

# Project Log - AI Voice Platform

---

## Concept Overview

**Goal:** Build an AI-driven music app that can deepfake a user's own voice for singing or rapping, with professional-quality filters and FX. The app only clones the user's real microphone recordings (no uploads, no impersonation) and allows full creative control over vocal style and production.

**Core Features:** - Voice cloning (user-only) - Mic verification + replay detection - Singing voice synthesis / rap performance rendering - Post-FX chain (EQ, compression, reverb, harmonizer) - Secure backend & consent logs - Hybrid AI pipeline (vendor API + local GPU)

---

## System Architecture Decisions

- **Architecture Type:** Hybrid – vendor API for fast prototype, local GPU models for scalability.
  - **Security:**
    - Real mic verification
    - AI authenticity classifier (mic / replay / synthetic)
    - Fingerprint + consent linking
    - Audio watermarking and full deletion control.
  - **Stack:** Python + FastAPI backend with PostgreSQL or SQLite for prototyping.
  - **Frontend:** React (planned) for recording UI and generation dashboard.
- 

## Development Timeline

### Phase 1: Foundations

-  Decided on **FastAPI** for backend — better for ML integration.
-  Set up project structure:

```
app/  
    main.py  
    routers/  
    models/  
    schemas/  
    utils/
```

-  Added routes: `/auth`, `/record`, `/voice`, `/generate`, `/fx`, `/audit`.
-  Config + DB setup with SQLAlchemy.

## Phase 2: Voice Recording + Mic Verification

- Added `/record/verify-device` → filters out virtual inputs.
- Added waveform validation (`audio_utils.py`) → ensures natural mic recordings.
- Implemented spectral analysis + fingerprinting system.

## Phase 3: Voice Training Pipeline

- Added `/voice/train` → starts background job for model training.
- Added `/voice/status` → job tracking.
- Added background worker `run_training_job`.

## Phase 4: AI Authenticity Layer

- Added CNN-based mic authenticity classifier (`authenticity_model.py`).
- Integrated with `/voice/verify` endpoint.
- Added output: `label`, `confidence`.

## Phase 5: Vendor Integration (ElevenLabs)

- Integrated ElevenLabs **Instant Voice Cloning (IVC)** API.
- Added new endpoint `/voice/train/vendor` → uploads verified mic sample, creates vendor job, stores voice ID.
- Background task `_monitor_vendor_training` added for job polling.
- Stored vendor data in `voice_models` table.
- Implemented `/generate/vendor` for **voice generation** using ElevenLabs text-to-speech render endpoint.

### Vendor Code Summary:

```
voice = client.voices.ivc.create(name="UserVoiceClone", files=[open(temp_path,
"rb")])
voice_id = voice.voice_id
# Generate audio
output = client.text_to_speech.convert(
    voice_id=voice_id,
    text="Your custom song lyrics here",
    model_id="eleven_multilingual_v2"
)
with open("output.wav", "wb") as f:
    f.write(output)
```

## Phase 6 (Planned): Local GPU Inference

-  Prepare PyTorch-based SVC pipeline (HiFi-GAN vocoder).
-  Add option to switch between `vendor` and `local` training automatically.

## Phase 7 (Planned): User Consent + Linking

- Add consent schema + endpoint `/record/consent`.
  - Link user → recording → model → generation chain.
- 

## Next Steps

1. Expand vendor generation endpoint with lyric + beat sync.
  2. Add error handling and logging for ElevenLabs API calls.
  3. Implement consent and deletion workflows.
  4. Begin local SVC prototype setup (Phase 6).
- 

## Notes & Decisions Log

- Hybrid architecture allows fast MVP and later full ownership.
  - Mic-only training protects users and prevents impersonation.
  - FastAPI chosen for high-quality audio model integration.
  - *Markdown-based project log* implemented for transparency and backup.
  - Vendor-first rollout — ElevenLabs API integrated for initial cloning and generation.
  - Full vocal generation path added (training + TTS render).
- 

## Change History

- **2025-11-08:** Established FastAPI backend structure.
- **2025-11-08:** Added mic authenticity classifier (CNN-based).
- **2025-11-08:** Decided on Markdown project log with continuous updates.
- **2025-11-09:** Integrated ElevenLabs vendor API for cloning + TTS generation.