

# Projet : Assistant Intelligent Multi-Compétences (RAG + Agents)

---

## Contexte

Les entreprises possèdent souvent de grandes quantités de documents internes (rapports, manuels, notes de cours, politiques, etc.). Un assistant intelligent capable à la fois de retrouver des informations précises dans ces documents et d'exécuter des actions (calculs, recherche en ligne, envoi d'e-mails, etc.) serait extrêmement utile.

Votre mission est de concevoir un système complet combinant :

1. RAG (Retrieval-Augmented Generation) pour répondre à des questions basées sur des documents choisis par l'équipe.
2. Agents et outils pour enrichir les capacités de votre système (ex. calcul, météo, recherche web, etc.).

## Objectif du projet

Construire un assistant intelligent personnalisé capable de :

- Répondre à des questions basées sur un corpus documentaire local (vos propres fichiers PDF, DOCX, etc.).
- Appeler automatiquement des outils lorsqu'une tâche l'exige.
- Maintenir une conversation contextuelle avec l'utilisateur.

## Livrables attendus

### Partie 1 – RAG (Retrieval-Augmented Generation)

Construisez un pipeline RAG capable de :

- Ingestion et indexation de documents (vectorisation via FAISS, ChromaDB, Milvus ou autre).
- Récupération pertinente selon la requête de l'utilisateur.
- Réponse générée par un modèle OpenAI, Mistral ou Llama3 via LangChain.
- Interface possible : Chainlit, Streamlit ou Gradio.

Exemple : « Selon le manuel de sécurité interne, quelles sont les procédures en cas d'incident ? »

### Partie 2 – Ajout d'outils (Agents)

Intégrez un ou plusieurs outils à votre assistant :

- Calculatrice (addition, soustraction, etc.).
- Météo (API fictive ou réelle).
- Recherche web (via Tavily, SerpApi ou DuckDuckGo).
- Lecture de calendrier ou todo list locale.

L'agent doit être capable de choisir quand utiliser un outil en fonction du contexte de la question.

### Partie 3 – Intégration finale

Combinez les deux systèmes :

- Si la question concerne les documents → le pipeline RAG répond.
- Si elle peut être répondue par l'agent → l'agent choisit un outil ou répond directement.
- Sinon, conversation normale

Exemple :

Utilisateur : Quelle est la politique de congés de l'entreprise ?

→ RAG recherche la réponse dans les fichiers internes.

Utilisateur : Et quelle est la météo à Paris aujourd'hui ?

→ L'agent appelle l'outil météo.

Utilisateur: Bonjour

→ le LLM répond simplement (sans faire de RAG ni d'appels de tools)

### Partie 4

- Ajout d'une mémoire conversationnelle (suivi de contexte).
- Interface conversationnelle (Chainlit/Streamlit/Gradio).
- Ajout d'un outil de recherche web pour compléter les documents internes.

### Contraintes techniques

- LangChain ou LlamaIndex pour la gestion du RAG.
- OpenAI, Mistral ou HuggingFace pour le LLM.
- Minimum trois outils intégrés.
- Code versionné sur GitHub.
- Documentation claire : architecture, dépendances, instructions d'exécution.
- (Optionnel) Déploiement sur un cloud ou un hébergeur gratuit (Render, HuggingFace Space, Streamlit Cloud).

## **Pistes de sujets possibles**

- Assistant juridique sur des textes de loi PDF.
- Assistant académique sur les notes de cours.
- Agent médical simulé sur des fiches de symptômes.
- Chatbot météo + RAG sur un rapport climatique.
- Conseiller financier + RAG sur des rapports d'investissement.