

🎵 AURA 프로젝트 지침서 V2.0

Project Trinity: 말론(CEO) + 클로드(CPO) + 제미나이(CTO)

문서 버전: 2.0

작성일: 2025년 12월 25일

최종 수정: 2025년 12월 25일

📋 목차

1. [프로젝트 개요](#1-프로젝트-개요)
2. [전략적 전환 배경](#2-전략적-전환-배경)
3. [타겟 고객 및 포지셔닝](#3-타겟-고객-및-포지셔닝)
4. [핵심 기능 명세](#4-핵심-기능-명세)
5. [기술 스택](#5-기술-스택)
6. [아키텍처](#6-아키텍처)
7. [UI/UX 디자인 시스템](#7-uiux-디자인-시스템)
8. [개발 로드맵](#8-개발-로드맵)
9. [역할 분담](#9-역할-분담)
10. [비즈니스 모델](#10-비즈니스-모델)
11. [경쟁사 분석](#11-경쟁사-분석)
12. [핵심 원칙](#12-핵심-원칙)

1. 프로젝트 개요

프로젝트명

AURA (아우라)

슬로건

"The Darkest Night, The Brightest Sound."

한 줄 정의

> "프로 뮤지션의 기존 자산(VST/샘플)을 100% 활용하며, AI로 창작의 한계를 넘어서는

구독형 하이브리드 AI DAW"

핵심 컨셉

- **Suno** = AI가 전부 해줌 (수동적, 아마추어 타겟)

- **AURA** = 내 VST + AI 어시스트 (능동적, **프로 타겟**)

2. 전략적 전환 배경

2.1 왜 웹 버전을 포기했는가?

웹 버전의 한계:

| 문제 | 설명 |

|-----|-----|

| VST 사용 불가 | 브라우저 샌드박스로 시스템 파일 접근 차단 |

| GPU 성능 제한 | AI 목소리 학습 등 고사양 작업 불가 |

| 레이턴시 | Web Audio API 한계 (50~100ms) |

| 경쟁 열세 | Suno Studio 출시로 웹 AI DAW 시장 포화 |

결론:

> "웹 버전만으로는 Suno의 열화판이 될 뿐. 프로 뮤지션이 원하는 기능 구현 불가."

2.2 Suno Studio 분석 (2025년 9월 출시)

Suno가 하는 것:

- 타임라인 기반 편집
- 스템 분리 (드럼/베이스/보컬)
- Audio-to-Audio 변환
- 프롬프트 기반 재생성

Suno가 **못하는 것** (= AURA의 기회):

| 기능 | Suno | AURA |

|-----|-----|-----|

| MIDI 편집 | | |

| VST 플러그인 | | |

| 내 목소리 AI 학습 | | |

| 실시간 건반 연주 | | |

| 음표 단위 수정 | (재생성만) | |

3. 타겟 고객 및 포지셔닝

3.1 타겟 고객

Primary Target: 프로 뮤지션 / 세미프로

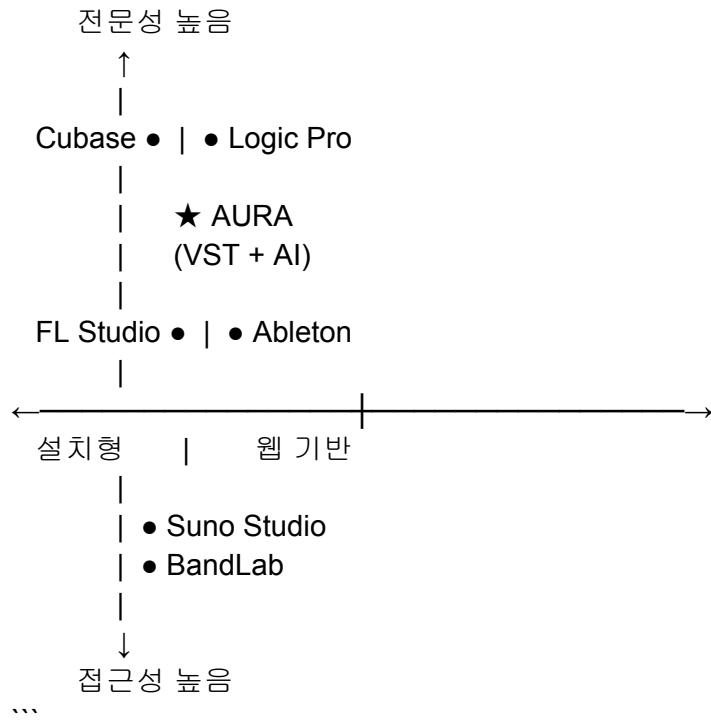
- 이미 **수백~수천만원** 상당의 VST/샘플 라이브러리 보유
- Cubase, Logic, FL Studio 등 기존 DAW 사용자
- AI의 편의성은 원하지만, **자신의 악기/사운드**를 포기 못함
- Suno로 넘어가도 기존 자산을 버릴 수 없음

