim of "AI" is misleading — the system primarily uses a third-party AI model and then applies heuristics.

1 Critical Security Risks

Hardcoded credentials and arbitrary code execution are immediate and existential threats that must be addressed before any further development or deployment.

2 Unsustainable Architecture

The monolithic, tightly coupled design will make the ambitious 12-month vision (integrating multiple platforms, movies) impossible to achieve efficiently, reliably, or at scale.

3 Algorithmic Misrepresentation & Bias

The core "AI" aspect is underdeveloped and prone to significant bias, opacity, and lacks true machine learning for recommendations — undermining the product's value proposition and posing ethical risks.

4 High Technical Debt

The estimated 182 hours of remediation represent a substantial refactoring effort that is essentially a partial rebuild of the core system. This debt will paralyze future innovation.

Review Opportunity: A review opportunity exists if the founders commit to a complete re-architecture, a robust re-evaluation of the AI/ML strategy (moving beyond heuristics to actual learning models), and thorough remediation of all critical security issues. The current codebase serves as a functional demo only.

INVESTOR VERDICT

Final Recommendation for Investors (VC/PE/Angels)

HIGH RISK · LOW IP VALUE

BUY (Current Asset) — Not Recommended

- Inherits immense technical debt and immediate security liabilities
- Cannot scale to the stated 12-month vision
- Core "AI" value proposition is thin and misleading
- 182 hours is conservative — true roadmap cost is significantly higher
- Acquisition cost buys the *idea*, not a defensible tech asset

BUILD (From Scratch) — Strongly Favored

- Clean, scalable microservices-based architecture from day one
- Proper security controls and robust error handling
- Genuinely adaptive AI/ML recommendation engine
- More efficient long-term, faster time-to-market for roadmap features
- The "head start" from current code is minimal due to required refactoring

Intellectual Property Value Evaluation

Uniqueness

Concept has novelty. Implementation (hardcoded VAD, similarity search) is **not unique or proprietary**. DeepFace is open-source.

Robustness

Severely lacking. Fragile, prone to crashes, not designed for production. **Zero robustness** as a commercial IP asset.

Defensibility

Algorithmic core is generic and easily replicable. No unique data advantage or proprietary ML model. **Not defensible.**

Overall Investor Verdict: Proceed with extreme caution, or refuse unless significant re-architecting and re-evaluation of the AI strategy is planned. An investment would be in the *team and the vision*, not the existing technology.

NEXT STEPS

Human Validation & Next Steps

Don't let algorithms alone decide your company's fate. This report was generated via the **Micah 6 AI Audit Engine**. Its content is AI-generated and should be validated by human engineers before any investment or acquisition decision is made.

Book a Human-in-the-Loop Audit

Get a bespoke Tech Due Diligence Audit with our executive data architects for human validation of these findings. Visit micah6ai.com to get started.

Evaluating a Second Sprint?

Access a new automated audit at micah6ai.streamlit.app/?vip=true and use code **WELCOMEBACK20** at checkout for 20% off your next automated audit.

Permanent Access Link

Access your full audit session at any time via your permanent session link provided at checkout. All findings are preserved and shareable with your technical advisors.

This report was generated via the Micah 6 AI Audit Engine. Its content is AI-generated and should be validated by human engineers. © 2023 Micah 6 AI · micah6ai.com