



ÉCOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE ET D'ANALYSE DES  
SYSTÈMES - RABAT

Filière : Business Intelligence & Analytics

---

*Rapport de projet de Data Retrieval :*  
**Recherche Sémantique dans le Saint Coran**

---

*Réalisé par :*

Chams Doha BEKKAI

Yahya BELGHITI-ALAOUI

Amin BENALI

Oumaima GHAZOUAN

Omar YOUNG

*Encadré par :*

Pr. Nouredine KERZAZI

*Année universitaire 2024/2025*

*Rapport de projet de Data Retrieval :*  
Recherche Sémantique dans le Saint Coran

# Table des matières

<b>Introduction Générale</b>	<b>3</b>
<b>1 Contexte et objectifs du projet</b>	<b>4</b>
1.1 Contexte du projet et défis . . . . .	4
1.2 Objectifs du projet . . . . .	4
<b>2 Conception du Système</b>	<b>6</b>
2.1 Introduction . . . . .	6
2.2 Besoins Fonctionnels . . . . .	6
2.3 Besoins Non Fonctionnels . . . . .	6
2.4 Architecture RAG . . . . .	7
2.5 Choix Technologiques . . . . .	8
2.5.1 Modèle d'embedding . . . . .	8
2.5.2 Vector Store . . . . .	8
2.5.3 Modèle Grand du langage(LLM) . . . . .	9
2.5.4 Interface utilisateur . . . . .	10
2.5.5 Environnement de développement . . . . .	10
2.6 Conclusion . . . . .	10
<b>3 Implémentation</b>	<b>11</b>
3.1 Alimentation de la base de données vectorielles . . . . .	11
3.1.1 Prétraitement des données . . . . .	11
3.1.2 Alimentation . . . . .	11
3.2 Le flux d'exécution . . . . .	11
3.3 Interface utilisateur . . . . .	12
<b>4 Requêtes et résultats</b>	<b>14</b>
<b>Conclusion générale et perspectives</b>	<b>16</b>

# Table des figures

2.1	Architectire du système . . . . .	7
2.2	logo Chroma vector store . . . . .	9
2.3	Caractéristiques de gemini-2.0-flash-exp . . . . .	9
2.4	Gemini logo . . . . .	9
2.5	Google Colab logo . . . . .	10
3.1	Requete d'amélioration . . . . .	12
3.2	Prompt LLM . . . . .	12
3.3	Flux d'exécution . . . . .	12
3.4	Interface utilisateur . . . . .	13
3.5	Interface utilisateur . . . . .	13
4.1	Requete Paradis Hadith . . . . .	14
4.2	Requete Enfer Hadith . . . . .	15
4.3	Requete Paradis Quran . . . . .	15
4.4	Suite Requete Paradis Quran . . . . .	16

# Introduction Générale

Dans le cadre de ce projet, nous avons développé un moteur de recherche dédié au Coran et aux Hadiths, visant à offrir une expérience utilisateur intuitive et efficace pour la recherche et l'exploration de ces textes fondamentaux de l'islam. Ce livrable documente le processus de conception, de développement et de mise en œuvre des différentes étapes techniques et fonctionnelles de ce projet.

L'objectif principal est de fournir un guide complet permettant de reproduire, d'exécuter et d'exploiter le moteur de recherche. Ce document se structure autour des thématiques suivantes :

- Mise en place de l'environnement et des dépendances.
- Traitement et indexation des données.
- Développement et déploiement de l'interface utilisateur.

Ce moteur repose sur des outils modernes, tels que Chroma et Gemini pour la recherche textuelle et la génération de réponses, Python pour le traitement des données, et Streamlit pour l'interface utilisateur. Cette documentation est donc un support essentiel pour comprendre et maîtriser les étapes de mise en œuvre de ce projet.

# Chapitre 1

## Contexte et objectifs du projet

### 1.1 Contexte du projet et défis

La numérisation croissante des textes religieux et leur accessibilité sur des plateformes en ligne soulignent le besoin d'outils performants pour la recherche textuelle. Les recherches au sein du Coran et des Hadiths nécessitent souvent une précision et une rapidité qui dépassent les capacités des outils traditionnels.