Secondary Target: 프로 지망생 / 유튜버

- DAW 학습 곡선은 부담스러움
- 하지만 Suno의 "랜덤 뽑기"보다 **통제력** 원함

3.2 포지셔닝 맵

...



3.3 차별화 메시지

> **"Suno는 작곡 비서, AURA는 내 악기"**

Suno	AURA
"다시 뽑아줘" (재생성)	"이 음표 하나만 수정해줘"
감독(Director) 느낌	**연주자(Player) 느낌**
AI가 주는 대로 사용	내 VST + 내 샘플 사용
낯선 AI 목소리	**내 목소리 AI 클론**

4. 핵심 기능 명세

4.1 워크플로우 & UI (3-Way Hybrid Interface)

Beat Mode (FL Studio 스타일)

- 스텝 시퀀서 기반
- 드럼/루프 빠르게 제작
- 패턴 → 타임라인 드래그앤파울

Arrange Mode (Cubase/Logic 스타일)

- 리니어 타임라인
- 정밀 오디오/MIDI 편집
- 오토메이션, 믹싱

Session Mode (Ableton 스타일)

- 클립 런칭
- 라이브 퍼포먼스
- 아이디어 스케치

> **탭 하나로 3가지 모드 전환!**

4.2 프로페셔널 오디오 기능

| 기능 | 설명 |

|-----|-----|

- | **VST3/AU 호스팅** | Serum, Kontakt 등 서드파티 플러그인 로딩 |
- | **ASIO/CoreAudio** | 레이턴시 제로에 도전 |
- | **Advanced MIDI** | MPE 지원, 외부 하드웨어 연동 |
- | **Web MIDI API** | USB 건반 플러그앤플레이 |

4.3 AURA 퀄리 AI 기능

4.3.1 AI Voice Cloning (My Persona) ★

...

내 목소리 1분 녹음

→ AI가 톤/음색 학습

→ 텍스트 또는 흥얼거림으로 코러스/가이드 보컬 생성

→ 내 목소리로 AI가 노래!

...

경쟁 우위: Suno는 랜덤 AI 목소리만, AURA는 **내 목소리** 사용

4.3.2 Generative MIDI (Not Audio) ★

...

Suno: 오디오 파일 출력 (수정 어려움)

AURA: MIDI 노트 출력 (수정 자유)

예: "내 Kontakt 피아노로 재즈 힙합 코드 만들어줘"

- 내 VST가 연주하는 MIDI 노트 생성
- 음표 하나하나 수정 가능!
- ...

4.3.3 Stem Separation & Remix

- MP3/WAV 드래그앤파인
- 드럼/베이스/보컬/기타 즉시 분리
- 각 스템 개별 편집 가능

4.3.4 AI Smart Knob (카멜레온 노브)

트랙 내용에 따라 노브 기능이 자동 변경:

트랙 노브 이름 숨겨진 처리
----- ----- -----
Kick PUNCH Transient + Low EQ + Compression
Snare CRACK High-Mid EQ + Saturation + Gate
HiHat CRISP High-Pass Filter + Exciter
Bass GRIT Tube Saturation + Mid Boost
Piano SPACE Stereo Imager + Hall Reverb
Vocal AIR High-Shelf EQ + De-esser

4.3.5 Ghost Note (고스트 노트)

- 피아노롤에 투명한 가이드 음표 표시
- AI가 다음 멜로디 제안
- 클릭하면 실제 노트로 확정

4.3.6 Kit Morph (키트 모핑)

- 리듬 패턴 유지 + 사운드만 교체
- BPM/패턴 밀도 분석 → 어울리는 키t 자동 추천
- Parameter Randomization으로 매번 조금씩 다른 느낌

5. 기술 스택

5.1 전략적 결정: Python + Pedalboard

왜 C++/JUCE 직접 개발을 포기했는가?

