

ÉCOLE NORMALE SUPÉRIEURE DE
L'ENSEIGNEMENT TECHNIQUE -
MOHAMMEDIA

RAPPORT DU PROJET TUTORÉ «SGBDR»
FILIÈRE : «BDCC & GLSID»

Ask.N7 : Chatbot universitaire



Réalisé par :

Mohamed Ed Deryouch
Noureddine Imgharn
Hanane Chaaouan
Sana Ait Braim

Encadré par :
Pre N. BENMOUSSA

Contents

1	Introduction	2
1.1	Contexte Général	2
1.2	Problématique	2
1.3	Objectifs du Projet	2
2	Technologies et Outils Choisis	3
2.1	Frontend (Interface Utilisateur)	3
2.2	Backend (API et Logique Métier)	3
2.3	Base de Données et Stockage	3
2.4	Infrastructure	4
3	Architecture et Conception	5
3.1	Architecture Globale (RAG)	5
4	Modélisation des Données	6
4.1	Modèle Conceptuel de Données (MCD)	6
4.2	Modèle Logique de Données (MLD)	8
5	Réalisation et Interfaces	10
5.1	Page d'Accueil (Landing Page)	10
5.2	Module d'Authentification	10
5.3	Interface de Chat (Expérience Étudiant)	12
5.4	Dashboard Administrateur (Back-Office)	12
6	Conclusion et Perspectives	14
6.1	Bilan	14
6.2	Perspectives d'Évolution	14
7	Bibliographie et Webographie	15
	Bibliographie et Webographie	15

1 Introduction

1.1 Contexte Général

Dans un environnement universitaire dynamique tel que l'ENSET Mohammedia, l'information est le moteur de la réussite des étudiants. Des calendriers d'exams aux procédures de stage, en passant par les règlements intérieurs et les notes administratives, des centaines d'informations critiques sont générées chaque semestre.

Cependant, l'accès à cette information reste un défi majeur. Les étudiants se retrouvent souvent confrontés à un écosystème fragmenté où les données sont dispersées entre des tableaux d'affichage physiques, des groupes WhatsApp non officiels, et divers sites web départementaux.

1.2 Problématique

Le problème principal identifié est la **surcharge et la fragmentation de l'information**. Nous avons identifié trois défis majeurs :

1. **Les silos d'information (Le "Cimetière des PDF")** : Les informations critiques sont souvent verrouillées dans des fichiers statiques (PDF scannés, images) qui ne sont pas consultables par mots-clés. Pour trouver une simple date d'examen, un étudiant doit parfois ouvrir dix fichiers différents.
2. **Le goulot d'étranglement administratif** : Le personnel administratif a des ressources limitées et des horaires de travail fixes. Ils passent un temps précieux à répondre aux mêmes questions répétitives, ce qui crée des files d'attente et laisse les étudiants sans réponse le soir ou le week-end.
3. **La propagation de rumeurs** : En l'absence d'information officielle facilement accessible, la désinformation se propage via les canaux non officiels, causant stress et confusion.

1.3 Objectifs du Projet

L'objectif de ce projet est de développer **Ask.N7**, un assistant numérique intelligent dédié aux étudiants et au personnel de l'ENSET.

Ask.N7 vise à agir comme un hub central pour toutes les connaissances universitaires, fournissant des réponses instantanées, précises et fiables 24h/24 et 7j/7. Contrairement à une recherche par mot-clé classique, Ask.N7 utilise

l'intelligence artificielle pour comprendre le langage naturel et fournir des réponses basées uniquement sur des documents officiels vérifiés.

2 Technologies et Outils Choisis

Pour répondre aux exigences de performance, de scalabilité et d'expérience utilisateur moderne, nous avons sélectionné une stack technique robuste divisée en plusieurs couches.

2.1 Frontend (Interface Utilisateur)

L'interface utilisateur a été conçue pour être intuitive et réactive.

