🌐 AI-Driven Development — 30-Day Challenge

📝 **Task 2**
🧍 Name: Zoya Afzal
📍 Class Slot: Friday — 6:00 PM to 9:00 PM
👨‍🏫 Instructor: Sir Hamzah Syed

📁Part A — Theory (Short Questions)

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?

The Nine Pillars of AIDD help a developer learn different parts of software development. AI tools handle hard or confusing tasks, so the developer can understand more areas like design, testing, planning, and documentation. Over time, they become skilled in several fields, not just one. That is how the Nine Pillars help someone become an M-shaped developer: a person with strong skills in multiple areas.

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

The Nine Pillars of AIDD provide a complete system that supports a developer at every step, combining specs, testing, AI tools, agents, documentation, and workflows. This lets one developer handle tasks usually done by a whole team while learning new skills. Over time, they gain experience in design, testing, planning, integration, and operations, building expertise across multiple areas. That's why the Nine Pillars help a developer become M-shaped, with deep knowledge in several fields.

2. Vibe Coding vs Specification-Driven Development

Why does Vibe Coding usually create problems after one week?

Vibe coding is the practice of writing code based purely on intuition without planning, specifications, or clear requirements. In the first week, this often feels productive because you are building fast. But after some days, the problems begin to appear. Without notes or documentation, it's hard to remember why things were done, so fixing or adding features gets tricky. Every new feature you add starts breaking something else, because there is no structure holding the system together.

How would Specification-Driven Development prevent those problems?
Specification-Driven Development (SDD) avoids the problems of vibe coding by planning everything before writing code. You write clear rules for what the system should do and how it should work. These rules act as a guide, making the code organized and easy to understand. It reduces bugs, saves time, and makes the system easy to maintain and add to. That's why many companies use SDD for serious projects.

3. Architecture Thinking

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

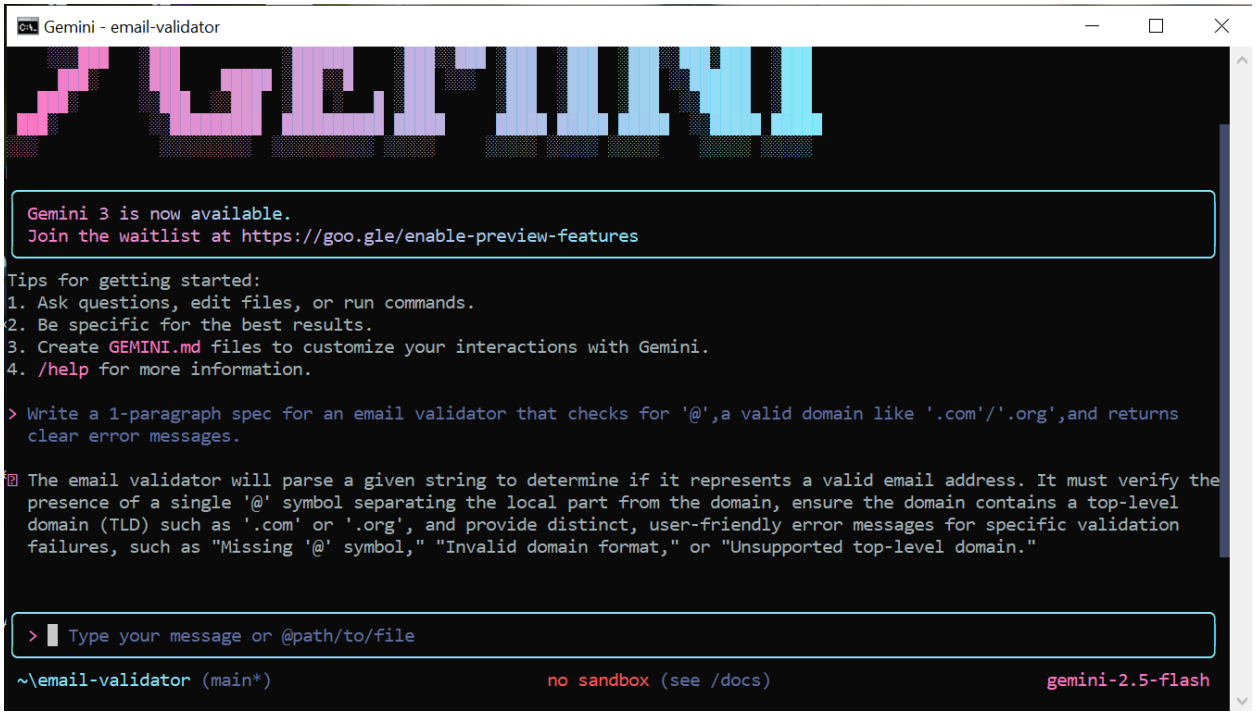
Architecture-first thinking changes a developer's role from coding to designing systems. AI can write code, but the developer decides how the system is structured and how parts work together. They guide AI with clear plans and focus on making the system organized and scalable. This lets the developer take a bigger, more strategic role while AI handles repetitive tasks.

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

In AI-driven development, the challenge is in the system, not the code. Thinking in layers helps developers see where models, tools, and agents

fit, keeping everything organized. Focusing only on code hides how parts interact and how data flows. Layered, systems thinking makes software easier to manage, scale, and maintain, and helps design complex AI systems more effectively.

📁 Part B — Practical Task (Screenshot Required)



📁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

✔️ Correct Answer: B. Clear requirements before coding begins

---

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

✔️ **Correct Answer: B. Thinking in systems and clear instructions**

---

3. Biggest failure of Vibe Coding?

A. AI stops responding
B. Architecture becomes hard to extend
C. Code runs slow
D. Fewer comments written

✔️ **Correct Answer: B. Architecture becomes hard to extend**

---

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

✔️ **Correct Answer: B. Handle repetitive tasks so dev focuses on design & problem-solving**

---

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

✔️ **Correct Answer: C. Deep skills in multiple related domains**

🌟 Reflection

This task helped me understand how quickly the developer's role is changing in the AI-Native world. Ideas like the Nine Pillars, SDD, and Development Agents aren't just extra tools—they show a completely new way of planning and building software. By focusing on architecture first and learning how to work alongside AI agents, I can move beyond just writing code and start thinking at a full system level. This approach helps developers grow into M-Shaped professionals with strong skills in multiple areas, supported by smart AI tools.