Task-2

Part-A

1. Nine Pillars Understanding

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

A: Offloading repetitive setup tasks to AI agents allows you to focus on designing the overall structure of the system rather than spending time on routine actions. This shift helps you think more strategically about how components should interact, how data should flow, and what architecture best supports long-term growth. By freeing you from mechanical work, you develop stronger architectural judgment and system-level thinking.

Q: How do the Nine Pillars of AIDD help a developer grow into an M-Shaped Developer?

A: The Nine Pillars encourage a balanced combination of deep technical skill and broad practical knowledge. They push developers to not only write code, but also understand architecture, evaluation, documentation, planning, testing, and effective collaboration with AI. This blend of depth and breadth shapes them into M-shaped developers—individuals with strong mastery in key areas while also having wide-ranging abilities that support complete system development.

---

2. Vibe Coding vs Specification-Driven Development

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

A: Vibe Coding relies on instinct and spontaneous decisions rather than structure. While it may feel fast at the beginning, the lack of planning causes the project to drift. After a short time, the code becomes inconsistent, difficult to read, and hard to extend because there is no clear foundation or shared direction. This lack of order leads to confusion, repeated work, and growing technical debt.

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

A: Specification-Driven Development creates clarity before coding begins. By defining requirements, system behaviour, and architectural boundaries in advance, it provides a stable guide for development. This reduces misunderstandings, keeps decisions consistent, and ensures that both developers and AI agents work toward the same structure. As a result, the system remains organized, predictable, and much easier to maintain.

---

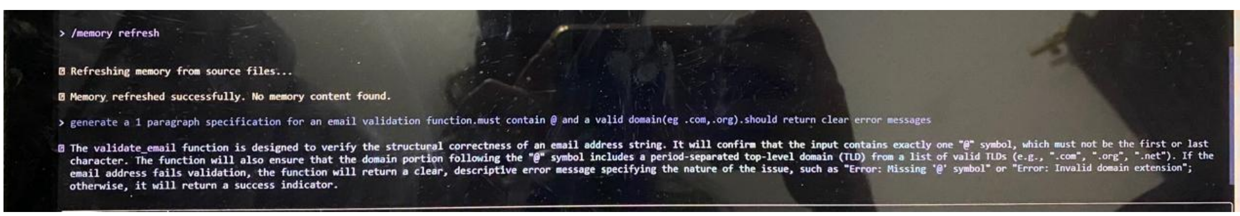
3. Architecture Thinking

Q: How does architecture-first thinking change the role of a developer in AIDD?
A: Architecture-first thinking transforms the developer’s role from a code-focused worker to a system designer. Instead of jumping straight into implementation, the developer is responsible for shaping how the

entire system should function, how components should be arranged, and how AI agents should support the building process. It elevates the developer to a more strategic position where guiding the system matters as much as writing code.

Q: Why must developers think in layers and systems instead of raw code?
A: Modern applications are too complex to manage by looking only at individual lines of code. Thinking in layers helps separate responsibilities—such as user interface, business logic, and data—which results in cleaner design and easier maintenance. Viewing the project as a system ensures that all parts align and interact properly, making the software more scalable, stable, and understandable over time.

Part-B: Practical task



Part-C

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

---

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

---

3. Biggest failure of Vibe Coding?

A. AI stops responding

✓ B. Architecture becomes hard to extend

C. Code runs slow

D. Fewer comments written

---

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

---

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

DONE BY: MARIUM ANWAR