Rapport de Projet – Assistant Pédagogique IA basé sur LangChain, FAISS et FLAN-T5

**Intitulé du projet** : Assistant pédagogique basé sur l’IA pour répondre à des questions sur des cours universitaires.  
**Technologies utilisées** : LangChain, FAISS, HuggingFace, Streamlit, SQLite, OpenAI (optionnel).

1. **Objectif du projet**

Ce projet a pour but de développer une application web permettant à des étudiants de poser des questions sur leurs cours (algorithmique, programmation, mathématiques, etc.) et d’obtenir des réponses claires, contextualisées et fiables.  
L’outil repose sur l'approche **RAG (Retrieval-Augmented Generation)**, combinant :

* **Recherche sémantique** dans des documents indexés.
* **Génération de texte** à l’aide d’un modèle de langage.

1. **Architecture du système**
2. **Schéma global de l’architecture**

Utilisateur → [Streamlit UI] → [LangChain RAG] →

↳ Recherche vectorielle (FAISS + MiniLM)

↳ Génération de réponse (FLAN-T5 local → fallback GPT-3.5)

1. **Modules principaux**

|  |  |
| --- | --- |
| Composant | Fonction |
| Streamlit | **Interface utilisateur (entrée de question, affichage de réponse et résumé)** |
| LangChain | **Orchestration entre les composants de recherche et de génération** |
| HuggingFaceEmbeddings | **Génération d'embeddings via all-MiniLM-L6-v2** |
| FAISS | **Index vectoriel pour la recherche sémantique rapide** |
| FLAN-T5 (local) | **Génération de réponse initiale** |
| GPT-3.5 (OpenAI) | **Fallback si la réponse locale est insuffisante** |
| DeepTranslator | **Traduction automatique FR ↔ EN** |
| SQLite | **Sauvegarde des requêtes, réponses, résumés et thématiques détectées** |

1. **Fonctionnalités implémentées**

|  |  |
| --- | --- |
| Fonction | Description |
| Recherche intelligente | **L’outil extrait le contexte pertinent depuis les documents vectorisés.** |
| Génération de réponse | **FLAN-T5 génère une réponse à partir du contexte extrait.** |
| Support multilingue | **Traduction automatique si la question est en français.** |
| Résumé synthétique | **Une synthèse de la réponse est générée automatiquement.** |
| Classification par thème | **Détection simple du domaine (programmation, maths, algorithmique, etc.)** |
| Historique (SQLite) | **Les interactions sont stockées avec horodatage, thème, résumé et question.** |

1. **Démonstration fonctionnelle**

**Scénario 1** : question simple en français

* **Question** : *C’est quoi un algorithme ?*
* **Réponse** : *Un algorithme est une méthode pour résoudre un problème.*
* **Résumé** : *A method for resolving a problem.*
* **Thème détecté** : *Algorithmique*
* **Modèle utilisé** : FLAN-T5

