**🧭 INSTRUCTION: HỆ THỐNG SINH PROMPT TỰ ĐỘNG DỰA TRÊN ĐỀ BÀI**

**(BMAD-inspired Prompt Framework for Gemini + LiteLLM)**

**🎯 MỤC TIÊU**

Tạo **một framework sinh prompt tự động** dựa trên **đề bài hoặc mô tả dự án**, trong đó:

* Framework tự xác định **các vai trò (roles / agents)** cần có.
* Sinh ra **prompt template chuẩn** cho từng vai trò.
* Định nghĩa đầy đủ các trường (persona, context, responsibilities, …).
* Có thể chạy qua **Gemini API** để sinh prompt chính thức bằng 1 lệnh duy nhất.

**🧩 CẤU TRÚC TỔNG QUAN**

**1️⃣ Input (Đầu vào)**

Là **đề bài**, **mô tả dự án**, hoặc **yêu cầu sản phẩm**, ví dụ:

“Tạo 1 video bản tin truyền hình tổng kết các hoạt động kỷ niệm 80 năm Quốc khánh 2/9...”

**2️⃣ Output (Đầu ra)**

Là **bộ prompt YAML** cho từng agent, ví dụ:

* mc.yaml (Người dẫn chương trình)
* editor.yaml (Biên tập viên nội dung)
* director.yaml (Đạo diễn sản xuất)
* lighting.yaml (Kỹ thuật âm thanh – ánh sáng)
* graphic.yaml (Thiết kế đồ họa – hiệu ứng)
* qa.yaml (Kiểm duyệt & truyền thông)

**⚙️ QUY TRÌNH KHỞI TẠO PROMPT TỰ ĐỘNG**

**Step 1: Chuẩn hóa đề bài**

Tạo JSON khung để giữ dữ liệu đầu vào cho model:

{

"project\_name": "Video bản tin Quốc khánh 80 năm",

"product\_type": "Video tổng kết truyền hình",

"requirements": [

"Có người dẫn chương trình ảo (MC)",

"Lời dẫn chuẩn, thời lượng 80 giây",

"Định dạng MP4, full HD (1920x1080)",

"Phải có câu: 'Các hoạt động chính kỷ niệm 80 năm Quốc khánh 2/9...'"

],

"theme": "Kỷ niệm Quốc khánh 2/9",

"style": "Trang trọng, hiện đại, truyền cảm",

"target\_audience": "Khán giả Việt Nam"

}

**Step 2: Xác định vai trò (Roles / Agents)**

Gọi Gemini để xác định danh sách các vai trò cần có.

**🔹 Prompt mẫu:**

query = f"""

Phân tích dự án sau đây và liệt kê tất cả các vai trò cần thiết để hoàn thành nó.

Hãy đặt tên mỗi vai trò ngắn gọn, kèm mô tả nhiệm vụ chính.

Dự án:

{project\_json}

"""

**🔹 Output mẫu:**

[

{"role": "Biên tập viên nội dung", "id": "editor"},

{"role": "Người dẫn chương trình", "id": "mc"},

{"role": "Đạo diễn sản xuất", "id": "director"},

{"role": "Quay phim", "id": "camera"},

{"role": "Kỹ thuật âm thanh – ánh sáng", "id": "soundlight"},

{"role": "Thiết kế đồ họa – hiệu ứng", "id": "graphics"},

{"role": "Kiểm duyệt & truyền thông", "id": "qa"}

]

**Step 3: Sinh prompt template cho từng vai trò**

Gemini sẽ dựa theo **prompt khung** để tự định nghĩa YAML cho từng role.

**🔹 Prompt Template Gốc (dành cho API gọi Gemini):**

prompt\_template = f"""

Tạo prompt YAML cho vai trò: "{role\_name}"

Trong khuôn khổ dự án: {project\_name}

Các trường bắt buộc phải có:

- agent:

- name: tên ngắn gọn

- id: viết thường, không dấu

- title: tên chính thức của vai trò

- icon: emoji biểu trưng

- persona:

- vai\_trò: mô tả công việc

- phong\_cách: phong thái, tính cách làm việc

- giọng\_điệu: cách giao tiếp

- khẩu\_hiệu: một câu slogan hoặc triết lý làm việc

- chuyên\_môn: danh sách kỹ năng chính

- context: bối cảnh làm việc (mục tiêu, giới hạn, nguồn dữ liệu)

- core\_responsibilities: các nhiệm vụ chính

- output\_format: mô tả dạng dữ liệu hoặc kết quả mong đợi

- command\_examples: ví dụ lệnh hoặc truy vấn mà người dùng có thể gọi

- communication\_rules: quy tắc giao tiếp (giọng, ngôn ngữ, đạo đức, phong cách)

Yêu cầu:

- Viết bằng tiếng Việt.