- **React 18** : Bibliothèque JavaScript pour construire des interfaces utilisateur. React permet une gestion efficace de l'état de l'application et offre une architecture basée sur les composants.
- **TypeScript** : Superset de JavaScript qui ajoute le typage statique. Il améliore la maintenabilité du code et réduit les erreurs lors du développement.
- **Vite** : Outil de construction de nouvelle génération, choisi pour sa rapidité de démarrage et de rechargement à chaud (HMR).
- **Tailwind CSS** : Framework CSS "utility-first" permettant de styliser l'application rapidement, garantissant un design cohérent et responsive.
- **Shadcn UI** : Une collection de composants réutilisables construits avec Radix UI et Tailwind, assurant une esthétique professionnelle.

2.2 Backend (API et Logique Métier)

Le backend gère les requêtes des utilisateurs et l'interaction avec l'IA.

- **FastAPI** : Framework web moderne et rapide pour construire des API avec Python. Il est idéal pour l'IA grâce à sa gestion native de l'asynchronisme.
- **LangChain** : Framework utilisé pour orchestrer le processus RAG (Retrieval-Augmented Generation), liant les documents de l'université aux capacités de réponse de l'IA.
- **OpenAI API** : Fournit le moteur d'intelligence artificielle pour la compréhension du langage naturel.
- **Celery** : File de tâches asynchrone distribuée, utilisée pour traiter l'indexation des fichiers PDF en arrière-plan sans bloquer l'interface.

2.3 Base de Données et Stockage

La gestion des données est cruciale pour le système RAG.

- **PostgreSQL** : SGBD relationnel pour stocker les profils utilisateurs et l'historique des conversations.
- **Qdrant** : Base de données vectorielle essentielle pour la recherche sémantique. Les documents PDF y sont stockés sous forme de vecteurs.
- **MinIO** : Stockage d'objets (compatible S3) pour les fichiers bruts (PDFs, images).
- **Redis** : Utilisé comme cache et "Message Broker" pour Celery.

2.4 Infrastructure

- **Docker** : Pour la conteneurisation de chaque service.
- **Nginx** : Serveur web utilisé comme reverse proxy.

3 Architecture et Conception

3.1 Architecture Globale (RAG)

L'architecture de Ask.N7 repose sur le principe du **RAG (Retrieval-Augmented Generation)**. Cette approche permet de combiner la puissance des modèles de langage (LLM) avec la précision d'une base de connaissances propriétaire.

Voici le flux de données simplifié :

1. **Ingestion** : L'administrateur télécharge un document (PDF) sur le Dashboard. Le backend traite ce fichier, extrait le texte, le découpe en segments ("chunks") et le vectorise via l'API d'OpenAI.
2. **Stockage Vectoriel** : Ces vecteurs sont stockés dans **Qdrant**, une base de données vectorielle optimisée pour la recherche par similarité.
3. **Interrogation** : Lorsqu'un étudiant pose une question, le système convertit sa requête en vecteur mathématique.
4. **Récupération (Retrieval)** : Qdrant identifie les segments de documents les plus proches sémantiquement de la question posée.
5. **Génération** : Le système envoie au LLM (OpenAI) un prompt contenant la question de l'étudiant ainsi que les segments de documents trouvés. Le modèle génère alors une réponse basée *uniquement* sur ces sources.

