🌐 AI-Driven Development — 30-Day Challenge

📝 Task 2 — Official Submission

⏱️ **Time: 1 Hour**
📈 **Marks: 10**
⏳ **Deadline: 48 Hours**
📍 **Class Slot: Friday — 6:00 PM to 9:00 PM**
👨‍🏫 **Instructor: Sir Hamzah Syed**

📌 Rules & Regulations

✔ **Answers must be written in your own words — keep them clear & original**
✔ **You may write in English or Roman Urdu**
✔ **Do not use GPT or any AI tool to write theory answers**
✔ **Practical task must include screenshot (CLI prompt + output)**
✔ **MCQs must be answered only by selecting A, B, C, or D**
✔ **Submit your Task in PDF or .md format**
✔ **Share your GitHub repository link in submission**

📚 Study Material

The following concepts and resources, taken from this book, cover the material needed for this task.

Resources

- **Part 1 — Chapter 2: The AI Turning Point**
- **Part 1 — Chapter 4: The Nine Pillars of AI-Driven Development**

Key Concepts to Understand

Concept	Description
3-Layer Architecture	Modular stack (Models → IDEs → Agents) separating intelligence, workspace, and orchestration
Vibe Coding	Coding without planning/specs — developer “vibes” into the code relying on intuition
Specification-Driven Development (SDD)	Writing clear, executable specifications before implementation — formalizing human intent
The Nine Pillars	Integrated system (TDD, AI CLI, MCP, agents, specs, etc.) allowing a single developer to function like an entire team
M-Shaped Developer	Developer with deep expertise in 2–4 complementary domains — enabled by AI bridging knowledge gaps
Development Agents	Autonomous Layer-3 tools (Gemini CLI, Claude Code, etc.) executing multi-step tasks from high-level instructions

📂Part A — Theory (Short Questions)

Answer in your own wording

1. **Nine Pillars Understanding**
   - Why is using AI Development Agents (like Gemini CLI) for repetitive setup tasks better for your growth as a system architect?
   - Explain how the Nine Pillars of AIDD help a developer grow into an M-Shaped Developer.
2. **Vibe Coding vs Specification-Driven Development**
   - Why does Vibe Coding usually create problems after one week?
   - How would Specification-Driven Development prevent those problems?
3. **Architecture Thinking**
   - How does architecture-first thinking change the role of a developer in AIDD?
   - Explain why developers must think in layers and systems instead of raw 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**

Submission:

Your exact CLI prompt (text or screenshot)

The 1-paragraph specification generated by the CLI

📁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**

🌟 Reflection

This task highlights how rapidly the developer role is evolving in the AI-Native era. Concepts like the Nine Pillars, SDD, and Development Agents are not just tools—they represent a complete shift in how software is planned, built, and maintained.

By practicing architecture-first thinking and learning to collaborate with agents, developers can grow from code writers into system-level thinkers, eventually becoming M-Shaped professionals who combine multiple deep skills with intelligent tooling.

🏷️ Prepared By:

Asma Yaseen & Yusra Saleem — Class Coordinators

Supervised By: Sir Hamzah Syed