

# Trang 3.2: Skills - Hệ thống kỹ năng và kiến thức chuyên sâu cho AI.

## 1. Skill là gì?

**Skill** là một gói kiến thức chuyên biệt chứa các nguyên lý, mẫu thiết kế (patterns) và khung ra quyết định (decision-making framework) cho một lĩnh vực cụ thể.

## Skills khác gì so với template code thông thường?

Đặc điểm	Template Code Thông thường	Skill trong Antigravity Kit
<b>Bản chất</b>	Code mẫu cứng nhắc để copy	Nguyên lý để suy nghĩ và quyết định
<b>Tính linh hoạt</b>	Cố định, áp sát một pattern	Thích nghi theo ngữ cảnh dự án
<b>Kết quả</b>	Code giống nhau cho mọi trường hợp	Code phù hợp với yêu cầu cụ thể

“**Ví dụ:** Thay vì đưa cho bạn một template API cố định, Skill `api-patterns` dạy AI *cách quyết định* khi nào dùng REST, khi nào dùng GraphQL, và khi nào dùng tRPC dựa trên ngữ cảnh dự án của bạn.

## 2. Cách Skills được tải (Load on-demand)

Skills **không** được tải toàn bộ một lúc. Chúng được tải **theo yêu cầu** theo 3 cơ chế:

1. **Cấu hình Agent:** Mỗi Agent trong YAML frontmatter liệt kê các Skills nó được phép truy cập (trường `skills:`).
2. **Ngữ cảnh tác vụ:** Khi nhận yêu cầu, AI đọc mô tả của các Skills có liên quan và chỉ tải những Skill thực sự cần thiết.
3. **Đọc có chọn lọc:** Chỉ đọc các section cần thiết trong **SKILL.md**, không đọc toàn bộ để tối ưu hiệu suất.

Ví dụ:

Yêu cầu: "Tạo API endpoint cho giỏ hàng"

|



backend-specialist được kích hoạt

|



Tải: api-patterns + nodejs-best-practices + database-design

|



Không tải: mobile-design, seo-fundamentals, red-team-tactics...

## 3. Cấu trúc một Skill

.agent/skills/react-best-practices/

├ SKILL.md # Bắt buộc — Metadata, nguyên lý chính, khung quyết định

├ sections/ # Tùy chọn — Hướng dẫn chi tiết theo từng chủ đề nhỏ

├ examples/ # Tùy chọn — Ví dụ triển khai tham khảo thực tế

├ scripts/ # Tùy chọn — Script Python/Bash có thể chạy tự động

└ assets/ # Tùy chọn — Hình ảnh, assets tham khảo

**Phân đầu của**

**SKILL.md (Frontmatter YAML):**

```
yaml
---
name: react-best-practices
description: React & Next.js performance optimization from Vercel Engineering.
allowed-tools: Read, Write, Edit, Bash
---
```

## 4. Danh sách đầy đủ 36 Skills theo nhóm

### 📄 Frontend & UI (5 Skills)

Skill	Mô tả
<code>react-best-practices</code>	Tối ưu React & Next.js — 57 quy tắc từ Vercel Engineering
<code>web-design-guidelines</code>	Kiểm toán UI web — 100+ quy tắc về Accessibility, UX, Performance
<code>tailwind-patterns</code>	Tailwind CSS v4, CSS-first config, container queries
<code>frontend-design</code>	Hệ thống thiết kế, bảng màu, typography, micro-animations
<code>ui-ux-pro-max</code>	<b>50 phong cách</b> thiết kế, <b>21 bảng màu</b> , <b>50 font chữ</b> chuyên nghiệp

### ⚙️ Backend & API (4 Skills)

Skill	Mô tả
<code>api-patterns</code>	Thiết kế API: REST vs GraphQL vs tRPC, versioning, pagination
<code>nestjs-expert</code>	NestJS modules, DI container, decorators, Guards
<code>nodejs-best-practices</code>	Node.js async patterns, bảo mật, kiến trúc module

Skill	Mô tả
python-patterns	FastAPI, Django, async vs sync, type hints, Pydantic v2

## 📦 Database (2 Skills)

Skill	Mô tả
database-design	Thiết kế schema, chiến lược indexing, tránh N+1 query
prisma-expert	Prisma ORM, migrations, relations, query optimization

## 📦 TypeScript / JavaScript (1 Skill)

Skill	Mô tả
typescript-expert	Lập trình kiểu nâng cao (generics, conditional types), hiệu năng biên dịch

## ☁️ Cloud & Infrastructure (3 Skills)

Skill	Mô tả
docker-expert	Containerization, Docker Compose, multi-stage builds
deployment-procedures	CI/CD, chiến lược deploy 5 giai đoạn, rollback
server-management	Quản lý process, monitoring, scaling

