

***** AIDD 30-Day Challenge _Task 2 *****

Name: Alishba

Slot: Friday(6 to 9)

Part A (QUESTION ANSWER)

NINE PILLARS UNDERSTANDING:

Qestion1: Why is using AI Development Agents (like Gemini CLI) for repetitive setup tasks better for your growth as a system architect?

Answer1: AI agents boring aur repeat hone wale tasks jaldi aur sahi tarah kar lete hain. Is se developer ka time bach jata hai, aur wo system ki planning aur design par focus kar sакta hai. Ye cheez developer ko system architect jaisi soch develop karne mein help karti hai.

Qestion2: Explain how the Nine Pillars of AIDD help a developer grow into an M-Shaped Developer.

Answer2: Nine Pillars developer ko alag alag areas sikhate hain, jaise specs, testing, AI tools, aur automation. Isse developer ek se zyada fields mein strong ho jata hai, jo usay M-Shaped Developer banne mein help karta hai.

VIBE CODING VS SPECIFICATION-DRIVEN DEVELOPMENT:

Qestion1: Why does Vibe Coding usually create problems after one week?

Answer1: Vibe Coding mein planning nahi hoti, is liye code jaldi messy ho jata hai. Kuch din baad samajhna mushkil ho jata hai ke kya kahan likha hai, aur errors bhi zyada aane lagtay hain.

Qestion2: How would Specification-Driven Development prevent those problems?

Answer2: Specification-Driven Development mein pehle clear likh liya jata hai ke kya banana hai. Is se code clean aur organized rehta hai, aur baad mein changes karna bhi easy hota hai, is liye problems nahi hoti.

ARCHITECTURE THINKING:

Qestion1: How does architecture-first thinking change the role of a developer in AIDD?

Answer1: Architecture-first thinking developer ko sirf code likhne wale se system plan karne wala bana deti hai. Wo pehle pura system ka structure sochta hai, phir coding karta hai. Is se uska role zyada mature ho jata hai.

Qestion2: Explain why developers must think in layers and systems instead of raw code.?

Answer2: Layers mein sochna se project clean aur organized rehta hai. Har cheez apni jagah hoti hai, bugs kam hotay hain, aur project ko future mein update ya expand karna easy ho jata hai. Raw code se cheezin jaldi messy ho jati hain.

Part B(Practical Task)



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.

> write a one paragraph specification for an email validation function. the email must contain @, and must contain a valid domain e.g:(.com, .org), and should return clear error messages

✉ The email validation function will check if a given string is a properly formatted email address. It will ensure the string contains exactly one "@" symbol, which must not be at the beginning or end of the string. The function will also verify that the domain part of the email (the portion after the "@") contains at least one period, and that the top-level domain (the part after the final period) consists of at least two letters, such as ".com", ".org", or ".net". If the email fails any of these checks, the function will return a clear, specific error message indicating the nature of the problem, such as "Missing '@' symbol", "Invalid domain name", or "Domain must end with a valid extension like .com or .org"; otherwise, it will confirm the email is valid.

Part C(MCQs)

- 1(B)
- 2(B)
- 3(B)
- 4(B)
- 5(C)