Documentation Technique - Chatbot Formations

1. Objectif du Projet

Ce projet a pour but de développer un assistant intelligent capable de recommander des formations pertinentes à partir d?une base de données structurée. Le backend utilise FastAPI, tandis que l?interface frontend est développée en Angular 19 avec des composants standalone. Les données sont vectorisées pour permettre une recherche sémantique via un LLM (Large Language Model) intégré dans un pipeline RAG (Retrieval Augmented Generation).

2. Backend (FastAPI)

Le backend repose sur FastAPI et propose une route POST `/query` qui reçoit une requête utilisateur et renvoie une réponse simulée ou issue d?un LLM.

- Fichier `main.py`: contient l?API FastAPI avec un endpoint `/query`.
- Fichier `test_query.py`: script pour tester les requêtes localement.

3. Frontend (Angular 19)

Le frontend utilise Angular 19 sans `app.module.ts`, uniquement des composants standalone :

- Composant principal : `ChatComponent`
- Fonctionnalité : permet de saisir une question, l?envoyer au backend, et afficher la réponse de l?assistant.

4. Pipeline de Traitement des Données

Le pipeline est découpé en plusieurs scripts permettant d'assurer une traçabilité claire des étapes :

- `main.py` : scraping des pages de formation.
- `clean.py` : nettoyage du contenu HTML, enrichissement des métadonnées.
- `prepare_vectorisation.py` : découpage des textes en chunks exploitables.
- `vectorize_chunks.py` : vectorisation et indexation dans une base ChromaDB ou FAISS.
- `README_generator.py` : génération automatique d?un fichier README récapitulatif des formations.
- `run_pipeline.py` : permet d'exécuter le pipeline étape par étape de manière interactive.

5. Structure du Dossier `content/`

- `json/formations/` : fichiers JSON bruts (resume_html inclus) et nettoyés.

- `csv/formations/` : fichiers CSV équivalents des JSON.
- `processed/chunks/` : chunks de texte prêts pour vectorisation.
- `vector_data/` : vecteurs calculés.
- `chroma_db/` : base de données vectorielle persistée.

6. Résumé HTML

Chaque formation contient un champ `resume_html` extrait de la page web d?origine. Il permet :

- D?avoir un aperçu riche (HTML) du contenu complet de la formation.
- De faciliter l?exploitation par un LLM pour des réponses plus pertinentes.
- Deux versions sont disponibles : brute (avec balises) et nettoyée (texte pur).

7. Travail Collaboratif

La partie RAG/LLM est développée par Mohammed. Michel s?occupe de l?extraction, du nettoyage, du pipeline et de l?interface utilisateur.

Le dossier `content/` regroupe tous les éléments nécessaires pour que Mohammed puisse commencer directement l?intégration LLM.