C++ 직접 개발 Python + Pedalboard
----- -----
컴파일 에러 수천 개 저장 → 실행 → 바로 됨
C++ 개발자 채용 필요 (연봉 1억+) **우리 셋이서 가능**
개발 기간 1년+ 빠른 프로토타이핑

| 에러 디버깅 지옥 | AI가 코드 잘 짬 |

Pedalboard란?

...

Spotify가 만든 Python 라이브러리

→ 내부는 JUCE (C++) = 성능 보장

→ 겉은 Python (쉬움) = 개발 속도

= "페라리 엔진에 앉아서 핸들만 돌리는 것"

...

VST 로딩 비교:

```python

# C++ (JUCE 직접): 수백 줄 코드 + 컴파일

# Python (Pedalboard): 1줄

```
from pedalboard import load_plugin
vst = load_plugin("C:/VST/Serum.dll")
```
```

5.2 최종 기술 스택

Application Layer

| 기술 | 용도 |

|-----|-----|

| **Electron** | 크로스플랫폼 데스크탑 앱 (Windows/Mac) |

| **Python** | 백엔드 로직, AI 처리 |

| **React + TypeScript** | UI 프론트엔드 |

Audio Engine

| 기술 | 용도 |

|-----|-----|

| **Pedalboard (Spotify)** | VST 호스팅, 오디오 처리 (내부 JUCE) |

| **PyAudio / sounddevice** | 오디오 I/O |

| **mido / python-rtmidi** | MIDI 입출력 |

AI Engine (On-Device)

