AI-Driven Development — 30-Day Challenge

Part A — Theory (Short Questions)

Answer in your own wording

1. **Nine Pillars Understanding**

   Why is using AI Development Agents (like Gemini CLI) for repetitive setup tasks better for your growth as a system architect?

   Using AI agents like Gemini CLI handles repetitive tasks, so you can focus on system design and big-picture thinking

   Explain how the Nine Pillars of AIDD help a developer grow into an M-Shaped Developer.

   The Nine Pillars train you in multiple areas, turning you into an M-Shaped Developer with both depth and breadth.

2. **Vibe Coding vs Specification-Driven Development**

   Why does Vibe Coding usually create problems after one week?

   Vibe Coding creates messy, buggy code after a week because it lacks planning

   How would Specification-Driven Development prevent those problems?

   Specification-Driven Development sets clear instructions upfront, keeping code clean, maintainable, and predictable.

3. **Architecture Thinking**

   How does architecture-first thinking change the role of a developer in AIDD?

   Architecture-first thinking makes you a system designer, not just a coder

Explain why developers must think in layers and systems instead of raw code.

Thinking in layers ensures software is organized, scalable, and works effectively with AI tools.

Part B — Practical Task (Screenshot Required)

Task:

Using any AI CLI tool, generate a 1-paragraph specification for an email validation function.

Requirements:

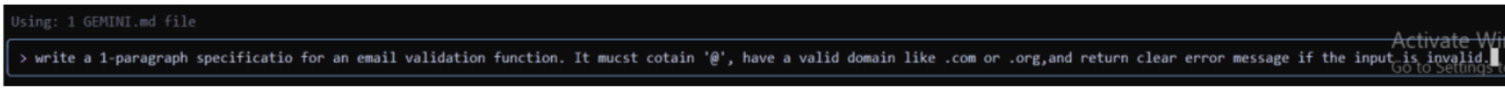
Must contain “@”

Must contain a valid domain (e.g., .com, .org)

Should return clear error messages

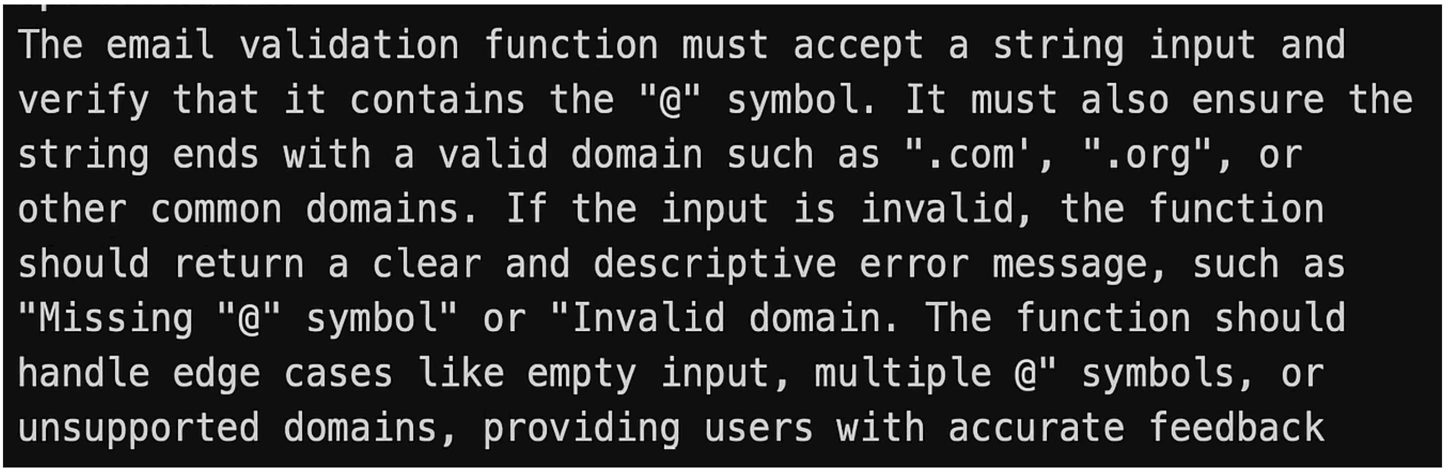
Submission:

1️⃣Your exact CLI prompt (text or screenshot)



```
Using: 1 GEMINI.md file
> write a 1-paragraph specificatio for an email validation function. It mucst cotain '@', have a valid domain like .com or .org,and return clear error message if the input is invalid.
```

2️⃣The 1-paragraph specification generated by the CLI



```
The email validation function must accept a string input and
verify that it contains the "@" symbol. It must also ensure the
string ends with a valid domain such as ".com', ".org", or
other common domains. If the input is invalid, the function
should return a clear and descriptive error message, such as
"Missing "@" symbol" or "Invalid domain. The function should
handle edge cases like empty input, multiple @" symbols, or
unsupported domains, providing users with accurate feedback
```

Part C — Multiple Choice Questions

1. **What is the main purpose of Spec-Driven Development?**

   A. Make coding faster

   B. Clear requirements before coding begins

   C. Remove developers

   D. Avoid documentation

2. **What is the biggest mindset shift in AI-Driven Development?**

   A. Writing more code manually

   B. Thinking in systems and clear instructions

   C. Memorizing more syntax

   D. Working without any tools

3. **Biggest failure of Vibe Coding?**

   A. AI stops responding

   B. Architecture becomes hard to extend

   C. Code runs slow

   D. Fewer comments written

4. **Main advantage of using AI CLI agents (like Gemini CLI)?**

   A. They replace the developer completely

   B. Handle repetitive tasks so dev focuses on design & problem-solving

   C. Make coding faster but less reliable

   D. Make coding optional

5. **What defines an M-Shaped Developer?**

A. Knows little about everything

B. Deep in only one field

C. Deep skills in multiple related domains

D. Works without AI tools