

# Projet Noté : Développement d'un Agent IA avec LangChain et un LLM

BUT Informatique – 3<sup>e</sup> année

## Contexte du projet

Dans le cadre de ce projet, vous devrez concevoir et implémenter un **agent intelligent** utilisant le framework **LangChain** associé à un **modèle de langage (LLM)**. L'objectif principal est de développer un agent capable de répondre à plusieurs besoins concrets, en s'appuyant sur des outils variés et une logique de raisonnement multi-étapes.

Vous êtes libres de choisir le **thème** de votre agent (ex. : aide à la recherche, gestion de projet, accompagnement éducatif, support technique, planificateur de voyage, etc.).

## Objectif général

Développer un **agent autonome** (type ReAct) intégrant plusieurs outils (tools) afin de répondre à des besoins complexes. L'agent devra être accessible via une petite **interface graphique** développée avec **Gradio**.

## Attendus techniques

- **Spécification des besoins** : L'agent doit répondre à **au moins quatre besoins distincts**. Ces besoins doivent être clairement identifiés et décrits.
- **Complexité des besoins** : Chaque besoin doit être **complexe**, c'est-à-dire nécessitant la coordination de plusieurs outils et la mise en œuvre du paradigme **ReAct** (Reason + Act) avec une chaîne de pensée.
- **Utilisation d'outils multiples** : L'agent doit mobiliser **au moins six outils différents** (appel à des APIs externes, fonctions Python, etc.). Tous les besoins ne nécessitent pas forcément six outils, mais au total, au moins six outils distincts doivent être intégrés.
- **Mémoire et/ou RAG** : L'agent doit soit :
  - Intégrer un mécanisme de **mémoire** appropriée, **ou**
  - Utiliser une approche **RAG (Retrieval-Augmented Generation)** pour s'appuyer sur des données externes.
- **Interface utilisateur** : Une **interface Gradio** doit permettre d'interagir facilement avec l'agent (saisie d'une question, affichage de la réponse, etc.).

## Livrables attendus

### 1. Rapport technique en PDF

Ce rapport devra inclure :

- Une **description de l'objectif et de l'architecture** globale de l'agent ;
- La **spécification de chaque outil** : rôle, entrées, sorties, dépendances, appel à des APIs ou librairies externes ;
- Des **exemples concrets de besoins** traités par l'agent et la manière dont les outils collaborent pour y répondre. Des schémas seraient appréciés ;
- Une **liste de requêtes** pouvant être soumise à l'agent (au moins 10) afin de faire la démonstration de l'étendue de ses capacités ;
- Les **noms et prénoms des membres du groupe** ;
- Le **lien de partage du Google Colab** (en haut du rapport).

### 2. Jupyter Notebook hébergé sur Google Colab

Le projet doit être livré sous la forme d'un **notebook Google Colab**. Le lien de partage doit être inclus au début du rapport (attention à bien activer le partage par lien). Le notebook doit être directement exécutable : **la seule action à réaliser sera d'ajouter la clé d'API du correcteur**. Aucune erreur ne doit apparaître lors de l'exécution.

## Principaux critères d'évaluation

- Originalité du thème de l'agent ;
- Pertinence et clarté des besoins définis ;
- Cohérence de l'architecture et bon usage de LangChain ;
- Intégration correcte et variée des outils (au moins six mais plus serait apprécié) ;
- Mise en œuvre du paradigme ReAct et gestion de la mémoire ou du RAG ;
- Qualité de l'interface Gradio ;
- Fonctionnalité du notebook (exécution fluide et sans erreurs) ;
- Qualité du rapport (structure, complétude, clarté).

## Date et modalité de rendu

Le projet est à rendre pour le **vendredi 28 novembre 2025 à 23h59**. Tout retard entraînera une pénalité. Il doit être réalisé par groupes de 2. Le **rapport, contenant le lien vers le notebook**, devra être déposé sur la plateforme **eLearn**, dans l'espace dédié au projet, avant la date de rendu.

**Conseil** : commencez par définir l'objectif, les besoins de l'agent et les outils nécessaires avant de débuter le développement.

**Bon courage à toutes et à tous !**