Les travaux concernant le traitement de la langue arabe sont peu nombreuses. De plus, les textes arabes posent des défis uniques en raison de leur structure linguistique complexe, de la richesse sémantique et des nuances dans l'interprétation. Par ailleurs, les systèmes actuels (ChatGPT, Llama, ...) génèrent des réponses très générales et parfois fausses quand la requête de l'utilisateur est écrite en langue arabe. Les réponses deviennent encore plus approximatives quand la requête de l'utilisateur se rapporte à la religion.

Ces spécificités et défis exigent l'utilisation de technologies avancées, comme le RAG, combinées à des techniques de prétraitement linguistique adaptées.

### 1.2 Objectifs du projet

Le projet vise à :

- 1. Concevoir une infrastructure robuste :** Mettre en place un environnement capable de gérer efficacement des volumes importants de données issues du Coran et des Hadiths, en permettant un accès rapide et une recherche pertinente.
- 2. Préparation des données :** Créer des représentations vectorielles des textes du Coran et des Hadiths à l'aide de modèles d'embedding, afin de faciliter la recherche sémantique et la récupération d'informations pertinentes.

**3. Utilisation d'un modèle de langage (LLM) :** Intégrer un LLM pour répondre aux questions des utilisateurs en se basant sur les représentations vectorielles des deux corpus (Coran et Hadiths), en générant des réponses adaptées au contexte de la requête.

**4. Proposer une interface utilisateur intuitive :** Développer une interface utilisateur permettant une recherche simple, rapide et ciblée dans les deux corpus (Coran et Hadiths), avec un affichage clair des réponses contextuelles générées par le modèle de langage.

**5. Garantir l'extensibilité et la répliquabilité :** Documenter les étapes pour faciliter l'installation et l'utilisation sur d'autres machines, tout en permettant des améliorations futures et l'ajout de nouveaux corpus ou modèles.

## Chapitre 2

# Conception du Système

### 2.1 Introduction

Dans ce chapitre, nous détaillons les besoins fonctionnels et non fonctionnels du système, ainsi que l'architecture du pipeline RAG (Retrieval-Augmented Generation) personnalisée ainsi que les choix technologiques pour la partie réalisation.

### 2.2 Besoins Fonctionnels

1. **Recherche Personnalisée** : Offrir à l'utilisateur la possibilité de sélectionner une source de données spécifique (Coran ou Hadiths).
2. **Amélioration des Requêtes** : Utiliser un modèle LLM pour reformuler automatiquement les requêtes des utilisateurs, en améliorant leur clarté et précision.
3. **Extraction d'Informations Pertinentes** : Récupérer les passages les plus pertinents à partir de la base vectorielle, en fonction de la requête utilisateur.
4. **Génération de Réponse** : Générer une réponse contextuelle et cohérente en utilisant les données récupérées.
5. **Affichage Intuitif** : Présenter les réponses via une interface conviviale et simple à utiliser.

### 2.3 Besoins Non Fonctionnels

1. **Rapidité** : Le système doit fournir une réponse en temps réel avec une latence minimale.



2. **Précision** : Assurer une correspondance précise entre la requête utilisateur et les passages extraits.
3. **Extensibilité** : Intégrer facilement de nouvelles données ou technologies.
4. **Fiabilité** : Maintenir une disponibilité constante et un fonctionnement stable.
5. **Simplicité d'Utilisation** : Garantir une accessibilité même pour les utilisateurs ayant peu de compétences techniques.

