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?

Using AI Development Agents **(like Gemini CLi)** for setup work removes the boring, repetitive steps from your workflow. Instead of wasting time on configurations, scaffolding, and boilerplate code, you stay focused on architecture, planning, and decision-making. This builds your thinking at a higher level, like how systems fit together, not just how to type code. Over time, this habit trains you to think like a system architect instead of a task executor.

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

The Nine Pillars create a complete environment where a developer works with AI, specifications, testing, agents, and automated tools. Because AI fills knowledge gaps, the developer can learn and practice multiple skills at once architecture, coding, testing, documentation, automation, and orchestration. This multi-domain exposure builds deep strengths in several areas, turning the developer into an M-Shaped professional with expertise across multiple connected fields.

2. Vibe Coding vs Specification-Driven Development

Q1: Why does Vibe Coding creausually te problems after one week?

Vibe Coding depends on mood and intuition instead of planning. The code works at first, but after a few days it becomes messy, inconsistent, and hard to understand. No one remembers why certain decisions were made, and adding new features becomes painful. Because nothing is documented, the system becomes fragile very quickly.

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

SDD forces us to write clear, testable specifications before touching the code. These specs act like a blueprint, so the implementation stays organized and predictable. Anyone can understand the system just by reading the specifications. This prevents confusion, reduces bugs, and keeps the project stable even as it grows.

3. Architecture Thinking

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

When developers think architecture first, they stop behaving like code typists and start acting like system designers. Their job becomes planning modules, defining data flow, and coordinating

AI agents to implement the structure. Instead of focusing on syntax, they focus on the system's long-term stability and scalability.

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

Thinking in layers helps us to understand how different parts of the software work together. Raw code only shows the surface, but layered thinking reveals structure, boundaries, responsibilities, and scalability. In AIDD, AI tools can write code for us but only if we provide a clear layered design. This mindset leads to cleaner, modular, and maintainable systems.

Part B — Practical Task (Screenshot Required)

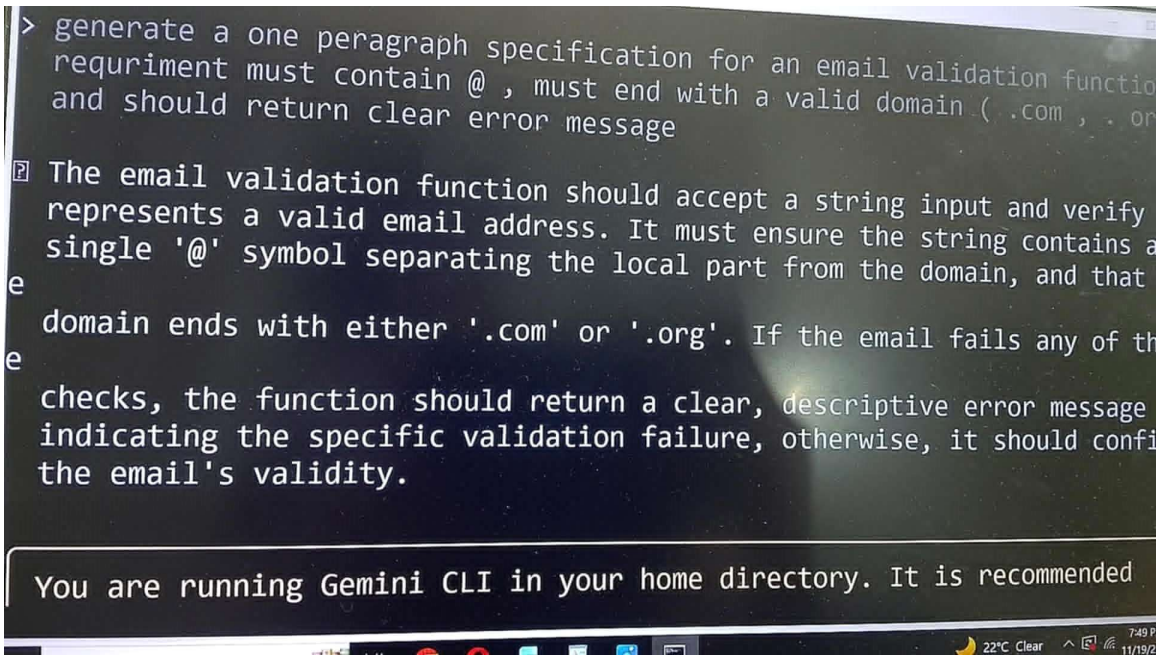
Task: Using any AI CLI tool, generate a 1-paragraph specification for an email validation function.

Requirements:

Must contain "@"

Must contain a valid domain (e.g., .com, .org)

Should return clear error messages



Part C — Multiple Choice Questions

1 = B

2 = B

3 = B

4 = B

5 = C