

Trang 6.1: Tìm hiểu về file quy tắc GEMINI.md và cách tùy chỉnh hành vi AI.

1. GEMINI.md là gì?

GEMINI.md là file quan trọng nhất trong toàn bộ hệ thống Antigravity Kit. Nó nằm tại:

```
.agent/rules/GEMINI.md
```

File này là "**hiến pháp**" định nghĩa cách AI phải hành xử trong **mọi** tình huống. Khi AI đọc file này, nó hiểu rõ:

- **Ưu tiên nào cao hơn** (quy tắc nào được áp dụng khi có xung đột).
- **Phân loại yêu cầu** như thế nào (câu hỏi, code đơn giản, task phức tạp).
- **Chọn Agent** nào cho từng loại yêu cầu.
- **Các quy tắc code toàn cục** phải tuân thủ.

2. Kiến trúc phân cấp ưu tiên

P0: GEMINI.md ← Cao nhất — luôn có hiệu lực

P1: Agent .md ← Quy tắc riêng của từng Agent

P2: SKILL.md ← Kiến thức chuyên sâu từng lĩnh vực

Khi có xung đột giữa các cấp, cấp cao hơn luôn thắng.

3. Cấu trúc của GEMINI.md

File được tổ chức theo hệ thống **TIER** (Tầng):

TIER 0 — Universal Rules (Luôn áp dụng)

Các quy tắc này hoạt động trong **mọi** yêu cầu, bất kể Agent nào được kích hoạt:

Quy tắc	Nội dung
Language Handling	Khi người dùng không dùng tiếng Anh, AI dịch nội bộ nhưng trả lời bằng ngôn ngữ của người dùng
Clean Code	Mọi code phải theo tiêu chuẩn <code>clean-code</code> — ngắn gọn, tự mô tả, không over-engineer
File Dependency Awareness	Trước khi sửa file nào, kiểm tra file phụ thuộc và cập nhật đồng bộ
System Map Read	Đọc ARCHITECTURE.md khi khởi động để hiểu hệ sinh thái

Request Classifier — Phân loại yêu cầu

Bảng phân loại tự động để AI biết cần làm gì:

Loại yêu cầu	Từ khóa kích hoạt	Kết quả
QUESTION	"là gì", "giải thích", "tại sao"	Chỉ trả lời văn bản
SURVEY	"phân tích", "liệt kê", "tổng quan"	Phân tích, không tạo file
SIMPLE CODE	"sửa", "thêm", "đổi" (1 file)	Chỉnh sửa trực tiếp
COMPLEX CODE	"xây dựng", "tạo", "triển khai"	Tạo file kế hoạch .md
DESIGN/UI	"thiết kế", "UI", "giao diện"	Tạo file kế hoạch + components

Intelligent Routing Checklist

Checklist bắt buộc AI phải hoàn thành trước khi viết code:

- Bước 1: Đã xác định Agent phù hợp chưa?
- Bước 2: Đã đọc file .md của Agent chưa?
- Bước 3: Đã thông báo Agent cho người dùng chưa?
- Bước 4: Đã tải Skills cần thiết chưa?

TIER 1 — Code Rules (Khi viết code)

Chỉ áp dụng khi AI cần viết hoặc sửa code:

Quy tắc	Nội dung
Project Type Routing	Mobile → <code>mobile-developer</code> , Web → <code>frontend-specialist</code> , API → <code>backend-specialist</code>
Socratic Gate	Với tác vụ phức tạp: bắt buộc hỏi ít nhất 3 câu làm rõ trước khi code
Final Checklist Protocol	Khi người dùng nói "final checks", chạy <code>checklist.py</code>

4. Tùy chỉnh `GEMINI.md` cho đội ngũ của bạn

Nguyên tắc tùy chỉnh an toàn

□ Có thể sửa:

- Thêm quy tắc mới vào cuối file.
- Thêm các ràng buộc về ngôn ngữ lập trình hoặc naming convention.
- Định nghĩa các pattern bị cấm trong dự án.

⚠ Cẩn thận khi sửa:

- Bảng Request Classifier (có thể làm AI nhận diện sai loại yêu cầu).
- Agent Routing rules (có thể gây định tuyến sai Agent).

□ Không nên xóa:

- Cấu trúc TIER 0 và TIER 1 cơ bản.
- Intelligent Routing Checklist.

Ví dụ tùy chỉnh thực tế

Thêm quy tắc ngôn ngữ:

markdown

CUSTOM TEAM RULES

Quy tắc ngôn ngữ

- Tất cả comment trong code phải bằng tiếng Anh
- Tên biến và hàm theo camelCase (không dùng snake_case trong JS/TS)
- Không dùng `any` trong TypeScript — phải có type cụ thể

Thêm pattern bị cấm:

markdown

Các pattern bị cấm trong dự án này

- KHÔNG dùng `console.log` trong production code (dùng logger service)
- KHÔNG hard-code URL — phải dùng environment variables
- KHÔNG commit secret key hoặc password
- KHÔNG viết SQL raw — phải dùng Prisma ORM

Thêm convention đặt tên:

markdown

Naming Convention

- Component: PascalCase (UserCard, ProductList)
- Hook: camelCase bắt đầu bằng "use" (useAuth, useCart)
- File component: PascalCase.tsx
- File util: camelCase.ts
- Test file: tên-file.test.ts

Thêm quy tắc về Tech Stack:

markdown

Tech Stack bắt buộc cho dự án này

- Framework: Next.js 14 (App Router) — không dùng Pages Router
- Styling: Tailwind CSS — không dùng CSS Modules hay styled-components
- Database: Prisma + PostgreSQL — không dùng raw SQL
- Auth: NextAuth.js — không tự viết auth logic

5. Chia sẻ cấu hình với cả đội

Để toàn đội cùng dùng chung một bộ quy tắc:

```
bash
# Commit thư mục .agent/ vào Git
git add .agent/
git commit -m "chore: add/update Antigravity Kit team rules"
git push
```

Khi thành viên mới clone dự án, họ sẽ tự động có đầy đủ cấu hình đã được thống nhất.

6. Kiểm tra quy tắc có được áp dụng không

Sau khi sửa `GEMINI.md`, bạn có thể kiểm tra bằng cách:

☐☐ Bạn: "Hãy kiểm tra và cho tôi biết các quy tắc nào bạn đang áp dụng cho dự án này"

☐☐ AI: Tôi đang áp dụng các quy tắc từ `.agent/rules/GEMINI.md`:

- TIER 0: Universal Rules (Language, Clean Code...)
- Custom Team Rules: No console.log, PascalCase components...
- Tech Stack: Next.js 14 App Router, Tailwind CSS...

Phiên bản #2

Được tạo 2026-03-04 06:13:02 UTC bởi Nam Đặng

Được cập nhật 2026-03-04 10:05:03 UTC bởi Nam Đặng