

Sujet Projet Faq Assited by IA : FAA

A réaliser en groupe. Attendu :

- Réalisation de l'application
- Rapport de conception
- Soutenance du projet.
- Travail possible en équipe.
- Déploiement.

Sujet de Projet Web : Implémentation d'un Site Web avec FAQ Intelligente et Administration

1. Introduction

Ce projet a pour objectif de développer un site web interactif en Python, intégrant une section "Foire Aux Questions" (FAQ) intelligente. Cette FAQ affichera des questions-réponses qui pourront être initialement générées et enrichies par une IA générative basée sur un modèle RAG (Retrieval Augmented Generation), à partir d'un corpus de documents PDF décrivant des formations. Un système d'administration complet permettra de gérer, modifier, supprimer et ajouter manuellement ces entrées de la FAQ. De plus, des fonctionnalités d'analyse et de prédiction des visites seront mises en place.

2. Objectifs du Projet

Les objectifs principaux de ce projet sont les suivants :

- Développement d'un site web robuste : Créer un site web fonctionnel et intuitif en utilisant un framework Python (Flask ou Django).
- FAQ Intelligente (RAG) : Implémenter une FAQ dont les réponses sont initialement générées ou assistées par un modèle RAG entraîné sur des documents PDF de descriptions de formations. Les utilisateurs consulteront une liste de questions/réponses prédéfinies.
- Gestion Administrative de la FAQ : Développer une interface d'administration sécurisée permettant aux administrateurs de créer, modifier, supprimer et visualiser les entrées de la FAQ (questions/réponses générées ou ajoutées manuellement).
- Intégration d'Ollama et d'un SML : Utiliser Ollama pour la gestion et l'inférence du modèle de langage et un Small Language Model (SML) pour l'assistance à la génération des réponses de la FAQ.

- Analyse et Prédiction des Visites : Mettre en place un système de journalisation des visites et de création d'un modèle de prédiction des visites basé sur le Machine Learning.

3. Fonctionnalités Détaillées

3.1. Partie Utilisateur (Frontend)

- Page d'accueil : Présentation du site, informations générales.
- Section FAQ :
 - Affichage d'une liste de questions/réponses prédéfinies.
 - Possibilité de filtrer ou de rechercher parmi les questions/réponses affichées (recherche textuelle classique sur les questions et réponses existantes).
- Page de contact.

3.2. Partie Administration (Backend)

- Tableau de bord : Vue d'ensemble de l'activité du site (statistiques de visites, etc.).
- Gestion des utilisateurs (administrateurs) : Création, modification, suppression des comptes administrateurs.
- Gestion de la FAQ :
 - Ajout manuel : Possibilité d'ajouter manuellement des paires question/réponse.
 - Modification : Modifier des questions et/ou leurs réponses.
 - Suppression : Supprimer des entrées de la FAQ.
 - Visualisation : Lister toutes les entrées de la FAQ avec leur source (manuelle ou initialement générée par l'IA).
 - Gestion des documents PDF : Interface pour uploader et gérer les fichiers PDF servant de corpus pour l'IA générative.
 - Génération assistée par l'IA : Un module permettra à l'administrateur de lancer un processus de génération de questions/réponses basé sur les PDF via l'IA. Les résultats seront ensuite soumis à l'administrateur pour validation, modification et ajout à la FAQ.
- Gestion des modèles IA : Possibilité de configurer ou de recharger le modèle de langage utilisé par l'IA.
- Statistiques et Logs :

- Consultation des logs d'accès au site (horodatage, IP, page visitée).
- Consultation des actions effectuées par les administrateurs (ajout/modification/suppression de FAQ).

3.3. Module d'Intelligence Artificielle (RAG) (pour l'administration)

- Traitement des PDF : Extraction de texte à partir des documents PDF de descriptions de formations.
- Indexation : Création d'un index vectoriel des connaissances extraites des PDF.
- Moteur de recherche sémantique : Utilisé côté administration pour identifier les passages pertinents en fonction de requêtes (ex: pour générer de nouvelles questions/réponses ou vérifier l'existence de réponses).
- Génération de réponses : Utilisation d'un Small Language Model (SML) via Ollama pour assister l'administrateur dans la génération de questions et de réponses cohérentes et pertinentes en se basant sur les passages récupérés. Cette génération se fera en arrière-plan ou via une interface spécifique pour les administrateurs, et non directement par les utilisateurs finaux.
- Fine-tuning (optionnel) : Possibilité de fine-tuner le SML sur des données spécifiques aux formations pour améliorer la pertinence.