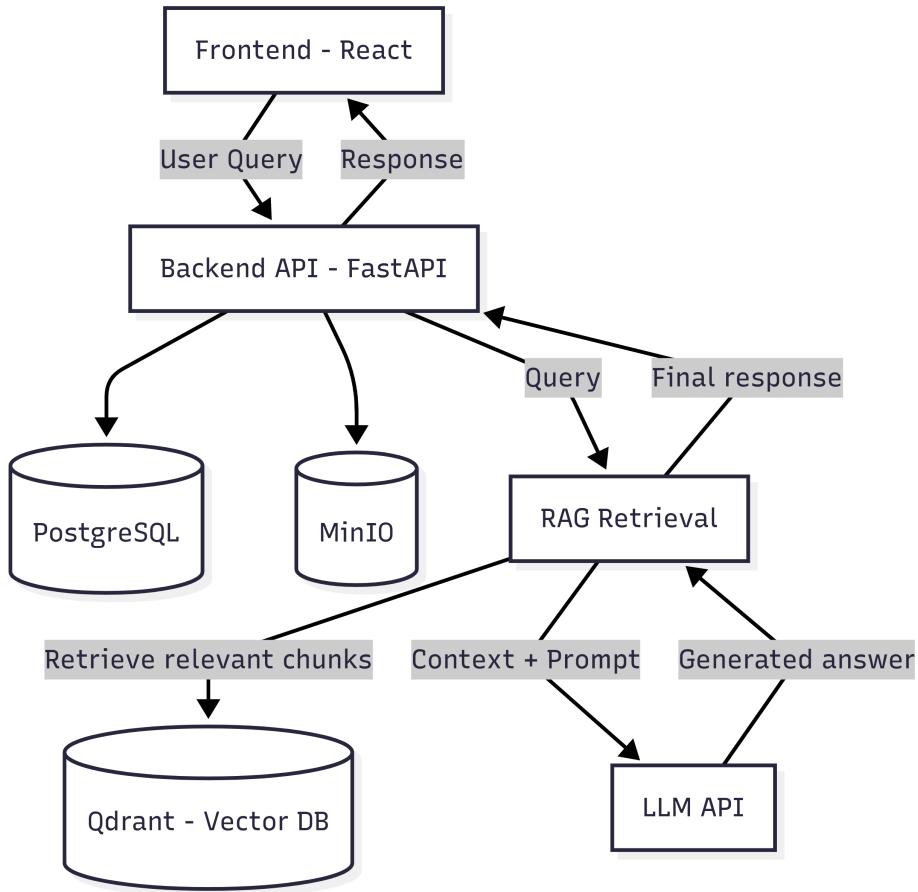


Figure 1: Architecture Globale

4 Modélisation des Données

Pour garantir la persistance et la cohérence des données, nous avons suivi une démarche de modélisation rigoureuse, passant du niveau conceptuel (MCD) au niveau logique (MLD).

4.1 Modèle Conceptuel de Données (MCD)

Le MCD (Figure ??) définit les entités du système et les relations sémantiques entre elles, indépendamment des contraintes techniques.

- **L'Utilisateur (Users)** : Entité centrale représentant les acteurs du système (étudiants et administrateurs). Un utilisateur peut initier plusieurs conversations.
- **La Conversation (Chat_Conversations)** : Représente une session d'échange. La cardinalité entre Utilisateur et Conversation est de type (0,N), car un utilisateur peut avoir plusieurs historiques de chat.

- **Le Message (Chat_Messages) :** Contient le texte de la question et de la réponse. Il est lié strictement à une conversation unique (1,1).
- **Le Document (Document_Uplets) :** Représente les fichiers sources (PDF) ajoutés par les administrateurs pour nourrir l'IA.