| 기술 | 용도 |

|-----|-----|

| **PyTorch / ONNX Runtime** | GPU 가속 AI 추론 |

| **RVC (Retrieval Voice Conversion)** | 목소리 학습/변환 |

| **Demucs** | 스템 분리 |

| **Crepe** | 피치 감지 |

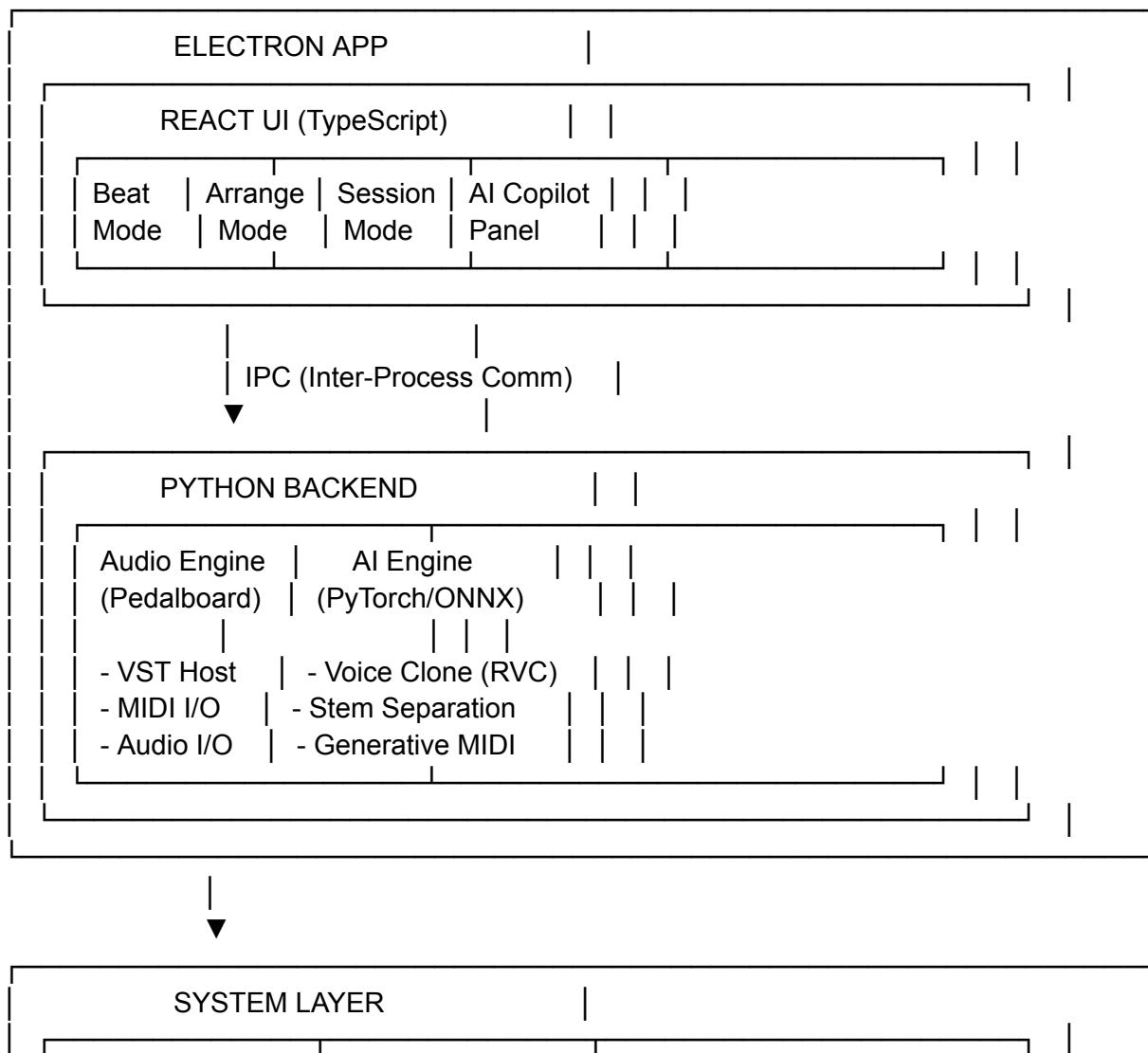
| **Whisper** | 음성 인식 (Voice Command) |

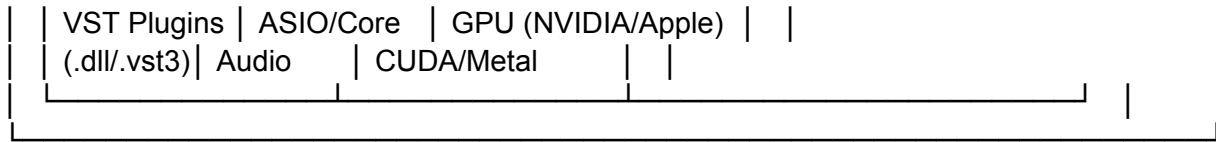
Data & Auth

- | 기술 | 용도 |
- |-----|-----|
- | **SQLite** | 로컬 프로젝트 저장 |
- | **Firebase / Supabase** | 회원가입, 구독 관리 |
- | **Stripe** | 결제 |

6. 아키텍처

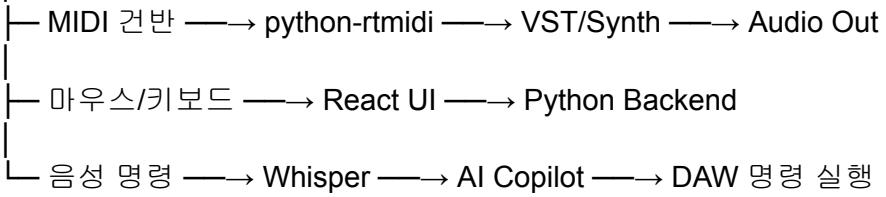
6.1 시스템 아키텍처





6.2 데이터 흐름

[사용자 입력]



7. UI/UX 디자인 시스템

7.1 디자인 컨셉: "Neon Noir"