3.4. Module d'Analyse et de Prédiction (Machine Learning)

- Journalisation des visites : Enregistrement de chaque visite sur le site (URL, horodatage, informations de l'utilisateur anonymisées).
- Modèle de prédiction des visites :
 - Utilisation de techniques de Machine Learning (e.g., séries temporelles, régression) pour prédire le nombre de visites futures sur le site.
 - Visualisation des prédictions dans le tableau de bord administrateur.

4. Requirements Techniques

4.1. Requirements Logiciels

- Système d'exploitation : Linux (Ubuntu/Debian recommandé), macOS, ou Windows.
- Langage de programmation : Python 3.9+
- Framework Web : Flask ou Django (choix à justifier, Django pour un projet plus conséquent avec administration intégrée, Flask pour une plus grande légèreté et personnalisation).

- Base de données : PostgreSQL (recommandé pour la robustesse et les fonctionnalités), ou SQLite (pour le développement local) ou SQL Server.
- Pour l'IA générative (côté administration) :
 - Ollama : Pour exécuter et gérer les modèles de langage locaux.
 - Bibliothèques Python pour le traitement du langage naturel (NLP) :
 - PyMuPDF ou pdfminer.six pour l'extraction de texte des PDF.
 - Langchain ou LlamaIndex pour l'implémentation du RAG (indexation, recherche, orchestration avec Ollama).
 - sentence-transformers pour la génération d'embeddings.
 - Un Small Language Model (SML) compatible avec Ollama (ex: Llama 3 mini, Phi-3, Gemma 2B, Mistral 7B quantisé).
- Pour le Machine Learning :
 - scikit-learn pour les algorithmes de prédiction.
 - pandas et numpy pour la manipulation de données.
 - matplotlib ou seaborn pour la visualisation.
- Pour la gestion des dépendances : pip et venv (environnement virtuel Python).
- Serveur Web : Gunicorn ou uWSGI (pour le déploiement en production avec Flask/Django).
- Serveur HTTP : Nginx ou Apache (en tant que reverse proxy en production).

4.2. Requirements Infrastructure (pour le déploiement)

- Serveur de développement : Machine locale (ordinateur portable/desktop) avec une configuration raisonnable (RAM, CPU) pour le développement.
- Serveur de production (minimum) :
 - CPU : 2-4 cœurs (selon la charge attendue).
 - RAM : 8 Go - 16 Go ou 32Go (essentiel pour Ollama et le modèle de langage, peut être plus si le SML est conséquent ou si plusieurs modèles sont chargés).
 - Disque : SSD de 50 Go - 100 Go (pour le système d'exploitation, le code, la base de données et les modèles Ollama qui peuvent être volumineux).
 - Connectivité : Accès Internet stable et port 80/443 ouvert pour le trafic web.

- GPU (recommandé pour l'inférence rapide du SML) : Une carte graphique compatible CUDA (NVIDIA) ou ROCm (AMD) avec au moins 4 Go de VRAM (8 Go ou plus est idéal pour des performances optimales avec certains SML). Si pas de GPU, l'inférence sera plus lente sur le CPU.
- (En option) Environnement de conteneurisation (recommandé) : Docker et Docker Compose pour faciliter le déploiement et la gestion des différents services (application web, base de données, Ollama).

5. Étapes de Réalisation (Aperçu)

1. Initialisation du projet : Choix du framework (Flask/Django), configuration de l'environnement virtuel.
2. Modélisation de la base de données : Définition des modèles pour la FAQ, les utilisateurs, les logs, etc.
3. Développement du Frontend : Création des interfaces utilisateur (HTML, CSS, JavaScript).
4. Développement du Backend (API) : Implémentation des vues, routes, gestion des requêtes.
5. Intégration d'Ollama et du SML (pour l'administration) :
 - Installation d'Ollama.
 - Téléchargement du SML choisi.
 - Implémentation du pipeline RAG (extraction PDF, indexation, recherche, génération assistée pour les administrateurs).
6. Développement du système d'administration : Interfaces CRUD pour la FAQ et autres entités, y compris le module de génération assistée par l'IA.
7. Mise en place de la journalisation des visites.
8. Développement du module ML pour la prédiction des visites : Collecte de données, entraînement du modèle, intégration.
9. Tests unitaires et d'intégration.
10. Déploiement : Configuration du serveur web, base de données, Ollama, et l'application Python.