@everyone

🎓 Class 03 Summary — Context Engineering (24 Oct 2025)

Today we explored **Context Engineering** — how developers structure what information LLMs receive — and key **image generation concepts**.

~~----------~~

🧠 **What is Context Engineering?**

It’s the art of giving an LLM the *right data, in the right structure, at the right time.*

As André Karpathy said: **“The LLM is the CPU, and the context window is the RAM.”**

Developers optimize that RAM — choosing what information fits best.

🧠 *Prompt engineering* → user-level

⚙️ *Context engineering* → developer-level

📘 [Context Engineering Tutorial](https://github.com/panaversity/learn-low-code-agentic-ai/blob/main/00_prompt_engineering/context_engineering_tutorial.md)

~~----------~~

⚔️ **Prompt vs Context Engineering**

Prompt = chat-style instructions.

Context = structured, code-like setup for agents (XML, JSON, markdown).

Agents can’t rely on back-and-forth; they must *think ahead*.

~~----------~~

🧩 **Six Essential Components of AI Agents**

1. **Model** – The AI engine (GPT, Claude, etc.)
2. **Tools** – APIs and external functions
3. **Knowledge & Memory** – Static + dynamic info
4. **Audio & Speech** – Natural voice interaction
5. **Guardrails** – Safety, tone, and policy filters
6. **Orchestration** – Coordinating agent activities

🍔 **Burger Analogy:**

- 🫓 Bun = Model (holds everything together)

- 🍖 Patty = Core functionality

- 🥬 Vegetables & Condiments = Tools, knowledge, audio, guardrails

- 🤹‍♂️🧿 Recipe that assembles it all = Context engineering

~~----------~~

🧠 **Example — AI Research Assistant**

A structured **system prompt** defines:

role, input/output format, XML-based tasks, and JSON summaries — enabling **autonomous action** through context control.