## 2.4 Architecture RAG

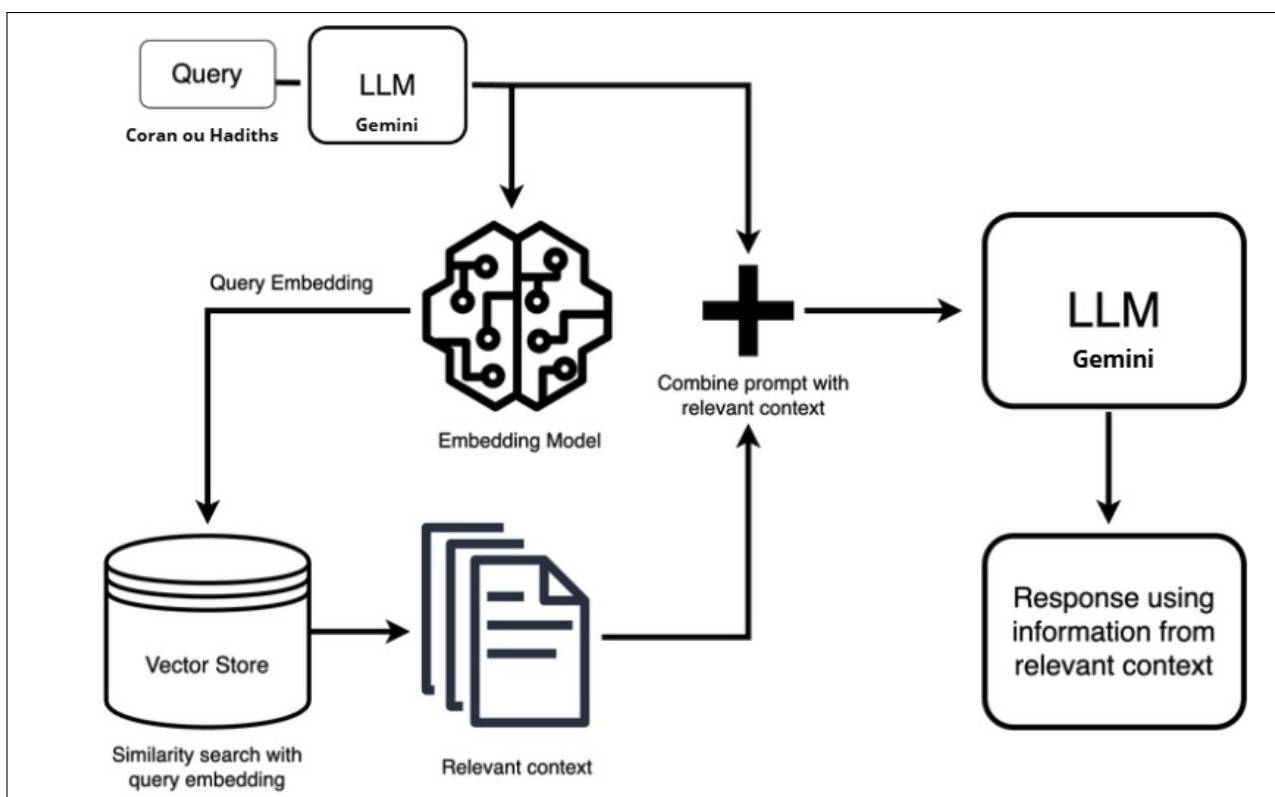


FIGURE 2.1 – Architecture du système

Description du Pipeline :

### 1. Entrée de l'Utilisateur

L'utilisateur saisit une requête textuelle et choisit une source d'information parmi deux options : textes du Coran ou des Hadiths.

### 2. Amélioration de la Requête

La requête est analysée et reformulée pour être plus claire, précise et adaptée au système. Cette reformulation prend en compte le contexte et le contenu attendu.

### 3. Représentation de la Requête sous Forme Numérique

La requête reformulée est transformée en une représentation numérique (vecteur) afin de permettre une comparaison avec les données stockées.

### 4. Recherche d'Informations Pertinentes

Le système compare le vecteur de la requête avec les représentations numériques des textes disponibles dans la base de données. Les passages les plus similaires et pertinents sont extraits.

### 5. Génération de Réponse

Les informations extraites sont utilisées pour produire une réponse complète et contextuelle en langage naturel, répondant directement à la requête de l'utilisateur.

### 6. Affichage de la Réponse

La réponse générée est présentée à l'utilisateur dans une interface claire et organisée, en indiquant la source des informations (Coran ou Hadiths).

