

Task-02

AI-Driven Development-30-Days Challenge

Name : Muhammad Rizwan.

Slot : Friday (6:00 to 9:00).

Task by : Asma Yaseen.

Insructor : Sir Hamza Syed.

Part A Theory (Short Questions)

1. Nine Pillars Understanding

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

Using AI Development Agents for repetitive setup work frees you from spending time on low value tasks. Instead of manually creating folders, configs, or boilerplate, the agent handles all of that automatically. This allows you to focus on the *architecture*, logic, and design decisions skills that actually help you grow as a system architect. It makes your time more valuable because you spend it thinking about systems, not clicking buttons.

b. How do the Nine Pillars of AIDD help a developer grow into an M-Shaped Developer?

The Nine Pillars guide a developer to master several connected areas spec writing, architecture, evaluation, collaboration with agents, testing, and iteration. By practicing all these pillars, a developer becomes strong in multiple deep skill-sets, not just coding. This combination of depth across several domains is what forms an **M-Shaped Developer**.

2. Vibe Coding vs Specification Driven Development

a. Why does Vibe Coding usually create problems after one week?

Vibe Coding depends on “just trying things” and chatting with the AI without clear direction. After a week, the code becomes messy, hard to extend, inconsistent, and full of hidden decisions that no one documented. This makes debugging and scaling extremely painful.

b. How would Specification-Driven Development prevent those problems?

Specification-Driven Development forces the developer to define clear requirements before coding begins. Because the AI follows a precise spec, the code stays consistent, predictable, and easy to maintain. The whole system becomes easier to extend because decisions were made upfront instead of randomly during coding.

3. Architecture Thinking

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

Architecture-first thinking shifts the developer's role from "writing code" to "designing systems." Instead of jumping straight into implementation, the developer decides how components should work together, how data should flow, and how agents will handle tasks. This makes the developer more like a system planner than a traditional coder.

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

Modern AI-powered software is too complex to understand through raw code alone. Developers need to think in layers UI, logic, storage, services, agents so they can design flexible and maintainable systems. Layered thinking helps prevent chaos, reduces duplication, and keeps the architecture scalable.

Part B Practical Task



```
Windows PowerShell
3. Create GEMINI.md files to customize your interactions with Gemini.
4. /help for more information.

Gemini CLI update available! 0.13.0 -> 0.15.4
Installed with npm. Attempting to automatically update now...

Update successful! The new version will be used on your next run.

> GEMINI

Tips for getting started:
1. Ask questions, edit files, or run commands.
2. Be specific for the best results.
3. Create GEMINI.md files to customize your interactions with Gemini.
4. /help for more information.

Gemini CLI update available! 0.13.0 -> 0.15.4
Installed with npm. Attempting to automatically update now...

Update successful! The new version will be used on your next run.

> Generate a one paragraph specification for an email validation function. Requirements: must contain "@", must include a valid domain such as
.com or .org, and must return clear error message.

E:\30-days-challenge\task-02 no sandbox (see /docs) auto
```

1 CLI Prompt Used

Generate a one paragraph specification for an email validation function. Requirements: must contain '@', must contain a valid domain such as .com or .org, and it must return clear error messages.

2 One Paragraph Specification Generated

The email validation function should accept an input string and check whether it contains a single “@” symbol, followed by a properly structured domain ending with a standard extension such as .com or .org. The function must detect missing elements such as an absent “@”, an invalid domain format, or unsupported extensions and return clear, human readable error messages indicating the exact issue. If all conditions are met, the function should return a confirmation that the email is valid.

Part C Multiple Choice Questions

1. **B. Clear requirements before coding begins**
2. **B. Thinking in systems and clear instructions**
3. **B. Architecture becomes hard to extend**
4. **B. Handle repetitive tasks so dev focuses on design & problem solving**
5. **C. Deep skills in multiple related domains**

Reflection

This assignment shows how AI-Native development changes the developer’s responsibilities. Instead of focusing on writing code line by line, modern developers work with AI agents, create precise specifications, and think about architecture before touching implementation. Techniques like SDD and the Nine Pillars help developers grow into M-Shaped professionals people who have deep skills across several areas and can work smoothly with AI tools. By practicing these habits early, developers prepare themselves for the future of software development, where system thinking and intelligent collaboration matter more than manual coding.