> 칠혹 같은 어둠 속에서 네온 빛이 발광하는 사이버펑크/퓨처리스틱 스튜디오

7.2 Color Palette

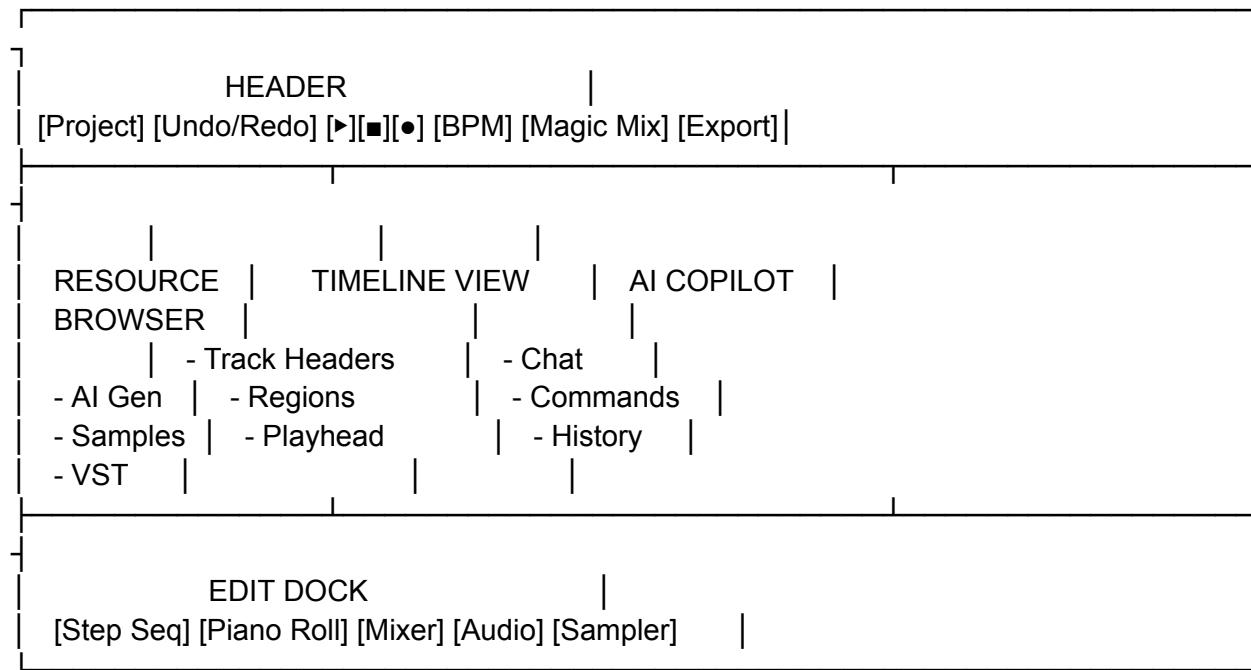
용도	색상	HEX
----- ----- -----		
Global Background	The Void	`#121212`
Panel/Surface	Background	`#1C1C1C`
Primary Accent	Neon Olive	`#B5D948`
Secondary Accent	Neon Cyan	`#4DFFFF`
Alert/Record	Neon Red	`#FF4D4D`
Text Primary	Off-white	`#EEEEEE`
Text Muted	Gray	`#808590`
Grid/Separator	Dark Gray	`#333333`

7.3 UI Style Rules

- **Corner Radius:** 6px ~ 8px
- **Glow Effect:** 주요 버튼 활성화 시 Outer Glow (5px)
- **Waveform:** 배경 없이 Neon Olive 실선

- **MIDI Notes:** Neon Olive → Neon Cyan 그라데이션

7.4 5-Zone Layout



8. 개발 로드맵

Phase 1: MVP (2개월)

목표: "소리가 나는가?"

- | 태스크 | 설명 |
- |-----|-----|
- | Electron 프로젝트 세팅 | Python 백엔드 연동 |
- | Pedalboard 연동 | VST 하나 로딩 테스트 |
- | MIDI 입력 | USB 건반 연결 → 소리 출력 |
- | 기본 Transport | Play/Stop/BPM |

완료 기준: VST 악기를 건반으로 연주하고 소리 날

Phase 2: Hybrid UI (3~4개월)

목표: "작곡이 되는가?"

태스크 설명
----- -----
3-Way View Beat/Arrange/Session 모드 전환
트랙 시스템 오디오/MIDI 트랙 관리
피아노롤 MIDI 노트 편집
믹서 볼륨/팬/Smart Knob
녹음 기능 오디오/MIDI 녹음

완료 기준: 기본적인 곡 제작 가능

Phase 3: AI 기능 (3개월)

목표: "차별화가 되는가?"

태스크 설명
----- -----
Voice Clone RVC 연동, 내 목소리 학습
Generative MIDI AI가 MIDI 노트 생성
Stem Separation Demucs 연동
Ghost Note AI 멜로디 제안
Smart Knob 트랙 분석 → 자동 이펙트

완료 기준: AURA만의 킬러 기능 작동

Phase 4: 상용화 (2개월)

목표: "돈이 벌리는가?"

