

Synthèse du Projet : Création d'un jumeau numérique (Portfolio IA)

Réalisé par : Gaspar RIGOT

1. Objectif du projet

L'objectif de ce projet était de créer une application web interactive capable de répondre à des questions sur mon parcours professionnel, mes compétences et mes projets.

2. Architecture Technique

Pour réaliser ce projet, j'ai utilisé une architecture RAG (Retrieval Augmented Generation). C'est une technique qui permet de donner une mémoire personnelle à une IA.

Voici les outils que j'ai connectés ensemble :

- **Interface Web** : Développée en Python avec la bibliothèque Streamlit. C'est ce qui permet d'avoir le chat visuel.
- **Le Cerveau (LLM)** : J'utilise l'API d'OpenAI pour générer les phrases et comprendre le langage naturel.
- **La Mémoire Longue (Base Vectorielle)** : J'utilise Upstash Vector. C'est ici que sont stockées mes informations.
- **La Mémoire Court Terme (Historique)** : J'utilise Upstash Redis pour que le chatbot se souvienne du fil de la conversation pour ne pas qu'il oublie ce qu'on a dit deux minutes avant.

3. Stratégie de Gestion des Données (Data Management)

C'est la partie science des données du projet. Pour que l'IA réponde juste, il fallait que mes données soient propres. J'ai fait deux choix techniques importants pour la structuration de mon fichier source (profil.md).

A. Le choix du fichier unique (Centralisation)

Au début, on pourrait penser à faire plusieurs fichiers (un pour les études, un pour les expériences, un pour les passions). J'ai décidé de tout regrouper dans un seul fichier Markdown (**profil.md**).

Cela simplifie énormément la maintenance et l'indexation. Avec mon expérience de Data Steward chez Covea, je sais que la dispersion des données crée des erreurs. Ici, tout est centralisé quand le script d'indexation lit le fichier, il garde une cohérence globale. Si je mets à jour mon profil, je n'ai qu'un seul endroit à toucher, ce qui réduit le risque d'informations contradictoires. Même si vous nous avez conseillé de faire avec plusieurs fichiers j'ai préféré faire comme ça.

B. Le formatage en listes à puces (Optimisation pour l'IA)

J'ai rédigé mon profil principalement sous forme de listes à puces plutôt que de faire de longs paragraphes littéraires.

C'est une optimisation pour le moteur de recherche vectoriel (le RAG).

- **Clarté pour la machine** : Une phrase longue et complexe peut perdre l'IA sur le sens exact. Une liste est factuelle : Entreprise : Covea | Poste : Data Steward | Année : 2023.
- **Découpage (Chunking)** : Quand le script découpe mon fichier en petits morceaux pour la base de données, les listes se coupent plus proprement que les paragraphes. Cela évite que l'IA ne récupère qu'une moitié de phrase et raconte n'importe quoi. Cela permet au chatbot d'être beaucoup plus factuel et précis dans ses réponses.

4. Fonctionnement de l'application

Concrètement, voici ce qui se passe quand un utilisateur pose une question :

1. L'utilisateur demande : *"Quelles sont tes compétences en Python ?"*
2. L'application transforme cette question en vecteur (une suite de nombres).
3. Upstash Vector compare ces nombres avec ma base de données pour trouver le passage de mon profil.md qui parle de Python.
4. L'application envoie à OpenAI : *"Voici ce que l'utilisateur demande, et voici les infos que j'ai trouvées dans le CV de Gaspar. Formule une réponse."*
5. L'IA répond à l'utilisateur.
6. La réponse est sauvegardée dans Redis pour que la conversation reste fluide.

5. Bilan et compétences acquises

Ce projet m'a permis de mettre en pratique plusieurs compétences clés du BUT SD :

- **Programmation Python** (Scripts, API, Logique).
- **Manipulation de données textuelles** (Nettoyage, structuration Markdown).
- **Compréhension du Cloud** (Déploiement, gestion des clés API, variables d'environnement).
- **Gestion de projet** (Git, GitHub, versioning).