## 📦 Testing & Quality (5 Skills)

Skill	Mô tả
testing-patterns	Pyramid kiểm thử, Jest/Vitest, AAA pattern, mocking
webapp-testing	E2E testing, Playwright, audit chiến lược sâu
tdd-workflow	Vòng lặp RED-GREEN-REFACTOR, test-first development
code-review-checklist	Tiêu chuẩn review code: chất lượng, bảo mật, best practice
lint-and-validate	ESLint, Prettier, TypeScript strict mode

## ☐☐ Security (2 Skills)

Skill	Mô tả
vulnerability-scanner	OWASP 2025, Supply Chain Security, phân tích lỗ hổng
red-team-tactics	Chiến thuật tấn công theo MITRE ATT&CK, báo cáo penetration test

## ☐☐ Architecture & Planning (4 Skills)

Skill	Mô tả
app-builder	Tạo khung ứng dụng full-stack, chọn tech stack, phối hợp Agent
architecture	ADR (Architecture Decision Records), đánh đổi trade-off
plan-writing	Phân rã công việc, định nghĩa phụ thuộc, tiêu chí xác minh
brainstorming	Giao thức Socratic: đặt câu hỏi làm rõ trước khi triển khai

## ☐☐ Mobile (1 Skill)

Skill	Mô tả
mobile-design	Touch interaction, Mobile-first, iOS/Android conventions

## ☐☐ Game Development (1 Skill)

Skill	Mô tả
game-development	Logic game, cơ học vật lý, vòng lặp game loop

## ☐☐ SEO & Growth (2 Skills)

Skill	Mô tả
seo-fundamentals	E-E-A-T, Core Web Vitals, Google algorithm principles

Skill	Mô tả
geo-fundamentals	Generative Engine Optimization (tối ưu cho AI search: ChatGPT, Perplexity)

## ☐☐ Shell / CLI (2 Skills)

Skill	Mô tả
bash-linux	Linux commands, piping, error handling, scripting
powershell-windows	PowerShell operators, quản lý lỗi, pitfalls

## ☐☐ Other (6 Skills)

Skill	Mô tả
clean-code	Tiêu chuẩn code sạch toàn cục, không over-engineering
behavioral-modes	Các chế độ hoạt động AI (brainstorm, implement, debug, ship...)
parallel-agents	Mẫu điều phối đa Agent, phân chia công việc độc lập
mcp-builder	Xây dựng MCP Server, thiết kế tool & resource patterns
documentation-templates	Cấu trúc README, API docs, comment code
i18n-localization	Phát hiện hardcoded string, quản lý translations, RTL
performance-profiling	Core Web Vitals, đo lường, bundle analysis
systematic-debugging	Quy trình 4 giai đoạn: Discovery → Hypothesis → Verification → Resolution
rust-pro	Rust 1.75+, async/await, Tokio, axum, systems programming

# 5. Skills đặc biệt kèm theo công cụ thực thi

Các Skills sau không chỉ cung cấp kiến thức mà còn có **scripts Python** để Agent có thể tự động kiểm tra:

Skill	Script	Tác dụng
vulnerability-scanner	security_scan.py	Quét lỗ hổng bảo mật trong dependencies
lint-and-validate	lint_runner.py	Chạy ESLint, kiểm tra type errors
database-design	schema_validator.py	Kiểm tra tính nhất quán schema DB
testing-patterns	test_runner.py	Chạy bộ test tự động (Jest/Vitest)
frontend-design	ux_audit.py , accessibility_checker.py	Kiểm tra UX Laws và WCAG Accessibility
seo-fundamentals	seo_checker.py	Kiểm tra meta tags, cấu trúc heading
performance-profiling	lighthouse_audit.py , bundle_analyzer.py	Đo lường Lighthouse score và bundle size
webapp-testing	playwright_runner.py	Chạy E2E test với Playwright
mobile-design	mobile_audit.py	Kiểm tra responsive và touch targets

## 6. Cách Skills được tích hợp với Agents (ví dụ thực tế)

frontend-specialist.md:

---

skills: react-best-practices, frontend-design, tailwind-patterns, web-design-guidelines

---

Khi yêu cầu: "Tạo component card sản phẩm đẹp và dễ tiếp cận"

|

Agent tải:

├─ react-best-practices → Áp dụng Server Components, tránh re-render thừa

├─ frontend-design → Chọn bảng màu, spacing hợp lý

├─ tailwind-patterns → Viết class Tailwind đúng chuẩn v4

└─ web-design-guidelines → Đảm bảo ARIA labels, contrast ratio

Phiên bản #2

Được tạo 2026-03-04 06:10:36 UTC bởi Nam Đặng

Được cập nhật 2026-03-04 07:17:56 UTC bởi Nam Đặng