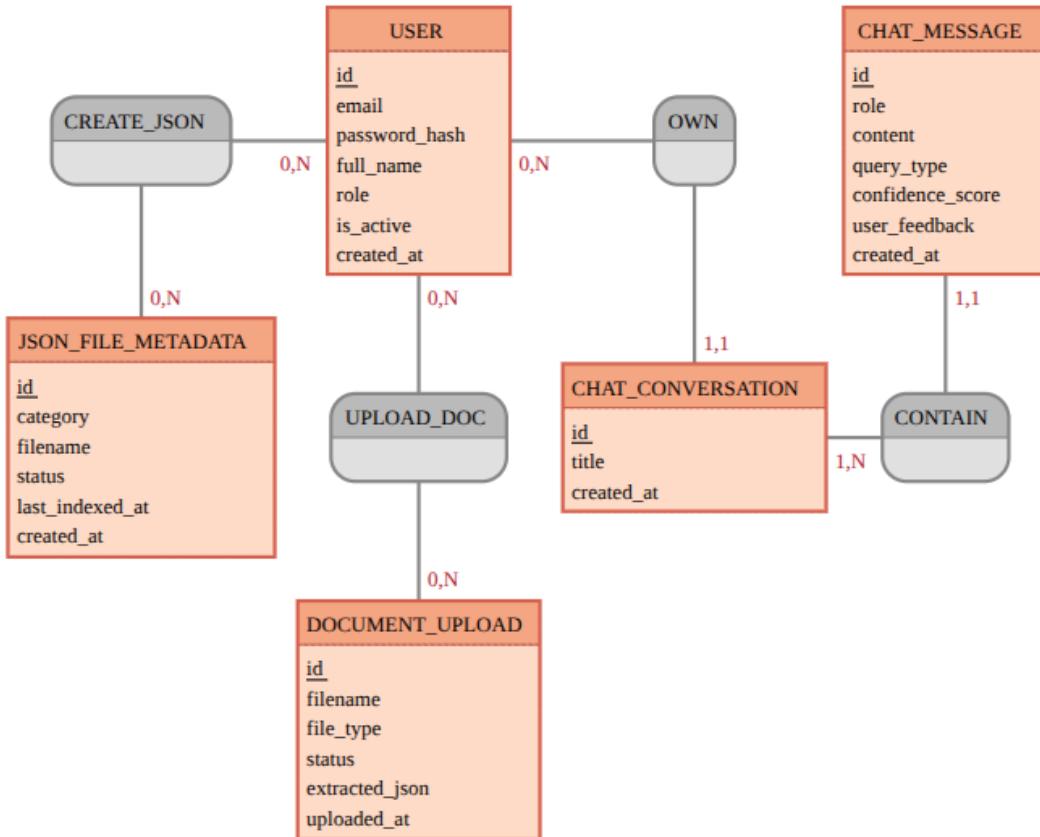


Figure 2: Modèle Conceptuel de Données (MCD)

Les principales règles de gestion illustrées sont :

- **Gestion des Utilisateurs :** L'entité USER est centrale. Elle possède des attributs d'identification (email, password_hash) et de rôle (role).
- **Historique de Chat :**
 - La relation **OWN** (Posséder) lie un utilisateur à ses conversations avec une cardinalité (0,N), signifiant qu'un utilisateur peut avoir plusieurs conversations, mais qu'une conversation appartient à un seul utilisateur.
 - La relation **CONTAIN** (Contenir) structure le dialogue : une CHAT_CONVERSATION est composée de plusieurs (1,N) CHAT_MESSAGE.
- **Gestion Documentaire :** Les entités DOCUMENT_UPLOAD et JSON_FILE_METADATA sont liées à l'utilisateur via les relations UPLOAD_DOC et CREATE_JSON, permettant de tracer qui a ajouté chaque ressource dans le système.

4.2 Modèle Logique de Données (MLD)

Le MLD (Figure ??) traduit le modèle conceptuel en un schéma relationnel implantable dans PostgreSQL. C'est ici que sont définis les types de données et les clés étrangères.

