

## **TASK 2: AI-DRIVEN DEVELOPMENT**

### **Part A**

#### **1. Nine Pillars Understanding**

Using AI Development Agents for repetitive setup work is helpful because it saves your time and lets you focus on planning and system design. This improves your thinking as a system architect. The Nine Pillars support developers by combining tools, testing, agents, and clear instructions. This helps a person grow into an M-shaped developer who has strong skills in many areas.

#### **2. Vibe Coding vs Specification-Driven Development**

**Vibe Coding** creates problems after a week because the developer writes code without planning or structure. Later, the code becomes confusing and hard to update.

**Specification Driven Development** avoids these issues by writing clear instructions before coding. This makes the system organized and easy to maintain.

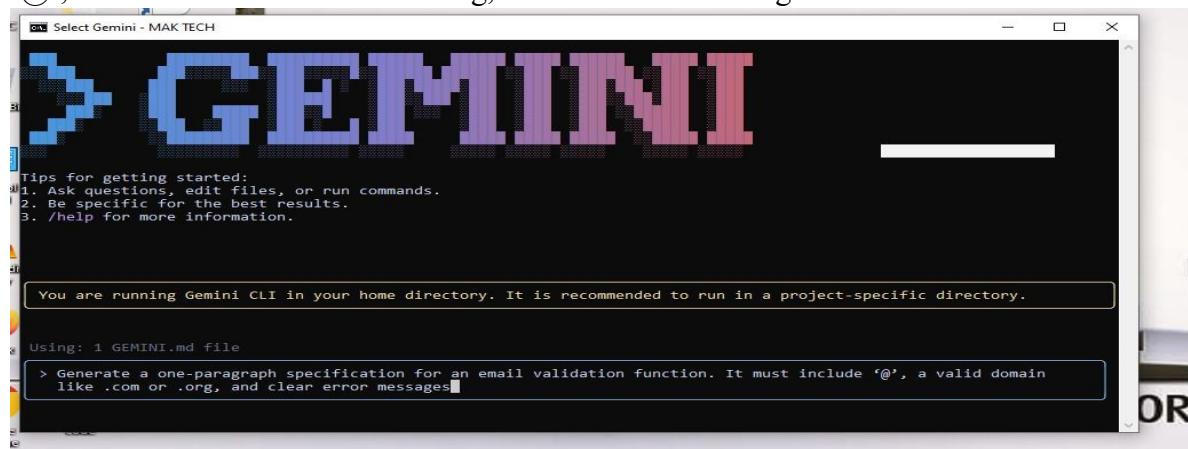
#### **3. Architecture Thinking**

Architecture-first thinking changes the developer's role from just writing code to designing full systems. Developers must think in layers models, tools, and agents so the system stays clean, scalable, and easy to manage.

### **Part B**

#### **Prompt:**

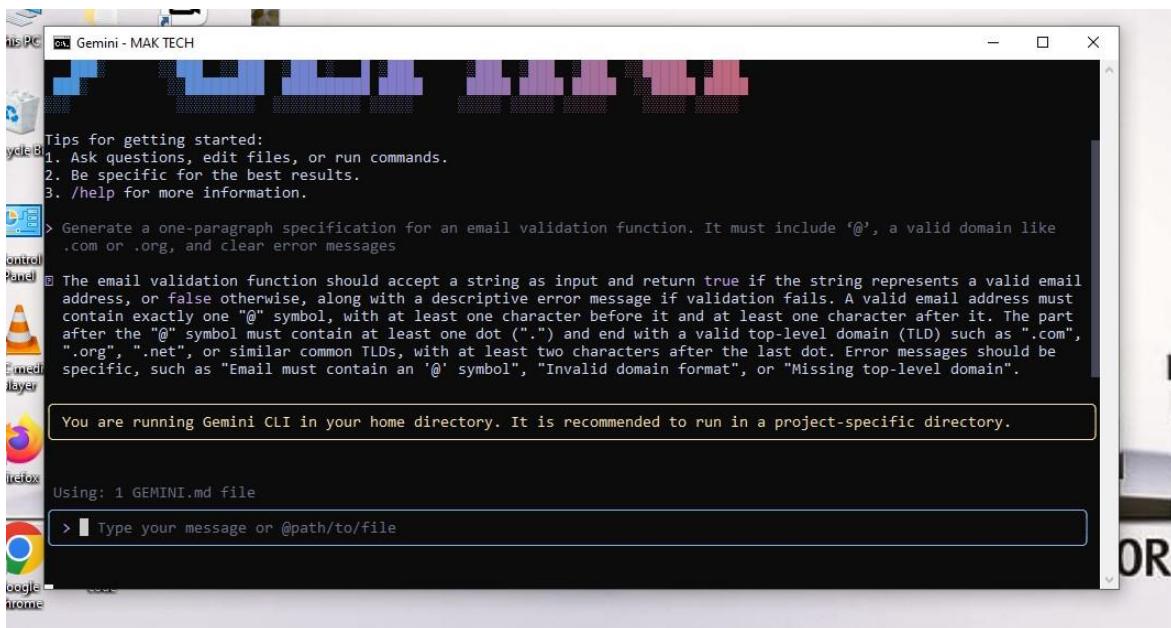
“Generate a one-paragraph specification for an email validation function. It must include ‘@’, a valid domain like .com or .org, and clear error messages.”



The screenshot shows a terminal window titled "Select Gemini - MAK TECH". The window features a large, stylized "GEMINI" logo at the top. Below the logo, there is a "Tips for getting started:" section with three numbered items: 1. Ask questions, edit files, or run commands. 2. Be specific for the best results. 3. /help for more information. A message at the bottom of the screen says, "You are running Gemini CLI in your home directory. It is recommended to run in a project-specific directory." In the command-line area, the user has typed "Using: 1 GEMINI.md file" and then issued a command starting with "> Generate a one-paragraph specification for an email validation function. It must include '@', a valid domain like .com or .org, and clear error messages." The terminal is running on a Windows operating system, as indicated by the taskbar at the bottom.

### ***Generated Specification:***

The email validation function should accept a string as input and return true if the string represents a valid email address, or false otherwise, along with a descriptive error message if validation fails. A valid email address must contain exactly one "@" symbol, with at least one character before it and at least one character after it. The part after the "@" symbol must contain at least one dot (".") and end with a valid top-level domain (TLD) such as ".com", ".org", ".net", or similar common TLDs, with at least two characters after the last dot. Error messages should be specific, such as "Email must contain an '@' symbol", "Invalid domain format", or "Missing top-level domain"



### **Part C**

#### **MCQS**

**1.B**

**2.B**

**3.B**

**4.B**

**5.C**