

AIDD (AI-Driven Development) 30-Days Challenge - Task 2 Submission

Participant: Syeda Hafiza Bbi Amna (00477608)

Slot: Friday - 6:00 PM to 9:00 PM

Instructor: Sir Hamzah Syed

Part A - Theory (Short Questions)

1. Nine Pillars Understanding

Q1: Why is using AI Development Agents (like Gemini CLI) for repetitive setup tasks better for your growth as a system architect?

AI Agents ko bar bar repetitive setup taske ke liye use karne se time consume kam hota hai or mistakes bhi kam hoti hain. Jo waqt hum testing, configs or folder create karne me lagate hain uski jagh system ki architecture, planning, design per focus karte hai. Architecture matlab ke system ke different parts kese work kar rahe hain usko hum or kese bhetar kar sakte hai. With the use of AI humare skills improve hote hain as a developers or AI humen replace nahi karta humen next level system create karne me help karta hai.

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

Nine Pillars of AIDD se hum sekhte hai ke step by step se kese kam karte hain. Jab hum step by step kam karte hain to humari planning, testing zyda bhetar hoti hai kiyo ke humm har part per focus kar sakte hain. 9 pillars covers specification, testing , automation ,planning ,agents and code workflows. Ye 9 pillars ka combination developers ko ek single skill per limited nahi rakhta.

2. Vibe Coding vs Specification-Driven Development

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

Vibe coding me bina soche samjhe , without clear plan code likhte hai. Starting me ye easy lagta hai mgr jese code ko samjhte hai humko errors or bugs nazar ahty hai, bar bar logic or features repeat hote hai , messy code bana dety hai jis se code sahi se kam bhi nahi karta.

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

Specification Driven Development me hum phele se clear instruction, rules, structure likhte hai. Is se jo code ka output humen milta hai wo humen easily samjh ahta hai or future me aghr change karna ho code to Edit karne me easy rehta hai new developer bhi humare code ko dekhh kar samjh jaty hai.

3. Architecture Thinking

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

IN Architecture-First Approach hum designing per focus karte hai ke different parts kese connect kare ge data flow manage kese hogta or future me grow kese karen ge code writing badme hoti hai. Is approach se humara system strong banta hai or manage karne me easy rehta hai. Humen structure or decisions per focus karna hai baki AI code ko manage karta hai.

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

Layer-based approach me hum system ko pura ek sath dekhte hai code ko alg alg lines me nahi. Har part dosre part ke sath kam karta hai. Is ke benefits hai ke debugging, testing and maintenance me easy rahta hai.

Part B - Practical Task

Task:

Using any AI CLI tool, generate a 1-paragraph specification for an email validation function.

Requirements:

Must contain “@” Must contain a valid domain (e.g., .com, .org) Should return clear error messages

CLI Prompt:

```
Using: 1 GEMINI.md file | 1 MCP server
> Generate a 1-paragraph specification for an email validation function. Requirements: it must include "@", it must contain a valid
domain extension like .com or .org, and it should return clear error messages when validation fails.
~\Desktop\Gemini CLI Projects\task (master*)          no sandbox (see /docs)          gemini-2.5-flash | ✘ 4 errors (F12 for details)
```

CLI Output:

```
GEMINI
> Generate a 1-paragraph specification for an email validation function. Requirements: it must include "@", it must contain a valid
domain extension like .com or .org, and it should return clear error messages when validation fails.

* 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. A valid email must contain a single "@" symbol separating the local part from the domain. The domain part
must include at least one period and end with a recognized top-level domain (e.g., .com, .org, .net, .edu, .gov). If the input string
does not meet these criteria, the function should return false and a specific error message detailing the reason for failure, such as
"Missing '@' symbol", "Invalid domain format", or "Unsupported top-level domain".
```

Part C - Multiple Choice Questions

1. What is the main purpose of Spec-Driven Development?
 - a) Clear requirements before coding begins
2. What is the biggest mindset shift in AI-Driven Development?
 - a) Thinking in systems and clear instructions

3. Biggest failure of Vibe Coding?

b) Architecture becomes hard to extend

4. Main advantage of using AI CLI agents (like Gemini CLI)?

b) Handle repetitive tasks so dev focuses on design & problem-solving