## 2.5 Choix Technologiques

### 2.5.1 Modèle d'embedding

Les embeddings des documents seront créés à l'aide du modèle `intfloat/multilingual-e5-large` (lien Hugging Face). Ce modèle est particulièrement adapté pour la création d'embeddings des textes arabes, comme le montre l'article intitulé "Semantic Embeddings for Arabic Retrieval Augmented Generation (ARAG)" qui a mis en évidence les performances exceptionnelles de la famille de modèles E5 (notamment l'E5-ML-Large) dans le cadre de la recherche d'information sémantique pour la langue arabe.

Lien de l'article : <https://thesai.org/Publications/ViewPaper?Volume=14&Issue=11&Code=IJACSA&SerialNo=135>

### 2.5.2 Vector Store

Les embeddings des corpus coranique et hadith seront enregistrés dans une base de données vectorielle avec le texte correspondant à chaque vecteur et ses métadonnées. Dans ce contexte, la technologie open-source la plus connue pour réaliser cette tâche est Chroma. Elle permet de

gérer efficacement de grandes quantités de données vectorielles, offrant des fonctionnalités pour la recherche rapide et la récupération d'informations basées sur la similarité des vecteurs.



FIGURE 2.2 – logo Chroma vector store

### 2.5.3 Modèle Grand du langage(LLM)

L'utilisation d'un modèle grand de langage vise à générer des réponses adaptées aux requêtes des utilisateurs, en s'appuyant sur un contexte extrait de la base de données vectorielle. Il est essentiel que ce modèle soit capable de générer des réponses en arabe et d'accepter des entrées volumineuses afin de fournir un contexte suffisant pour améliorer la précision des réponses. Le modèle de langage **Gemini gemini-2.0-flash-exp** répond à ces exigences, en permettant son utilisation via une API gratuite pour une intégration facile. Avec une capacité de traiter jusqu'à 10 requêtes par minute et de prendre en input plus qu'un million de tokens, ce modèle présente des caractéristiques particulièrement adaptées à ce projet.


 Token limits <sup>[*]</sup>	Input token limit 1,048,576	Output token limit 8,192
 Rate limits <sup>[**]</sup>	<ul style="list-style-type: none"><li>• 10 RPM</li><li>• 4 million TPM</li><li>• 1,500 RPD</li></ul>	

FIGURE 2.3 – Caractéristiques de gemini-2.0-flash-exp



FIGURE 2.4 – Gemini logo

#### 2.5.4 Interface utilisateur

Streamlit sera utilisé pour développer une interface utilisateur interactive et intuitive. Cette interface va permettre aux utilisateurs de saisir des requêtes, de sélectionner la source des données à récupérer (Coran ou Hadiths) et d'afficher les résultats.

#### 2.5.5 Environnement de développement

Google Colab sera utilisé pour héberger et exécuter le serveur créé à l'aide de la bibliothèque **FastAPI** de Python. Ce serveur traitera les requêtes et retournera les résultats pour l'affichage sur l'interface. Pour accéder à ce serveur, un tunnel **ngrok** sera créé afin de générer une URL pour accéder aux endpoints du serveur. De plus, Google Colab servira également d'environnement de développement pour tester les différentes étapes du pipeline, telles que l'amélioration des requêtes ou la recherche de similarité.



FIGURE 2.5 – Google Colab logo

## 2.6 Conclusion

Ce chapitre a défini le pipeline et les besoins du système, assurant ainsi une interaction fluide et précise entre l'utilisateur et les données. Il présente également les technologies choisies ainsi que les justifications derrière ces choix.

## Chapitre 3

# Implémentation

### 3.1 Alimentation de la base de données vectorielles

#### 3.1.1 Prétraitement des données

La tâche de la récupération sera réalisée sur des versets coraniques ainsi que des paroles du prophète Mohammed (paix et bénédictions sur lui). Afin d'assurer une gestion optimale des données, celles-ci doivent être préalablement traitées séparées en deux documents indépendants. Cette étape ayant déjà été réalisée dans le cadre de notre projet précédent, nous disposons des données traitées et prêtes à l'emploi dans les fichiers suivants : **quran.json** et **json-hadit.json**.

