Part A- Theory (Short Questions)

1. Nine Pillars Understanding

(i): 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 for repetitive setup work saves my time and mental energy. Instead of getting stuck on boring or repeated tasks, I can focus on the bigger picture—how the system should work, how different parts connect, and what the overall architecture should look like. This helps me think more like a system architect instead of just someone writing lines of code. It shifts my attention from "doing tasks" to "designing systems."

(ii): Explain how the Nine Pillars of AIDD help a developer grow into an M-Shaped Developer.
The Nine Pillars give a structured way of working where specs, testing, agents, and architecture all connect together. When I follow these pillars, I naturally learn more than just coding—I start understanding planning, automation, testing, workflow design, and system thinking. This combination builds deep skills in multiple areas, which is exactly what an M-Shaped Developer is: someone strong in several connected domains, supported by AI tools.

2. Vibe Coding vs Specification-Driven Development

(i): Why does Vibe Coding usually create problems after one week?
Vibe Coding feels fast at the start, but after a few days, everything becomes confusing. There are no clear requirements, no structure, and the code becomes messy. When I return to the project after a week, I forget why I wrote certain parts, and adding new features becomes difficult. Bugs start appearing because nothing was planned properly.

(ii): How would Specification-Driven Development prevent those problems?
Specification-Driven Development gives clarity before I write a single line of code. When the rules and behavior are written clearly, the whole project stays organized. Even if I come back after a week, I know exactly what the function should do, how it should respond, and why it exists. Specs act like a guide that keeps the project stable, easy to extend, and easy to maintain.

3. Architecture Thinking

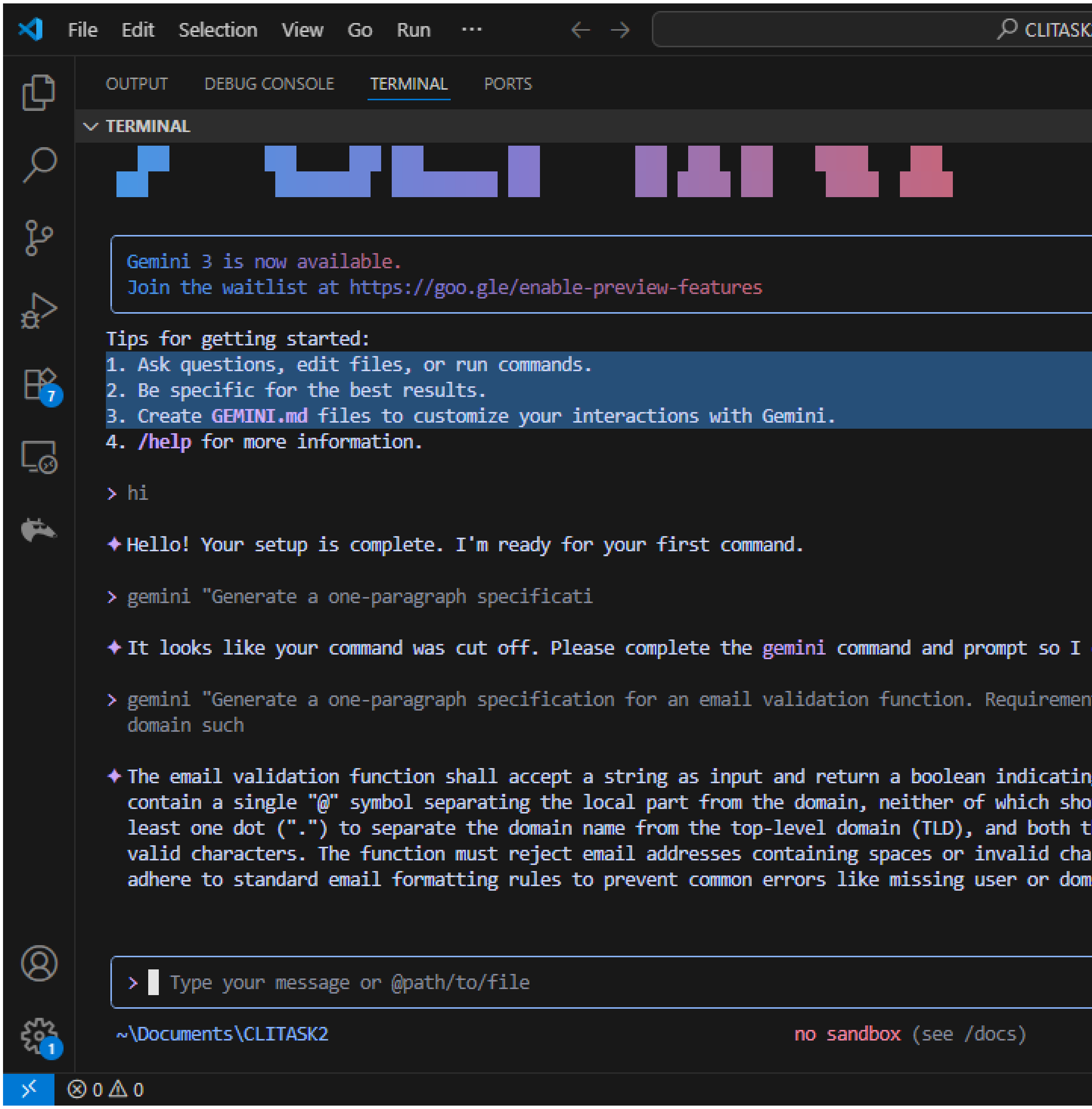
(i): How does architecture-first thinking change the role of a developer in AIDD?
With architecture-first thinking, a developer is no longer just a coder. The role becomes more strategic. I start thinking like a designer who plans the system, not just someone who writes

functions. AI handles many coding tasks, so my main job is to give clear instructions, define structure, and design how everything should work together.

(ii): Explain why developers must think in layers and systems instead of raw code.

Thinking in layers helps developers separate responsibilities. Instead of mixing everything together, I understand which part handles logic, which part handles the interface, and which part handles automation. This layered thinking makes projects easier to scale, easier to fix, and easier to hand over to AI agents. Raw code is just one small piece—systems thinking is what keeps the whole project stable.

Part B- Practical Task (Screenshot Required)



```
Gemini 3 is now available.
Join the waitlist at https://goo.gle/enable-preview-features

Tips for getting started:
1. Ask questions, edit files, or run commands.
2. Be specific for the best results.
3. Create GEMINI.md files to customize your interactions with Gemini.
4. /help for more information.

> hi

✦ Hello! Your setup is complete. I'm ready for your first command.

> gemini "Generate a one-paragraph specificati

✦ It looks like your command was cut off. Please complete the gemini command and prompt so I

> gemini "Generate a one-paragraph specification for an email validation function. Requiremen
  domain such

✦ The email validation function shall accept a string as input and return a boolean indicatin
  contain a single "@" symbol separating the local part from the domain, neither of which sho
  least one dot (".") to separate the domain name from the top-level domain (TLD), and both t
  valid characters. The function must reject email addresses containing spaces or invalid cha
  adhere to standard email formatting rules to prevent common errors like missing user or dom

> Type your message or @path/to/file

~\Documents\CLITASK2                              no sandbox (see /docs)
⊗ 0 ⚠ 0
```

Part C- Multiple Choice Questions

1. What is the main purpose of Spec-Driven Development?

B. Clear requirements before coding begins

2. What is the biggest mindset shift in AI-Driven Development?

B. 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

5. What defines an M-Shaped Developer?

C. Deep skills in multiple related domains