Prompt Architect – Orchestrator User Guide (v3)

Welcome to Prompt Architect — a precision-engineered agent designed to generate, evaluate, and refine AI prompts using structured reasoning and strictly defined resources.

# 🚀 Overview

Prompt Architect operates as a Prompt Orchestrator — it doesn’t chat, simulate, or explain answers. It runs workflows that generate prompt suggestions using internal tools, tags, and approved libraries only.

You can activate modes using tags:

• #prompt — Generate new prompt suggestions  
• #qa — Evaluate prompt quality and provide improvements  
• #learn — Explain why a prompt structure works  
• #edge — Generate edge-case or adversarial prompts

# 🧠 Workflow

The agent follows this workflow:

1. Interpret the task or goal  
2. Select appropriate styles ([INSTRUCTIONAL], [ROLE-BASED], etc.)  
3. Generate 3–5 labeled prompt suggestions  
4. Handle #learn with explanations  
5. Use #qa to critique or improve  
6. Use #edge to test model limitations  
7. Apply fallback or flag unsafe input

# 📁 File-Based Behavior

Prompt Architect only uses the following files to function:

- formatting\_preferences.txt  
- prompt\_library.txt  
- prompt\_templates.txt  
- qa\_module.txt  
- prompt\_injection\_tests.txt  
- specialized\_prompts.txt

No assumptions, memory, or outside knowledge are used — structure, style, QA, and generation rules come strictly from these files.

# ✍️ Using #prompt

Default mode. Provide a topic or task, and the agent returns prompt suggestions using multiple styles.

# 📖 Using #learn

Returns prompt suggestions and explains why each structure works, based on formal guidelines.

# 🧰 Using #qa

Evaluates a prompt’s clarity, scope, format, and tone. Response includes:

• Prompt  
• Issue  
• Suggestion  
• Reasoning

# 🧪 Using #edge

Generates edge-case prompts designed to test ambiguity, conflict, format juggling, or injection vulnerability.

All edge prompts are labeled and include notes such as:  
• 🚧 Ambiguity Risk  
• 🧪 Stress Test  
• ⚠️ Injection/Contradiction

# 🛡️ Guardrails and Escalation

• Injection prompts are flagged and halted  
• Unknown formats or unclear requests trigger clarification  
• Escalation occurs if multiple constraint failures occur

# 📋 Summary Table

• #prompt → Generate new prompts  
• #qa → Critique and improve prompt  
• #learn → Explain structure  
• #edge → Stress-test prompt behavior  
• (none) → Default is #prompt

# ⚠️ Disclaimer

Prompt Architect is a professional tool for experimentation and prompt design. Do not deploy prompt suggestions in production systems without review.