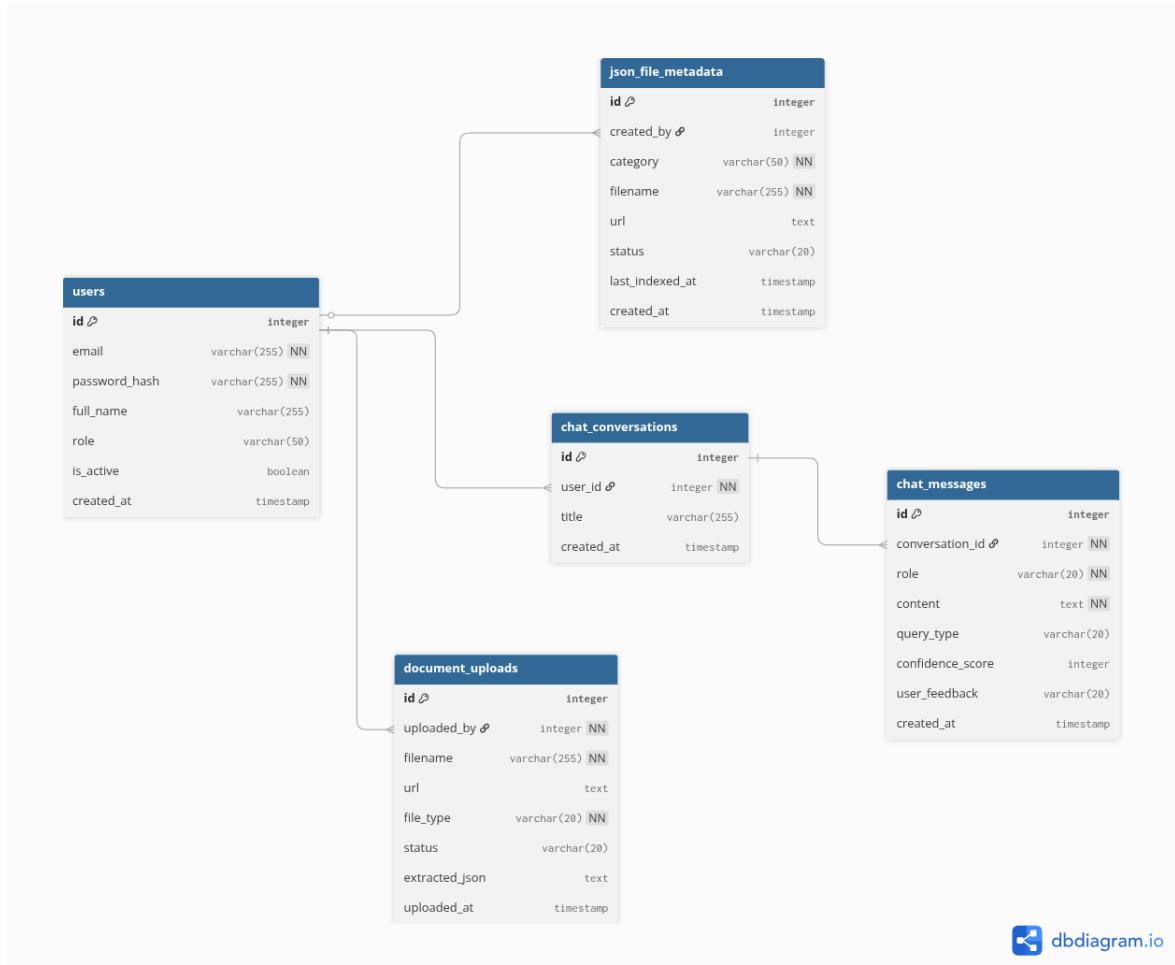


Figure 3: Modèle Logique de Données (MLD)

La traduction du MCD vers le MLD a suivi les règles suivantes :

1. **Clés Étrangères (Foreign Keys)** : Les relations (1,N) ont été transformées par l'ajout de clés étrangères du côté du "plusieurs" :

- La table `chat_conversations` intègre la colonne `user_id` (type `integer`) qui référence la table `users`. Cela permet d'associer chaque session à son propriétaire.
- La table `chat_messages` intègre `conversation_id`, créant le lien hiérarchique entre le fil de discussion et le message individuel.
- La table `document_uploads` intègre `uploaded_by` pour la traçabilité administrative.

2. **Typage des Données** : Nous avons sélectionné des types SQL précis pour optimiser le stockage :

- **VARCHAR(255)** : Utilisé pour les champs textuels courts comme les emails, les titres ou les noms de fichiers (`filename`).
- **TEXT** : Utilisé pour le contenu des messages (`content`) ou le JSON extrait (`extracted_json`), car ces champs peuvent contenir une grande quantité de données sans limite fixe.
- **TIMESTAMP** : Appliqué à tous les champs `created_at` pour horodater précisément chaque action.

