AIDD Task 2:

📁Part A — Theory (Short Questions)

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: Using AI development agents for repetitive tasks is a very strategic decision because it reduces our time to do the same tasks again and again and allows us to invest more time in decision making and debugging.

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

A: Nine pillars is necessary because every pillar gives developers a separate strong skill such as : using AI tools,creating AI agents for repetitive tasks, writing Specs,getting domain knowledge,VIBE coding. When these skills combine in one mind, the developer becomes a multi-skill and deep understanding geek.

Just like that developer not only does coding but can smartly integrate his knowledge in other domains and this thing is necessary for a M-shaped developer.

Vibe Coding vs Specification-Driven Development

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

A:Vibe Coding creates coding problems because initially there is no plan, the code becomes messy, and as the project grows, managing everything becomes difficult.

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

A:In an SDD, it is first clearly written what the system is supposed to do. This fixes the requirements before coding even begins. Result: less confusion, less rework, fewer bugs, and the team knows exactly what to build

Architecture Thinking

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

A: Architecture-first thinking transforms a developer's role from a coder to a system designer. In this approach, the developer first creates a clear system design and then writes the code. In AIDD, the developer essentially guides the AI with a strong design.

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

A: When you think only in raw code, everything becomes messy, slow, and confusing. Layers and systems help you divide the project into small, clear parts—like the UI layer, logic layer, and data layer. This makes understanding, testing, reusing, and fixing code 100x easier. System thinking teaches a developer to see the big picture, not just lines of code.

📁 Part B — Practical Task (Screenshot Required)

Task:

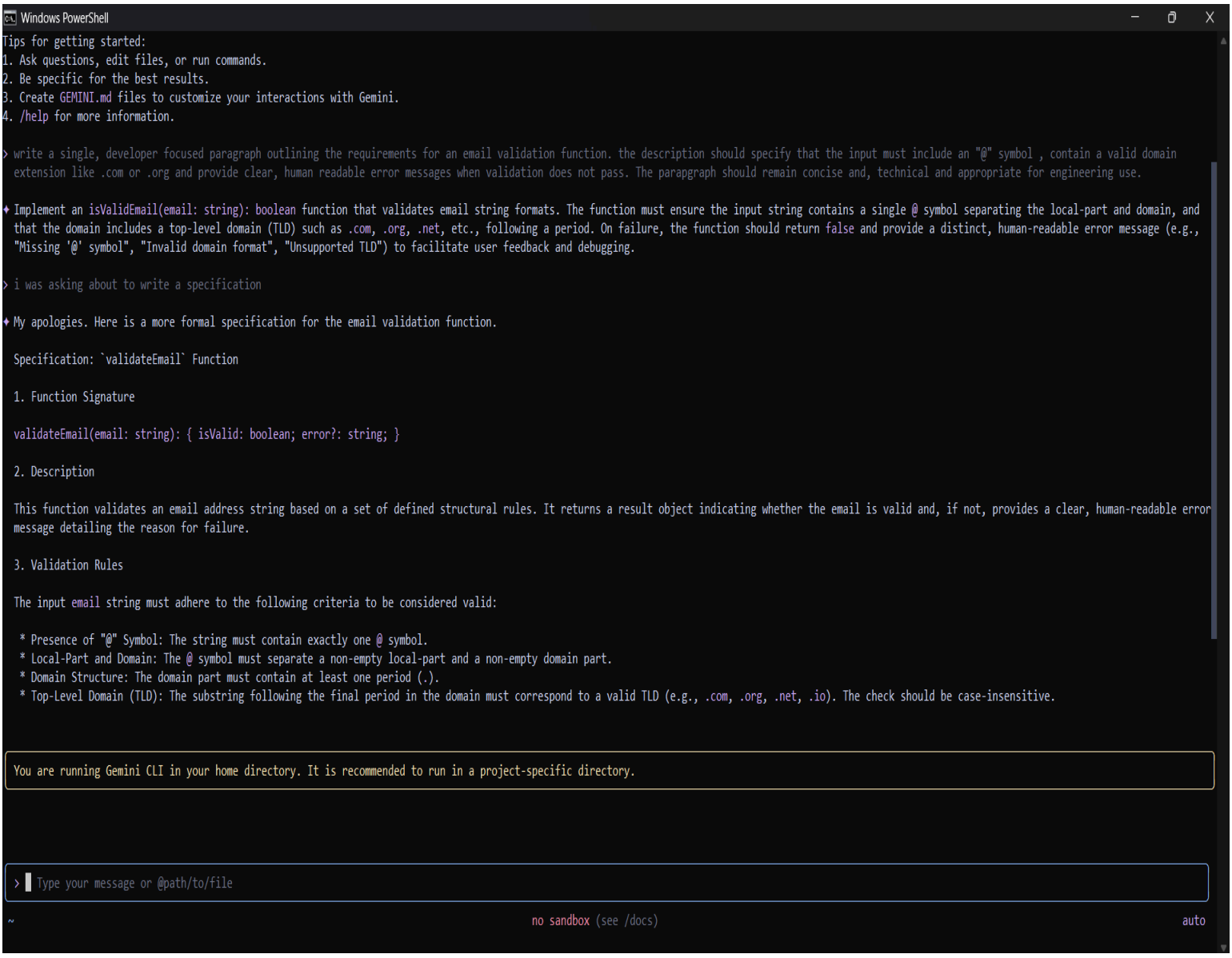
Using any AI CLI tool, generate a 1-paragraph specification for an email validation function.

Requirements:

Must contain “@”

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

Should return clear error messages



📁Part C — Multiple Choice Questions

1. 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

Name: Mohammad Danish

Roll No: 00037323