#### 3.1.2 Alimentation

Nous alimentons notre Chroma vector store à l'aide des deux fichiers mentionnés. Nous créons deux collections distinctes : l'une pour les embeddings des hadiths et l'autre pour les embeddings du Coran. Ces embeddings sont générés à l'aide du modèle **intfloat/multilingual-e5-large**, en se basant sur les champs **aya\_chakl** pour les documents de **quran.json** et **text\_ar** pour les documents de **json-hadit.json**.

### 3.2 Le flux d'exécution

Une fois la requête saisie dans l'interface, elle sera envoyée au serveur créé sur Google Colab avec FastAPI à l'endpoint `/query`. Cet endpoint va améliorer la requête en utilisant un modèle LLM de Gemini avec la commande suivante :

```
prompt = f"enhance this user query in arabic for better information retrieval in arabic,  
and give only the enhanced query : \n{data.query}"
```

FIGURE 3.1 – Requete d'amélioration

Après cette amélioration, le traitement continue par une recherche de similarité cosinus sur l'embedding de la requête améliorée. Ensuite, un appel est effectué vers l'endpoint du LLM pour générer la réponse à cette requête :

```
prompt = f"المقدمة:\n{source_data}\nالسؤال:\n{user_query}\nالجواب:"
```

FIGURE 3.2 – Prompt LLM

Finalement, le contexte et la réponse du LLM seront retournés et affichés sur l'interface.

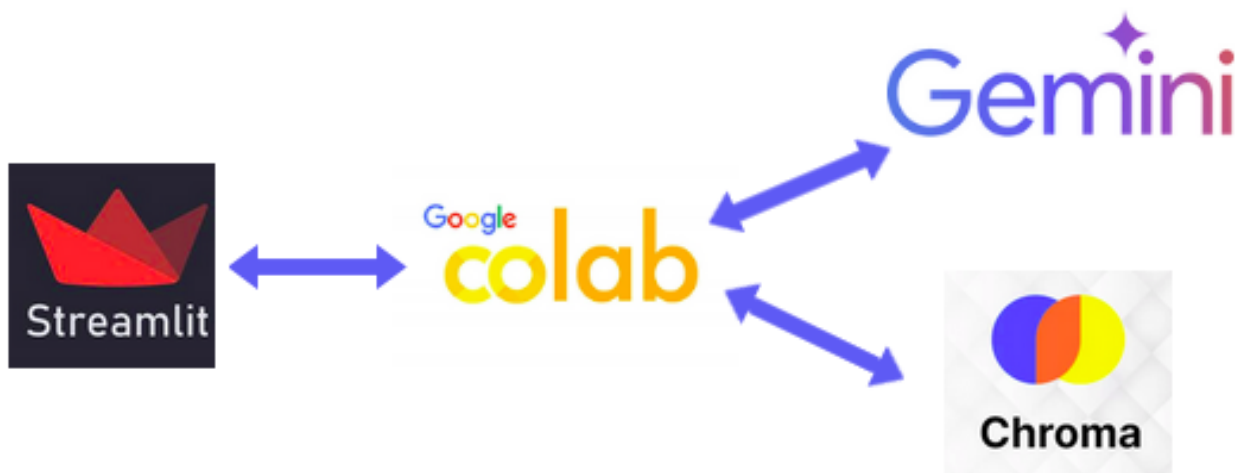


FIGURE 3.3 – Flux d'exécution

### 3.3 Interface utilisateur

L'interface utilisateur est créée pour être conviviale et pour permettre de choisir de quelle collection la récupération sera réalisée. Elle est conçue pour montrer deux colonnes pour la réponse, une pour le contexte récupéré et l'autre pour la réponse générée de LLM.



