**1/ Lý do**

**Nhiều prompt voi các nghiệp vụ co bản khác nhau (trích xuất, tóm tắt, phân tích):**

**Cần truyền dữ liệu động (ví dụ {product}, {customer\_feedback}) thay vì copy-paste thủ công.**

**Phải quản lý, tái sử dụng, versioning prompt giống như code.**

* Do đó, dùng thư viện prompt mẫu sẽ:
* Giúp quản lý prompt như template.
* Dễ dàng nhúng dữ liệu động.
* Dễ kết hợp nhiều bước (ví dụ: trích xuất dữ liệu → phân tích → sinh báo cáo).

-🡪 Quản lý / tạo thư viên prompt với:

- LangChain PromptTemplate để xây dựng prompt động, dễ tái sử dụng và tích hợp vào backend

- Github để lưu trữ prompt

**2/ Một số công cụ thường dùng**

LangChain PromptTemplate

Dùng để biến prompt thành template có biến đầu vào {variable}.

Tích hợp dễ dàng vào Python backend (FastAPI, Flask).

LlamaIndex (trước đây là GPT Index)

Dùng để quản lý prompt khi làm RAG (tìm kiếm tài liệu, sinh báo cáo).

Ví dụ: prompt để “tóm tắt báo cáo tài chính” sẽ nằm trong query engine.

PromptHub / PromptLayer (tùy công ty)

Giúp lưu trữ và versioning prompt trên cloud.

Mỗi lần đổi prompt → team khác có thể xem lịch sử, rollback.

GitHub + internal library

Nhiều công ty đơn giản chỉ lưu prompt dưới dạng file .py hoặc .yaml trong repo GitHub.

AI Specialist sẽ viết thư viện riêng để quản lý và gọi prompt theo use case.