



Golden Rule Before Building Any AI Application

"Before building an application, define the rules of intelligence."

AI agents are powerful — but without structure, they become chaotic.

So before writing even 1 line of code, we define **5 foundational documents**.

1 Agent.md – Brain of the Application

This file defines:

- 🤖 Agent Role
- 🎯 Goals
- 🧠 Reasoning Style
- 🚫 Guardrails
- 🛠️ Tools Allowed

EXAMPLE STRUCTURE

```
# Agent: Antigravity App Builder

## Role
You are a senior full-stack engineer.

## Rules
- Always create scalable architecture.
- Prefer clean folder structure.
- Ask for clarification before assumptions.

## Restrictions
- No hardcoded secrets.
- No deprecated libraries.
```

- 👉 This prevents hallucination
- 👉 Makes AI predictable
- 👉 Makes output production-grade

2 README.md – Human Contract

This is for developers. Contains:

- Project overview
- Tech stack
- Setup steps
- Architecture diagram
- How to run locally
- Deployment steps

This makes onboarding easy.

3 SKILLS.md – Agent Capability Memory 🧠

This is your powerful idea 🙌 Whenever the agent uses a skill while building:

- JWT Authentication
- Stripe Payment Integration
- GraphQL
- Docker
- Redis Caching
- Vector DB
- CI/CD

It must automatically append it to SKILLS.md.

Why This Is Genius?

Because:

- You can audit what AI actually knows.
- You track capability growth.
- You avoid repeating learning.
- It becomes an evolving intelligence log.

EXAMPLE

```
# Skills Used in This Project
- Spring Boot 3
- PostgreSQL
- JWT Authentication
- Dockerization
- REST API Design
- OpenAPI Documentation
```

4 CONTRIBUTING.md – Collaboration Rules

When multiple students contribute:

- Code standards
- PR rules
- Branch naming conventions
- Commit message format

- Testing requirements

EXAMPLE

```
## Contribution Guidelines
- Use feature branches.
- Follow Clean Code principles.
- Add unit tests.
- Follow conventional commits.
```

This avoids chaos in teams.

5 Proper Planning Before Execution (Most Important 🔥)

Before building any app, the agent MUST:

Step 1: Requirement Understanding

- Clarify objective
- Define scope
- Identify users
- Define constraints

Step 2: Architecture Plan

- High-level design
- Tech stack decision
- DB schema idea
- APIs required

Step 3: Task Breakdown

- Backend tasks
- Frontend tasks
- DevOps tasks
- Security tasks
- Testing tasks

Step 4: Execution Flow Plan

EXAMPLE

1. Setup project structure
2. Configure database
3. Implement authentication
4. Build core APIs
5. Integrate frontend
6. Add validation
7. Add logging
8. Write tests
9. Dockerize
10. Deploy

CRITICAL RULE: Ask Permission Before Building

Before writing code, agent should say:

"Here is the architecture plan and execution flow. Do you approve this plan? Should I proceed?"

This prevents:

- Wrong implementation
- Wasted effort
- Misaligned requirements

This teaches students: AI should assist — not assume.

Golden Rules Before Building Any AI Application — javathoughts.com