## Chapitre 4

# Requêtes et résultats

Requête Hadith : الجنة في الإسلام

Informations Récupérées (2 exemples)

1. **Source :** صحيح مسلم

**Chapitre :** كتاب الجنة وصفة نعيمها وأهلها

**Texte :** حدثني زهير بن حرب، حدثنا عبد الرحمن بن مهدي، حدثنا حماد بن سلمة، عن ثابت، عن أبي رافع، عن أبي هريرة عن النبي صلى الله عليه وسلم قال من يدخل الجنة ينعم لا يبأس لا تبلى ثيابه ولا يفنى شبابه

2. **Source :** صحيح البخاري

**Chapitre :** كتاب التوحيد

**Texte :** حدثنا أبو اليمان، أخبرنا شعيب، حدثنا أبو الزناد، عن الأعرج، عن أبي هريرة أن رسول الله صلى الله عليه وسلم قال إن لله تسعة وتسعين اسما مائة إلا واحدا، من أحصاها دخل الجنة

Texte Généré par le LLM (seulement le premier paragraphe)

الجنة في الإسلام هي دار الخلود والنعيم الأبدى التي أعدها الله تعالى لعباده المؤمنين، حيث ينعمون بلا بؤس ولا يشيخون أبدًا. من يدخل الجنة ينعم لا يبأس ولا تبلى ثيابه ولا يفنى شبابه. وقد ذكر النبي صلى الله عليه وسلم أن من أحصى أسماء الله الحسنى دخل الجنة.

