

1. Nine Pillars Understanding

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 jaise Gemini CLI repetitive setup tasks ke liye isliye behtar hai kyunke yeh hmein low-level, time-consuming kaam se free kar deta hai, takay hm apnay asli role system architect pr focus kr skein

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

Nine Pillars of AIDD ek developer ko M-Shaped developer banne mein isliye help karte hain kyunke har pillar developer ki skills ko alag-alag direction mein grow karta hai. Pehlay developers sirf coding ya syntax par focus karte thay, lekin AIDD ke pillars aapko architecture, prompting, evaluation, automation, collaboration, validation, aur system-level thinking sab sikhate hain

2. Vibe Coding vs Specification-Driven Development

Why does Vibe Coding usually create problems after one week?

Vibe Coding means flow mein code likhna jo dimagh mein aaye woh banate jana without proper planning, documentation, ya requirements likhe hue result 1 ya2 din tak to theek lagta hai but within a week problems explode hoati hein jesy k agents confuse hojatay hein , direction khatam hojati hai , code inconsistent ho jata hai

How would Specification-Driven Development prevent those problems?

Har feature ka purpose, flow, acceptance criteria pehle se likha hota hai zero confusion code consistent rehta hai style or architecture stable hota hai , testing easy hojati hai , refactoring safe hoti hai ek bar bnta hai or fast output receive hota hai

3. Architecture Thinking

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

Architecture-first thinking developer ka role AIDD mein badal deta hai. Ab developer line-by-line code nahi likhta; woh system ka blueprint banata hai, modules aur data flow design karta hai. Implementation aur repetitive tasks AI agents handle karte hain, jis se developer apna focus design, scalability, aur system quality par rakh sakta hai. Is tarah developer micro-level coder se evolve hoke ek system architect ban jata hai jo AI ke saath complex systems efficiently design aur implement karta hai.

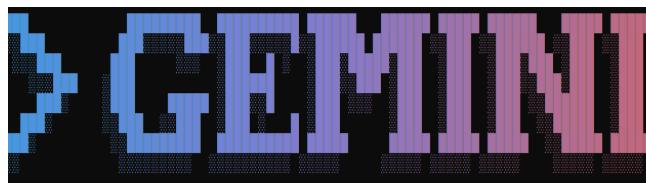
Explain why developers must think in layers and systems instead of raw code

Developers ko layers aur systems mein sochna is liye zaroori hai kyunki AIDD mein raw code likhna sirf choti problems solve karta hai, lekin complex software ko manage karne ke liye structure aur coordination chahiye. Jab layers (frontier model, AI-first IDE, development agents) aur systems ke hisaab se sochein tou

- AI repetitive aur low-level tasks handle karta hai,
- Developer design aur architecture par focus kar sakte ho,
- Code modular, maintainable aur scalable banta hai.

Part B — Practical Task

Prompt



```
tips 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 "@" , a valid domain (like .org , .com) and should return clear error messages
```

Answer

The validate_email function will analyze a given string to determine if it is a structurally valid email address. It must ensure the string contains exactly one "@" symbol, which serves to separate the local part from the domain. The domain part must include at least one period, and the top-level domain (e.g., ".com", ".org") must consist of two or more alphabetic characters. If the validation fails at any step, the function will not raise an exception but will instead return a clear, user-friendly error message specifying the exact reason for failure, such as "Email must contain an '@' symbol" or "Invalid email domain." A successful validation will return a confirmation of validity, ensuring the calling process can easily distinguish between success and failure.

Part C — Multiple Choice Questions

- What is the main purpose of Spec-Driven Development?
 - A. Make coding faster
 - B. Clear requirements before coding begins
 - C. Remove developers
 - D. Avoid documentation
- What is the biggest mindset shift in AI-Driven Development?
 - A. Writing more code manually
 - B. Thinking in systems and clear instructions
 - C. Memorizing more syntax
 - D. Working without any tools
- Biggest failure of Vibe Coding?
 - A. AI stops responding
 - B. Architecture becomes hard to extend
 - C. Code runs slow
 - D. Fewer comments written
- Main advantage of using AI CLI agents (like Gemini CLI)?
 - A. They replace the developer completely
 - B. Handle repetitive tasks so dev focuses on design & problem-solving
 - C. Make coding faster but less reliable
 - D. Make coding optional
- What defines an M-Shaped Developer?
 - A. Knows little about everything
 - B. Deep in only one field

C. Deep skills in multiple related domains

D. Works without AI tool