- Ngắn gọn nhưng chi tiết, có thể dùng ngay làm prompt khởi tạo agent.

- Giữ cấu trúc YAML chính xác.

"""

**Step 4: Gọi Gemini qua LiteLLM để sinh từng agent**

Ví dụ Python:

from litellm import completion

import os, yaml

roles = ["editor", "mc", "director", "camera", "soundlight", "graphics", "qa"]

for r in roles:

prompt = prompt\_template.replace("{role\_name}", r).replace("{project\_name}", "Video Quốc khánh 80 năm")

res = completion(

model="gemini/gemini-2.5-flash",

api\_base="https://api.thucchien.ai/v1",

api\_key=os.getenv("LITELLM\_API\_KEY"),

messages=[{"role": "user", "content": prompt}],

)

text = res["choices"][0]["message"]["content"]

with open(f"agents/{r}.yaml", "w", encoding="utf-8") as f:

f.write(text)

✅ Kết quả: bạn có thư mục agents/ chứa các prompt YAML sinh tự động — y hệt như BMAD team file.

**Step 5: (Tùy chọn) Sinh prompt tổng hợp cho cả team**

Bạn có thể ghép tất cả YAML lại để sinh 1 file tổng hợp:

import glob

with open("team\_prompt.yaml", "w", encoding="utf-8") as out:

for file in glob.glob("agents/\*.yaml"):

out.write(open(file, encoding="utf-8").read() + "\n---\n")

**🧱 CẤU TRÚC TEMPLATE CHUẨN (để dùng cho mọi đề thi)**

Mỗi agent sinh ra phải tuân theo format:

agent:

name: "<tên>"

id: "<mã\_tắt>"

title: "<vai trò chính>"

icon: "🎬"

persona:

vai\_trò: "<mô tả công việc>"

phong\_cách: "<phong cách làm việc>"

giọng\_điệu: "<tone giao tiếp>"

khẩu\_hiệu: "<câu slogan>"

chuyên\_môn:

- kỹ năng 1

- kỹ năng 2

- kỹ năng 3

context:

- Mục tiêu làm việc

- Ràng buộc thời gian / chất lượng

- Bối cảnh sản xuất / nhóm

core\_responsibilities:

- Nhiệm vụ 1

- Nhiệm vụ 2

- Nhiệm vụ 3

output\_format:

- Dạng kết quả mong đợi

- Ví dụ chi tiết đầu ra

command\_examples:

- \*hướng\_dẫn\_nhiệm\_vụ()\*

- \*tạo\_bản\_thảo(đề\_tài="...")\*

communication\_rules:

- Không viết lan man

- Giữ giọng chuyên nghiệp và cảm xúc

- Tránh nội dung không phù hợp

**💡 KINH NGHIỆM TRIỂN KHAI**

| **Thành phần** | **Nhiệm vụ** | **Cách dùng** |
| --- | --- | --- |
| **Template Prompt (YAML Schema)** | Định dạng khung chuẩn cho mọi role | Tạo sẵn file prompt-template.yaml |
| **Gemini via LiteLLM** | Sinh nội dung thực tế cho từng role | Gọi model "gemini/gemini-2.5-flash" |
| **Preprocessor Script** | Đọc đề bài, sinh danh sách vai trò | Một hàm Python đơn giản |
| **Prompt Generator** | Gọi API, sinh file YAML tự động | Lưu ra agents/<role>.yaml |
| **(Tùy chọn)** Team Bundler | Gộp nhiều YAML lại để in / export | Dành cho chấm thi hoặc demo |

**🧠 TƯ DUY KHUNG (Framework Thinking)**

“Prompt không còn là văn bản, mà là cấu trúc dữ liệu mô tả hành vi.”

Mỗi agent là một **object** có trường persona + context + task,  
và framework của bạn chỉ cần “thay đề” → “sinh prompt”.