5 Réalisation et Interfaces

Cette section détaille les interfaces utilisateur finales de l'application Ask.N7, mettant en évidence le soin apporté à l'expérience utilisateur (UX) et à l'interface (UI).

5.1 Page d'Accueil (Landing Page)

La page d'accueil sert de point d'entrée pour tous les visiteurs. Elle a été conçue pour inspirer confiance et expliquer immédiatement l'utilité de l'outil.



Figure 4: Page d'accueil présentant la proposition de valeur de Ask.N7

5.2 Module d'Authentification

Pour garantir la sécurité et la personnalisation des échanges, l'accès à l'assistant nécessite une authentification.

Création de Compte (Inscription)

Le formulaire d'inscription permet aux nouveaux utilisateurs (étudiants) de créer leur espace. Il demande les informations essentielles (Prénom, Nom, Email) et impose une vérification du mot de passe pour éviter les erreurs de saisie.

Créer un compte

First Name

Last Name

Email

Password

Confirm Password

[Create an account](#)

Vous avez déjà un compte? [Connectez-vous ici](#)

Figure 5: Formulaire d’inscription pour les nouveaux utilisateurs

Connexion

L’interface de connexion est épurée, demandant l’email (ou nom d’utilisateur) et le mot de passe. C’est la porte d’entrée sécurisée vers l’assistant.

Connectez-vous à votre compte

Email ou UserName

Password

[Connexion](#)

Nouvel utilisateur? [Registre](#)

Figure 6: Interface de Connexion sécurisée

5.3 Interface de Chat (Expérience Étudiant)

C'est le cœur de l'application. L'interface reprend les codes des messageries modernes pour une prise en main immédiate.

- **Sidebar (Gauche) :** Permet de naviguer dans l'historique des conversations passées ou d'en créer une nouvelle.
- **Zone de Chat (Centre) :** Affiche le dialogue. Les réponses de l'IA sont formatées clairement.

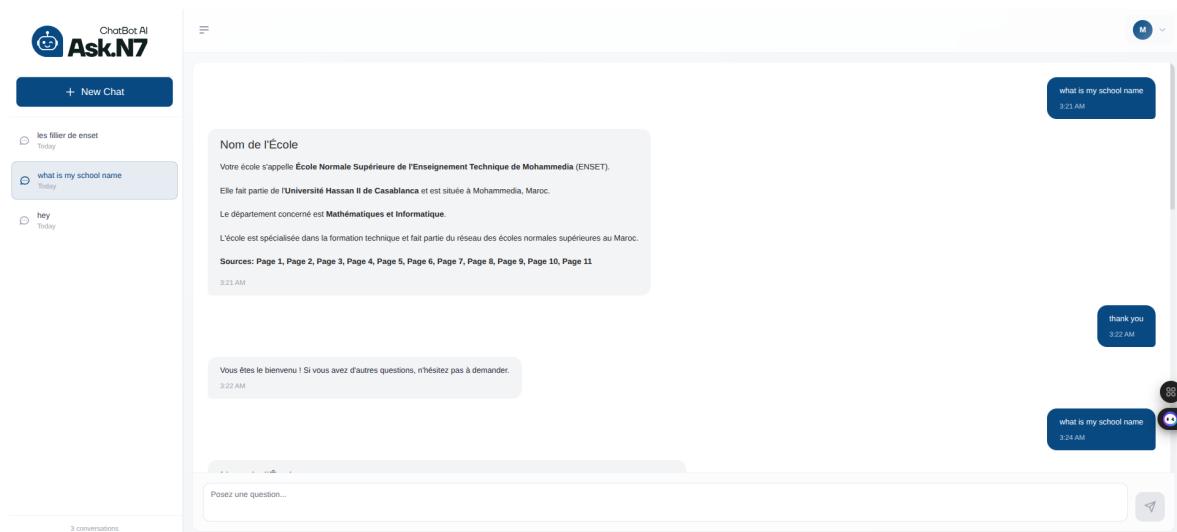


Figure 7: Interface de conversation montrant une réponse sourcée

5.4 Dashboard Administrateur (Back-Office)

Cette interface est réservée au personnel de l'ENSET. Elle permet de gérer la base de connaissances qui alimente l'IA.