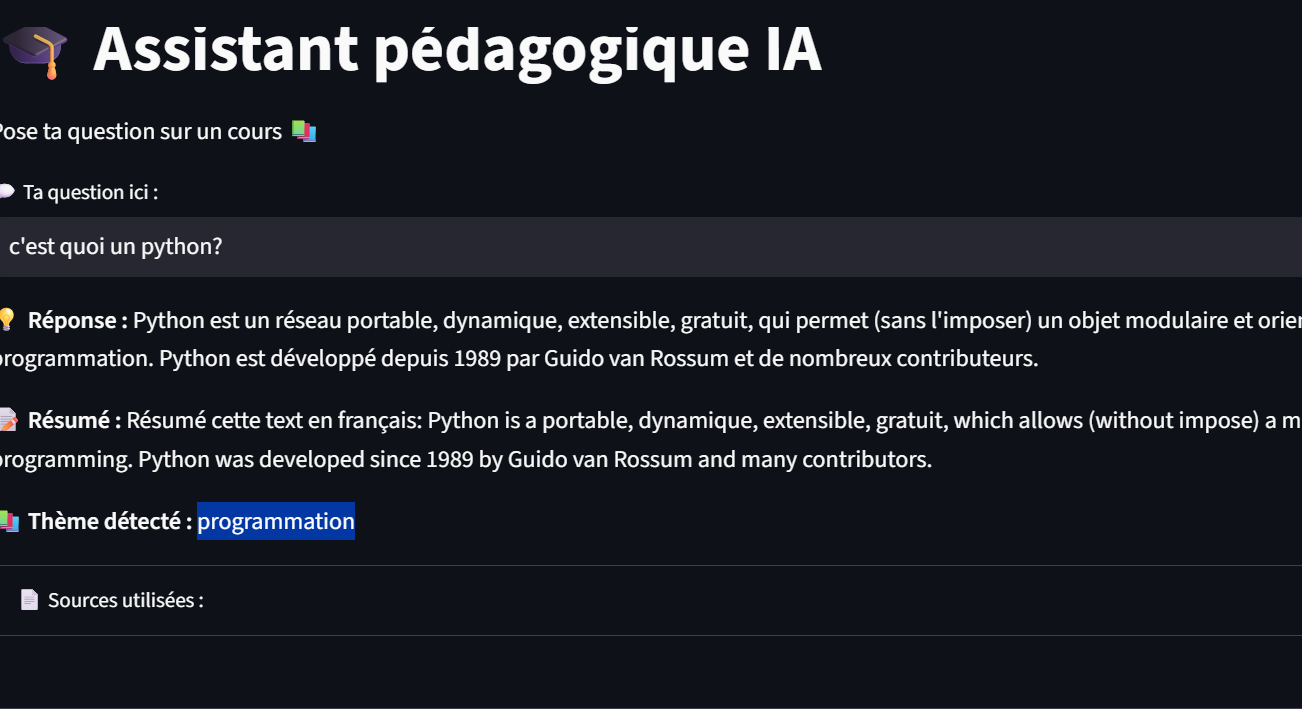
**Capture :**

****

**Scénario 2 : question programmation avec résumé**

* **Question** : *c'est quoi python?*
* **Réponse** : P*ython est un réseau portable, dynamique, extensible, gratuit, qui permet (sans l'imposer) un objet modulaire et orienté d'approche de la programmation.*
* **Résumé** : *Python is a portable, dynamique, extensible, gratuit, which allows (without impose) a modular and oriented approach to programming.*
* **Thème détecté** : *programmation*

**Capture:**



**Scénario 3 : question simple avec résumé**

* **Question** : *c'est quoi un programme?*
* **Réponse** : *Une liste de tâches que l'Ordinateur exécute*
* **Résumé** : *Une liste de tâches que l'Ordinateur exécute*
* **Thème détecté** : *Autre*

1. **Base de données SQLite (history.db)**

Chaque interaction utilisateur est sauvegardée automatiquement :

|  |  |  |
| --- | --- | --- |
| Champ | Type | Description |
| timestamp | TEXT | Date et heure de la requête |
| question | TEXT | Texte de la question |
| answer | TEXT | Réponse générée |
| summary | TEXT | Résumé synthétique de la réponse |
| topic | TEXT | Thématique détectée |

1. **Détail du code (fichier app.py)**

Voici un résumé des étapes importantes du code :

* Chargement du modèle FLAN-T5 local (transformers).
* Chargement de l’index FAISS contenant les documents vectorisés.
* Lorsqu’une question est posée :
  + Détection de langue → traduction vers EN si nécessaire.
  + Recherche vectorielle dans les documents.
  + Génération de réponse par FLAN-T5.
  + Fallback vers GPT-3.5 si la réponse est trop courte.
  + Résumé de la réponse.
  + Stockage de l’historique dans SQLite.
* Affichage clair dans Streamlit avec sections :
  + Réponse
  + Résumé
  + Thème
  + Sources

1. **Limites actuelles**

* Réponses de FLAN-T5 parfois trop courtes ou vagues.
* Traduction automatique basique (heuristique).
* Fallback GPT-3.5 nécessite une clé API valide.
* Thématique détectée via mots-clés uniquement.

1. **Perspectives d’évolution**

|  |  |
| --- | --- |
| Amélioration envisagée | Description |
| Fine-tuning du modèle local | **Adapter FLAN-T5 ou utiliser Mistral, Phi-2, etc.** |
| Classification thématique ML | **Utiliser un modèle supervisé pour détecter le thème.** |
| Tableau de bord analytique | **Visualiser les requêtes fréquentes via Streamlit ou Grafana** |
| Authentification utilisateur | **Ajouter un login pour usage multi-élèves** |
| Export PDF ou email de la réponse | **Génération automatique d’un rapport par réponse** |