AI Driven & AI Native Development

Task 2

PART A (Question Group)

Nine Pillars + AI Agents

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

My Answer:

AI agents handle boring and repetitive setup work for me. Because of that, I can focus on planning, designing, and understanding the system instead of wasting time on small tasks. This helps me grow into a better architect, not just a coder.

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

My Answer:

The Nine Pillars teach me different important skills. Like specs, testing, evaluations, agents, and architecture.
When all these skills combine, I become strong in multiple areas, not just one.
This is what makes me an M Shaped developer: someone who understands many connected parts of software development.

Vibe Coding vs Specification-Driven Development

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

My Answer:

Vibe coding becomes a problem because there is no clear plan.
You write whatever feels right in the moment, so after a few days the code becomes confusing, hard to extend, and full of mistakes.
You forget why you wrote something, and adding new features becomes difficult.

Q1. How would Specification-Driven Development prevent those problems?

My Answer:

Specification-Driven Development fixes everything by giving a clear written plan before coding.
You define exactly what the system should do, so there is no confusion later.
The work becomes organized, easy to extend, and the AI agent can follow your specs correctly.

Architecture Thinking

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

My Answer:

When you think like an architect, you stop being just a code writer.
You start planning how the whole system should work, how parts connect, and how AI agents will build it.
Your role becomes higher-level: you guide, design, and validate instead of typing every line yourself.

Q1. Why thinking in layers/systems is important

My Answer:

Thinking in layers keeps your system clean and organized.
Each layer has a job: models, tools, agents, etc.
This makes the system easier to understand, easier to update, and easier for AI agents to work with.
Without layers, everything becomes messy and hard to fix.

PART B (Practical Task)

Task:

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

My Answer:

Will be provided along with this document

PART C (Multiple Choice Questions)

1. 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

My Answer:

B. Clear requirements before coding begins

2. 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

My Answer:

B. Thinking in systems and clear instructions

3. Biggest failure of Vibe Coding?

A. AI stops responding

B. Architecture becomes hard to extend

C. Code runs slow

D. Fewer comments written

My Answer:

B. Architecture becomes hard to extend

4. 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

My Answer:

B. Handle repetitive tasks so dev focuses on design & problem-solving

5. 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 tools

My Answer:

B. Deep skills in multiple related domains