- **Upload :** Bouton permettant d'ajouter de nouveaux documents PDF ou Markdown.
- **Suivi :** Liste des fichiers avec leur statut d'indexation (ex: "Ready"). Cela permet de vérifier que les documents sont bien pris en compte par le système RAG.

The screenshot shows the 'Knowledge Base' section of the AskN7 Admin Dashboard. At the top, there's a header with the AskN7 logo, 'ChatBot AI', 'Admin Dashboard', and a 'Back to Chat' link. Below the header, the title 'Knowledge Base' is displayed, followed by the sub-instruction 'Manage documents for the shared knowledge base'. There are three tabs at the top: 'All' (selected), 'MD', and 'PDF'. A large blue button labeled '+Upload Document' is positioned on the right. The main area lists two documents:

- pdf** 1 ready
mi.emploi.general.25-26.v2.0.pdf (PDF, 418.2 KB • Uploaded on 15-Jan-2028) - Status: Ready
- md** 1 ready
enset.md (MD, 32.8 KB • Uploaded on 15-Jan-2028) - Status: Ready

A small circular icon with a question mark is located in the bottom right corner of the dashboard area.

Figure 8: Dashboard de gestion des documents (Base de Connaissance)

6 Conclusion et Perspectives

6.1 Bilan

Le projet Ask.N7 a permis de réaliser une solution concrète à un problème récurrent au sein de l'ENSET : la difficulté d'accès à l'information. En centralisant les documents administratifs et pédagogiques au sein d'une architecture RAG, nous avons transformé des "silos de PDF" en une base de connaissances interactive.

Les objectifs principaux ont été atteints :

- **Accessibilité** : L'information est disponible 24/7 sans dépendre des horaires administratifs.
- **Fiabilité** : Les réponses sont générées exclusivement à partir de documents officiels, réduisant la propagation de rumeurs.
- **Efficacité** : L'interface intuitive permet de gagner du temps tant pour les étudiants que pour l'administration.

Sur le plan technique, ce projet a permis de maîtriser une stack moderne (FastAPI, React, Vector DB) et de mettre en œuvre des concepts avancés d'Intelligence Artificielle générative appliquée.

6.2 Perspectives d'Évolution

Bien que fonctionnelle, l'application peut encore être améliorée pour offrir une expérience plus riche :

1. **Application Mobile Native** : Développer une version mobile (React Native ou Flutter) pour permettre aux étudiants de recevoir des notifications push.
2. **Mode Vocal** : Intégrer la reconnaissance vocale (Speech-to-Text) et la synthèse vocale pour rendre l'application plus accessible.
3. **Intégration Multi-Canal** : Connecter le bot à des plateformes existantes comme Microsoft Teams ou WhatsApp Business API.
4. **Analyse des Tendances** : Développer un module analytique pour l'administration qui identifie les questions les plus fréquemment posées.

7 Bibliographie et Webographie

Documentation Technique

- **FastAPI Documentation**

<https://fastapi.tiangolo.com/> - Consulté pour la structure de l'API et la gestion asynchrone.

- **React et Vite Documentation**

<https://react.dev/> et <https://vitejs.dev/> - Consultés pour le développement frontend et l'optimisation du build.

- **LangChain Python Docs**

<https://python.langchain.com/> - Ressource principale pour l'implémentation de la chaîne RAG et la gestion des documents.

- **Qdrant Vector Database**

<https://qdrant.tech/documentation/> - Utilisé pour comprendre l'indexation vectorielle et la recherche par similarité.

- **Tailwind CSS**

<https://tailwindcss.com/> - Référence pour les classes utilitaires de style.

Outils de Conception

- **dbdiagram.io** - Utilisé pour la modélisation de la base de données (MCD/MLD).

- **Figma** - Utilisé pour le prototypage des interfaces utilisateurs.