태스크 설명
----- -----
구독 시스템 Stripe 연동
라이선스 관리 무료/Pro/Ultra 등급
인스톨러 Windows/Mac 배포
베타 테스트 프로 뮤지션 피드백

총 예상 기간: 10~11개월

9. 역할 분담

Project Trinity

역할 담당자 책임
----- ----- -----

CEO	말론 대표	비전, 의사결정, 실행/테스트
CTO	제미나이	아키텍처, 성능 분석, 기술 전략
CPO	클로드	코드 구현, UI/UX, 문서화

업무 흐름

1. 말론 대표: 방향 제시 / 피드백
2. 제미나이: 기술 분석 / 설계
3. 클로드: 코드 작성 / 실행
4. 말론 대표: 테스트 / 결정

10. 비즈니스 모델

10.1 수익 모델: SaaS 구독

플랜	가격	기능
Free Trial	\$0 (14일)	전체 기능 체험
Pro	\$15/월	VST 호스팅, 기본 AI
Ultra	\$35/월	Voice Clone, Stem 분리, 우선 지원

10.2 경쟁사 대비 가격

DAW	가격	방식
Cubase Pro	\$579	일시불
FL Studio	\$499	일시불
Logic Pro	\$199	일시불
AURA	**\$15~35/월**	**구독**

> "일단 한 달만 써볼까?" → 진입장벽 ↓↓

10.3 고객 확보 전략

1. 무료 체험 14일 → 전체 기능 개방
2. 1~2개월 내 매료시킴 (킬러 AI 기능)
3. 구독 전환 → 평생 고객

11. 경쟁사 분석

11.1 vs Suno Studio

항목 Suno Studio AURA
----- ----- -----
타겟 아마추어 **프로 뮤지션**
방식 AI 생성 (재생성) **AI + 직접 편집**
VST X ✓
MIDI 편집 X ✓
내 목소리 X ✓
가격 \$30/월 \$15~35/월

11.2 vs 전통 DAW (Cubase, Logic, FL)

항목 전통 DAW AURA
----- ----- -----
AI 기능 X 없거나 미약 ✓ 핵심
학습 곡선 높음 **낮음 (AI 어시스턴트)**
가격 \$200~600 일시불 **\$15~35/월**
VST ✓ ✓

12. 핵심 원칙

12.1 개발 원칙

1. ***"Suno를 따라가지 않는다"***
- Suno가 못하는 것(VST, MIDI, 내 목소리)에 집중
2. ***"프로의 자산을 존중한다"***
- 기존 VST/샘플 100% 활용 가능하게
3. ***"AI는 대체가 아닌 어시스턴트"***
- 인간이 주도, AI가 보조
4. ***"Python First"***
- C++ 지옥 피하고 빠른 개발

12.2 비즈니스 원칙

1. ***"프로 타겟, 구독 모델"**
- 아마추어 시장은 Suno에게 양보

2. ***"진입장벽 낮추기"**
- 월 구독으로 부담 없이 시작

3. ***"1~2개월 내 매료"**
- 체험 기간 내 킬러 기능으로 락인

12.3 디자인 원칙

1. ***"3대 DAW 장점 막스"**
- FL(비트) + Cubase(편집) + Ableton(라이브)

2. ***"Neon Noir 일관성"**
- 다크 배경 + 네온 액센트

3. ***"Progressive Disclosure"**
- 처음엔 단순, 파고들면 프로 기능

🔗 부록

A. 참고 문서

- `AURA_기술_스택_합의서.pdf`
- `AURA_Cloud_Studio_시드투자_제안서.pdf`
- `AURA_V3_스텝시퀀서.pdf`
- `AURA_V3_믹서_AI_Smart_Knob_기능_명세서.pdf`
- `AURA_V3_기능_명세서_Ghost_Note.pdf`
- `Kit_Morph_키트_모핑_기능.pdf`

B. 개발 환경 세팅

```
```bash
1. Python 설치 (3.10+)
2. 가상환경 생성
python -m venv aura_env
source aura_env/bin/activate # Mac/Linux
aura_env\Scripts\activate # Windows

3. 필수 패키지 설치
pip install pedalboard
```

```
pip install numpy
pip install sounddevice
pip install mido python-rtmidi
```

```
4. AI 패키지 (나중에)
pip install torch
pip install onnxruntime
...
```

## ## C. 용어 정의

용어   정의
----- -----
DAW   Digital Audio Workstation
VST   Virtual Studio Technology (가상 악기/이펙터)
MIDI   Musical Instrument Digital Interface
Stem   개별 분리된 오디오 트랙
RVC   Retrieval Voice Conversion (음성 변환)
ASIO   Audio Stream Input/Output (저지연 드라이버)

---

\*\*문서 끝\*\*

\*이 문서는 AURA 프로젝트의 모든 개발 방향을 정의합니다.\*

\*새로운 대화창에서도 이 문서를 참조하여 일관된 개발을 진행하세요.\*