

AI-Driven Development - 30-Day 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 better?

AI development agents baar-baar hone wale kaam jaldi kar dete hain. Yeh hamara waqt bachate hain. Setup, basic code banana, aur environment set karna AI bohat tezi se kar leta hai. Is tarah developer ka dimagh free hota hai aur woh architecture, design aur system thinking par focus kar sакta hai — yani wo kaam jo aik system architect karta hai.

Q2: How do the Nine Pillars of AIDD help a developer grow into an MShaped Developer?

Nine Pillars aik complete structure dete hain jismein TDD, SDD, AI CLI, agents, multi-model workflows, evaluation waghera shamil hain. Yeh sab milkar developer ko mukhtalif areas mein gehri samajh dete hain. Iss tarah developer sirf coder nahi rehta balkay M-Shaped Developer ban jata hai jismein 2–4 domains mein strong skills hoti hain — jaise architecture, testing, planning, AI tools waghera.

2. Vibe Coding vs Specification-Driven Development

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

Vibe coding planning ke baghair hota hai. Pehle lagta hai sab fast chal raha hai, lekin aik haftay baad problems shuru hoti hain. Code messy ho jata hai, duplicate cheezin hoti rehti hain, bugs aate hain aur developer khud nahi samajh paata ke pehle kya likha tha. Future changes mushkil ho jati hain.

Q2: How does Specification-Driven Development prevent those problems?

SDD mein pehle clear specification banayi jati hai — kya banana hai, kaise chalega, aur kya rules follow honge. Jab specification clear ho, to code saaf, maintainable aur predict-able hota hai. Bugs kam hotay hain, system scalable rehta hai, aur long-term development aasaan hoti hai.

3. Architecture Thinking

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

Architecture-first thinking developer ko coder se system designer bana deti hai. Line-by-line coding ke bajaye, developer poore system ko sochta hai — uski flow, layers, interactions aur scalability. Is tarah developer ka role system architect jaisa ho jata hai.

Q2: Why must developers think in layers and systems instead of raw code?

Modern AI apps bohat complex hoti hain. Agar developer sirf raw code mein sochay, system toot jata hai. Layers (Models → IDEs → Agents) everything ko clear banati hain, debugging aasaan karti hain aur future updates simple kar deti hain.

Part B — (CLI prompt + output)



```
ips for getting started:  
. Ask questions, edit files, or run commands.  
. Be specific for the best results.  
. Create GEMINI.md files to customize your interactions with Gemini.  
. ./help for more information.  
  
generate 1 paragraph specification for an email validation function. must contain "@" , must contain a valid domain (e.g., .com, .org) should return clear error messages  
  
The email validation function should accept a string as input and return a boolean indicating validity, along with a descriptive error message if validation fails. Specifically, the function must ensure the presence of a single "@" symbol separating the local part from the domain. Furthermore, it must validate that the domain contains at least one period and ends with a recognized top-level domain (TLD) such as ".com", ".org", ".net", etc., with a minimum of two characters after the last period. Clear and distinct error messages should be provided for each failure case, such as "Email must contain an '@' symbol", "Invalid domain format", or "Missing top-level domain".
```

Part C — Multiple Choice Questions

1. Purpose of SDD?

B — Clear requirements before coding begins

2. Biggest mindset shift in AIDD?

B — Thinking in systems and giving clear instructions

3. Failure of Vibe Coding?

B — Architecture becomes difficult to extend

4. Advantage of AI CLI agents?

B — Handle repetitive tasks so developers can focus on design and problem-solving

5. M-Shaped Developer?

C — Deep skills in multiple related domains