

# □ AI-DRIVEN DEVELOPMENT – 30-DAY CHALLENGE

## TASK 2 – OFFICIAL SUBMISSION

---

### PART A – THEORY

#### 1. Nine Pillars Understanding

##### (A) Why using AI Development Agents (like Gemini CLI) is better for growth as a system architect?

AI Development Agents repetitive tasks jaise setup, configuration, folders aur boilerplate code automatically complete kar dete hain. Is se developer ka waqt bach jata hai aur wo system design, planning aur architecture wali soch develop karta hai. Yeh approach developer ko high-level, structured aur architect-style decision making ki taraf le jati hai.

##### (B) How do the Nine Pillars help a developer become an M-Shaped Developer?

Nine Pillars developer ko ek hi skill tak limited nahi rehnay dete. Yeh pillars specification writing, testing, automation, observability, documentation, agents aur architecture jaisi multiple domains cover karte hain. Har pillar developer ko ek nayi depth deta hai, jiski wajah se developer multiple fields mein strong ho jata hai. Isi multi-domain depth ko M-Shaped Developer kehte hain.

---

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

##### (A) Why does Vibe Coding create problems after one week?

Vibe Coding planning aur documentation ke baghair hoti hai. Shuru mein fast lagti hai, lekin ek week baad code messy aur inconsistent ho jata hai. Refactoring mushkil hoti hai, data structures bar bar change hote hain, aur project direction unclear ho jata hai.

##### (B) How does SDD prevent those problems?

Specification-Driven Development pehle se clear requirements define karta hai. Jab specs written hoti hain to implementation stable rehta hai, bugs kam hotay hain aur future updates smooth hotay hain. SDD project ko organized, stable aur maintainable banata hai.

---

## 3. Architecture Thinking

### (A) How does architecture-first thinking change the role of a developer?

Architecture-first thinking developer ko low-level coding se system-level planning aur designing role mein le jati hai. Developer components, flows aur system structure plan karta hai jab ke agents implementation handle kar lete hain. Is tarah developer system designer ban jata hai.

### (B) Why must developers think in layers and systems instead of raw code?

Layered design system ko clean, scalable aur maintainable banata hai. Raw code sirf function-level soch hoti hai, jab ke layered thinking system-level clarity, separation of concerns aur better debugging provide karti hai. AI agents bhi layered structure ke saath zyada efficiently kaam karte hain.

## PART B — PRACTICAL TASK

### CLI Prompt Used

Generate a one-paragraph specification for an email validation function.

The function must check that the email contains “@”, includes a valid domain like .com or .org, and returns clear and specific error messages if the email is invalid.



```
C:\Windows\system32\cmd.exe
> 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! 0.15.1 -> 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. The function must check that the email contains "@", include a valid domain like .com or .org, and returns clear and specific error messages if the email is invalid.

[+] The email validation function should accept a single string argument representing an email address and return a boolean indicating validity, along with a string containing a specific error message if invalid, or an empty string if valid. The function must verify the presence of an "@" symbol and ensure the domain part includes a top-level domain (TLD) such as ".com" or ".org", rejecting formats like "user@domain" without a TLD or "user@.com". Error messages should clearly articulate the reason for invalidity, for example, "Email must contain @" or "Invalid domain format".

> [ ] Type your message or @path/to/file

~\Desktop\assignment_2    no sandbox (see /docs)    auto
```

## **PART C — MCQs (Answers)**

1. B
2. B
3. B
4. B
5. C