

Rapport de Développement

Portfolio Chatbot IA

Auteur : Charles Auvrai

Date : Janvier 2026

Contexte : Projet (SAE) / Portfolio Data

1. Introduction

Ce rapport détaille la conception et le développement de mon application web interactive "Chatbot". L'objectif de ce projet technique est de mettre en valeur des compétences techniques propre à notre domaine. De par une interface dynamique capable de répondre aux requêtes d'un recruteur en temps réel.

Le projet démontre la mise en œuvre d'une architecture logicielle complète en Python, allant de l'intégration d'un modèle d'Intelligence Artificielle à la création d'une interface utilisateur web via le framework Streamlit.

2. Architecture et Choix Techniques

Le développement a été guidé par une exigence de robustesse et de maintenabilité. Le choix s'est porté sur une utilisation exclusive de python pour assurer une cohérence entre le traitement des données et l'affichage.

2.1. La Stack Technologique

Streamlit : Ce choix permet un prototypage rapide et une mise en production sécurisée, de plus nous n'avons pas vraiment travaillé sur streamlit jusque-là donc c'est aussi une découverte intéressante.

Gestion Asynchrone : Implémentation de la bibliothèque `asyncio`.

Les appels aux modèles d'IA (LLM) peuvent générer de la latence. L'asynchronisme permet de ne pas bloquer l'affichage pendant la génération de la réponse.

2.2. Gestion de l'État (Session State)

Contrairement à un site web classique, une application Streamlit recharge le script à chaque interaction. Pour maintenir le fil de la conversation, j'ai implémenté une persistance des données via le `Session State` :

L'interface globale s'articule autour d'une barre latérale (sidebar) qui centralise la gestion de la session. C'est ici que sont conservés l'historique complet des échanges entre l'utilisateur et l'assistant, ainsi que le compteur de requêtes garantissant le respect des quotas. Le système y mémorise également l'état actif de l'application, assurant une transition fluide entre le mode conversationnel et l'écran de fin.

3. Spécificités du Projet et Expérience Utilisateur (UX)

Ce projet se distingue par l'implémentation de logiques conditionnelles avancées qui guident l'utilisateur, au-delà d'un simple champ de texte.

3.1. Interface de Suggestion Dynamique

Pour orienter l'utilisateur et démontrer la capacité à structurer l'information, un système de boutons de suggestion a été développé nativement.

J'ai utilisé `st.columns` pour aligner quatre points d'entrée (Formation, Expériences, Identité, Passions).

Ces boutons injectent directement un prompt prédéfini dans le moteur de l'IA, facilitant l'interaction, l'utilisateur comprendra donc plus facilement et n'a pas besoin de se creuser la tête pour des réponses simples.

3.2. Mécanisme de Restriction (Crédits)

Une contrainte technique a été codée pour limiter la session à 5 interactions.

Mon Objectif est donc de forcer la fin de la discussion IA et d'agrandir potentiellement mon cercle de connaissance.

3.3. Bascule d'Interface (Le "Switch" Final)

L'application possède deux états distincts gérés par une condition logique (`if/else`). Lorsque la limite de requêtes est atteinte, l'application ne se contente pas de bloquer l'input, elle **bascule** vers une nouvelle interface.

Cette interface emmené l'utilisateur directement sur mes contacts linkedIn et mails.

4. Conclusion

La réalisation de ce Portfolio Chatbot valide plusieurs compétences techniques clés pour ma formation et ma futur intégration professionnel :

Le développement d'une application montre ma capacité à construire un produit fini et déployable.

Cela valorise aussi ma maîtrise de Python via l'utilisation de concepts "avancés" (Asyncio, Session State, Logique conditionnelle).

Intégration IA : Capacité à connecter une interface utilisateur à un moteur d'IA, compétence très importante aux vu des évolutions actuelles dans le monde du travail.