

# [PROJECT BRIEFING] AURA TRINITY DASHBOARD (기획 및 개념 설명서)

## 1. 프로젝트 정의 (Concept Definition)

본 프로젝트는 '**AURA CLOUD DAW**' 본체와는 별도로 운용되는 독립적인 관제 시스템이다. CEO(User)가 PC 앞에 없을 때나 이동 중(Mobile)일 때도, \*\*Local AI(Antigravity)\*\*와 \*\*Cloud AI(Gemini)\*\*의 협업 상황을 실시간으로 확인하고 지시를 내릴 수 있는 \*\*하이브리드 커뮤니케이션 허브\*\*를 구축하는 것을 목표로 한다.

## 2. 작업 위치 및 환경 (Location & Environment)

우리는 기존 프로젝트(**Node.js/Electron**)와 기술적 충돌을 방지하기 위해, 완전히 분리된 새로운 영토에서 작업을 시작한다.

- 현재 위치 (**Root Directory**): `C:\Users\hmyks\Desktop\aura-trinity-board`
- 운영 체제: Windows 11 (Local PC)
- 기술 스택:
  - **Core:** Python (단일 언어 사용)
  - **Framework:** Streamlit (PC/Mobile 반응형 웹 앱 구현)
  - **Database:** GitHub Gist (기존의 NEXUS 소통 채널 데이터 연동)

## 3. 개발 목표: "One Source, Multi Use" (Target Architecture)

우리는 PC용과 모바일용 프로그램을 따로 만들지 않는다. Streamlit의 반응형 레이아웃 기능을 활용하여, \*\*단 하나의 코드(`app.py`)\*\*로 두 가지 환경을 모두 대응한다.

- **PC 접속 시:** 넓은 화면을 활용한 **[좌측 로그 | 우측 채팅]** 2단 분할 레이아웃.
- **Mobile 접속 시:** 세로 화면에 최적화된 **[상단 로그 | 하단 채팅]** 1단 스크롤 레이아웃.

## 4. 시스템 작동 원리 (Mechanism)

이 대시보드는 \*\*'보는 눈(Viewer)'\*\*과 '말하는 입(Chat)' 역할만 수행한다.

1. **Monitor (Nexus View):** 안티그래비티가 Gist에 기록한 작업 로그를 주기적으로 읽어와(Read) 시각적으로 보여준다.
2. **Command (Chat Interface):** CEO가 내린 지시 사항을 Gemini에게 전달하거나, 안티그래비티가 확인할 수 있도록 기록한다.

## ⚠️ 개발자 유의사항 (Notice)

- 이 문서는 프로젝트의 \*\*\*설계도(Blueprint)\*\*\*이자 \*\*\*나침반\*\*\*이다.
- 현재 폴더(**aura-trinity-board**)는 이 목적을 위해 CEO가 미리 마련해둔 \*\*청정 구역(Clean Zone)\*\*임을 인지하라.
- **[대기 모드]:** 본 기획 내용을 충분히 숙지하고, 추후 CEO의 구체적인 \*\*\*파일 생성 및 코딩 지시\*\*\*가 내려올 때까지 시스템을 임의로 변경하지 않고 대기한다.