

# Agent Workflow Creation

---

You are AI professionals at various companies that are looking to create tools to help organize documents. Your goal is to utilize Flowise to create an Agentic RAG workflow for various domains like the Agentic RAG tutorial we went through together yesterday. Towards the end of the lab, you will have the chance to present your created workflow to your peers.

## Domains:

- **Workplace helper:** Create a knowledge-based assistant that answers questions regarding different company policies.
- **Cooking and nutrition:** Create a workflow that can help answer cooking questions about various cuisines and cooking techniques.
- **Tech Support:** Choose a set of products and create an assistant that can help answer technical questions regarding those products using the product documentation.
- **Exercise and Wellness:** Create a RAG that can answer questions regarding working out, wellness, sports, and other physical activities.
- **Tourism:** Choose a country and create an assistant that can help answer questions regarding various cities and sites in that country.
- **Finance Bot:** Create a RAG agent that can help parse through complicated financial documents from publicly traded companies to help traders and investors understand the company.

## Guidelines and Requirements:

- **Steps:**
  - Teams should first create 3-4 prompts about their topic they want to evaluate their agent against.
  - Teams should then evaluate how well the system handles these prompts without utilizing RAG.
    - This can be done by sending prompts to the **unmodified** Agentic RAG system built in Lab C.



- Teams should then search the internet for relevant documents for their topics and upload them to the various document stores provided.
  - Afterwards, teams should edit the prompts and document stores utilized in the Agentic RAG system from the prior lab and tailor it to their topic.
  - Teams should then qualitatively evaluate the same prompts now that the agent can access relevant documents.
- Data Guidelines
  - Keep the documents short but relevant.
  - Focus on finding publicly available, non-copyrighted information.

### Discussion Questions:

Throughout the development process, keep these questions in mind:

- How did the agent response improve after providing relevant documents?
- Were there any cases where adding documents did not help the agent provide information?
- Did you have to modify your system prompts to get better outputs? If so, what changes did you make?
- Did your self-correction or retry logic meaningfully improve the responses, or just repeat them?
- Did the answers feel factual and grounded in your documents, or did the model still “hallucinate”?
- Did you try any prompting techniques (ex. Few-shot prompting)? Did this improve the model performance?
- How easy was it to find the appropriate data to create your system? What issues could you run into with data when trying to scale up a system?

