AIDD-30-days-challenge - Task 2

■Part A — Theory (Short Questions)

1. {Nine Pillars Understanding}..

Q1. Why is using AI Development Agents (like Gemini CLI) for repetitive setup tasks?

Answer.. AI Development Agents jaise Gemini CLI repetitive setup tasks ko fast, sahi aur bina galti ke automate kar dete hain jis se time bachta hai aur developer high-level kaam par focus kar sakta hai.

Q2. Explain how the Nine Pillars of AIDD help a developer grow into an M-Shaped?

Answer.. Nine Pillars developer ko har direction me strong banate hain—specs, testing, AI tools, cloud, aur domain skills—jis se wo M-shaped developer ban jata hai jo AI ke sath end-to-end system build kar sakta hai.

Example:

Agar developer login system banana chahta hai, to wo

– SpecKit se clear spec likhta hai,

– AI CLI se code generate karwata hai,

– TDD se tests banwata hai,

– Aur Docker/Kubernetes se deploy kar deta hai.

Yeh sab skills milkar usko M-shaped expert bana dete hain.

2. {Vibe Coding vs Specification-Driven Development}..

Q1. Why does Vibe Coding usually create problems after one week?

Answer.. Vibe Coding aik weak-built coding environment hota hai, jisme bugs hoty hain. Is liye 1 hafte baad settings, extensions, ya file handling corrupt ho jati hai aur errors dena shuru kar deti hai.

Q2. How would Specification-Driven Development prevent those problems?

Answer.. Specification-Driven Development me pehlay se clear rules, steps, aur requirements likh di jati hain. Is se coding environment random errors nahi deta, kyunki sab kuch documented aur predictable hota hai.

3. {Architecture Thinking}

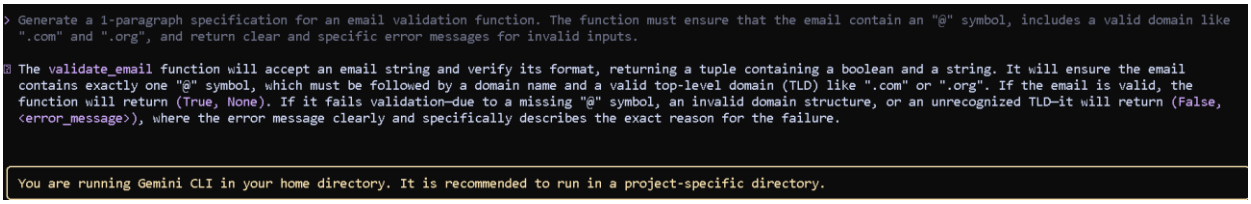
Q1. How does architecture-first thinking change the role of a developer in AIDD?

Answer.. Architecture first thinking developer ko sirf coder se utha kar system designer bana deta hai. Pehle pura structure flow aur data ka safar plan hota hai phir coding hoti hai is se mistakes rework aur confusion kam ho jata hai.

Q2. Explain why developers must think in layers and systems instead of raw code.

Answert.. Developers ko sirf raw code ke bare me nahi sochna chahiye. Agar wo layers aur systems me sochte hain, to unka software zyada scalable, maintainable aur AI-friendly ban jata hai. Har layer ka apna role hota hai, aur system approach se errors easily trace aur fix kiye ja sakte hain.

💼 Part B — Practical Task



Prompt..

Generate a 1-paragraph specification for an email validation function. The function must ensure that the email contains an "@" symbol, includes a valid domain like ".com" or ".org", and returns clear and specific error messages for invalid inputs.

Output..

The validate_email function will accept an email string and verify its format, returning a tuple containing a boolean and a string. It will ensure the email contains exactly one "@" symbol, which must be followed by a domain name and a valid top-level domain (TLD) like ".com" or ".org". If the email is valid, the function will return (True, None). If it fails validation—due to a missing "@" symbol, an invalid domain structure, or an unrecognized TLD—it will return (False, <error_message>), where the error message clearly and specifically describes the exact reason for the failu

🗂Part C — Multiple Choice Questions

Spec-Driven Development

1. What is the main purpose of Spec-Driven Development?
    - **B. Clear requirements before coding begins**

AI-Driven Development Mindset

2. What is the biggest mindset shift in AI-Driven Development?
    - **B. Thinking in systems and clear instructions**

Vibe Coding

3. Biggest failure of Vibe Coding?
    - **B. Architecture becomes hard to extend**

AI CLI Agents

4. Main advantage of using AI CLI agents (like Gemini CLI)?
    - **B. Handle repetitive tasks so dev focuses on design & problem-solving**

M-Shaped Developer

5. What defines an M-Shaped Developer?
    - **C. Deep skills in multiple related domains**