FIGURE 4.1 – Requete Paradis Hadith



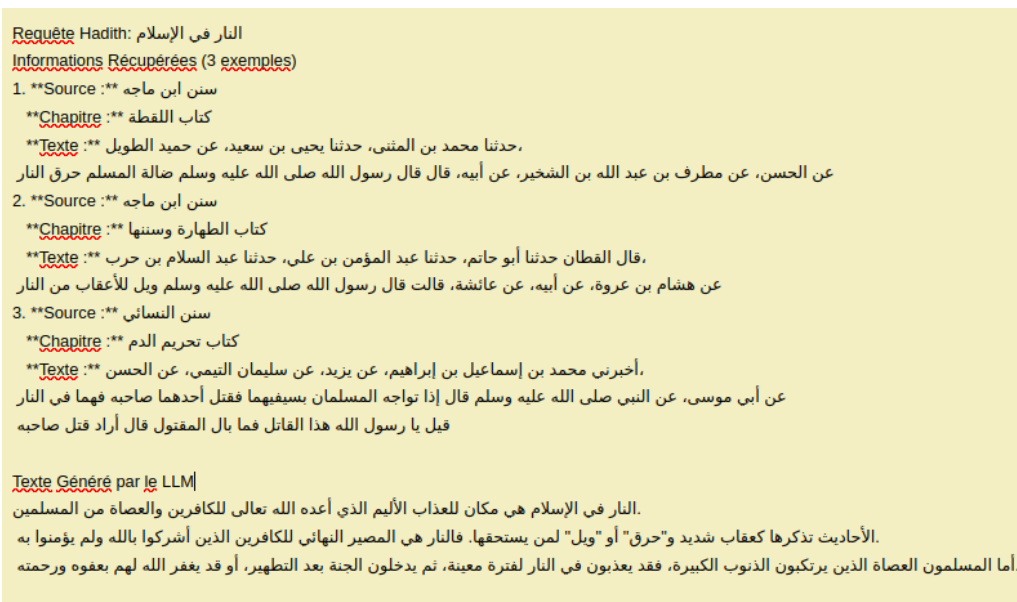


FIGURE 4.2 – Requete Enfer Hadith

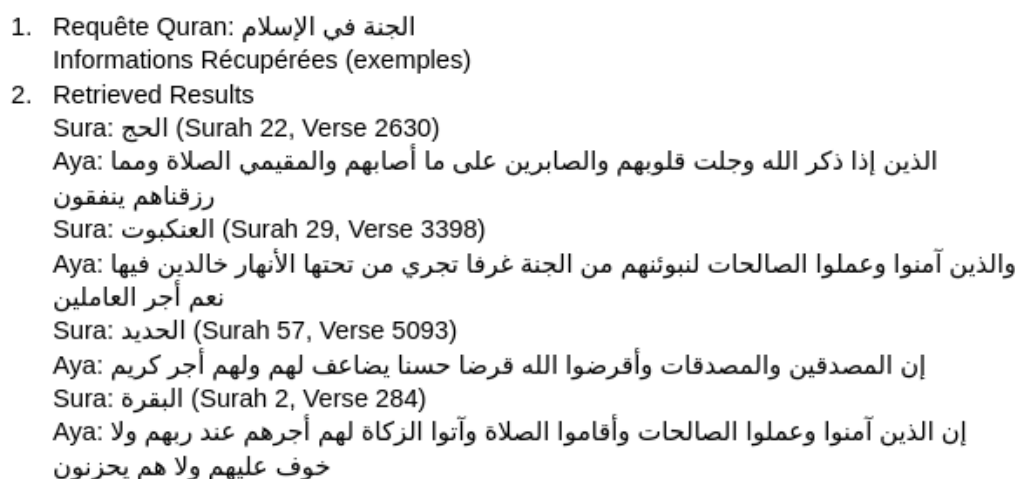


FIGURE 4.3 – Requete Paradis Quran

### 3. Generated Response

بال تأكيد، سأقدم لك شرحاً مفصلاً حول الجنة في الإسلام، مستنداً إلى المعلومات الواردة في النصوص القرآنية التي قدمتها، وإلى المعرفة العامة حول هذا الموضوع: صفات الجنة

يمكن تلخيص أبرز صفات الجنة التي وردت في القرآن والسنة كما يلي: دار الخلد: الجنة هي دار البقاء الأبدي، لا موت فيها ولا فناء. قال تعالى: "خَالِدِينَ فِيهَا أَبَدًا" (النساء: 57).

النعيم المقيم: تتميز الجنة بالنعيم الكامل والمتجدد، فلا يوجد فيها أي نقص أو ألم أو حزن. قال تعالى: "لَهُمْ فِيهَا مَا يَشَاءُونَ خَالِدِينَ" (الفرقان: 16).

الغرف العالية: ذكر القرآن وجود غرف عالية في الجنة. كما ورد في سورة العنكبوت: "لَنَبْوِّثَهُمُ مِنَ الْجَنَّةِ عُرُفًا يَجْرِي مِنَ تَحْتِهَا الْأَنْهَارُ" وهذا يدل على علو مكانتها ورفعة أهلها. الأنهار الجارية: تجري في الجنة أنهار من ماء غير آسن، ولبن لم يتغير طعمه

وخمر لذة للشاربين، وعسل مصفى. قال تعالى: "فِيهَا أَنْهَارٌ مِّنْ مَّاءٍ غَيْرِ آسِنٍ وَأَنْهَارٌ مِّنْ لَّبَنٍ لَّيِّمٍ يَّتَغَيَّرُ طَعْمُهُ وَأَنْهَارٌ مِّنْ خَمْرٍ لَّذَّةٍ لِلشَّارِبِينَ وَأَنْهَارٌ مِّنْ عَسَلٍ مُّصَفًّى" (محمد: 15).

الأشجار والظلال: تتميز الجنة بأشجارها الكثيفة وظلالها الوارفة، وهي أشجار لا تشبه أشجار الدنيا، كما ورد في كثير من النصوص الشرعية.

المساكن الطيبة: تتميز بمساكن طيبة ومريحة، وأهل الجنة يتنقلون بين هذه المساكن كيفما شاؤوا.

الجمال والبهاء: هي دار جمال وبهاء، ليس فيها شيء مكروه أو قبيح، بل كلها حسنة ومبهجة للناظرين.

السلام والأمن: لا يوجد في الجنة خوف أو حزن، بل السلام والأمن والطمأنينة. قال تعالى: "لَا يَمَسُّهُمْ فِيهَا تَصَبُّ وَمَا هُمْ مِنْهَا بِمُخْرَجِينَ" (الحجر: 48).

الرؤية الكريمة: من أعظم النعيم في الجنة هو رؤية الله تعالى، وهو فضل عظيم يختص به أهل الجنة.

FIGURE 4.4 – Suite Requete Paradis Quran

# Conclusion générale et perspectives

Ce projet de construction d'un système RAG (Retrieval-Augmented Generation) sur les corpus Hadith et Coranique a permis de développer une solution efficace pour la recherche sémantique et la génération de réponses par un modèle de langage (LLM). Grâce aux bases vectorielles et à l'IA générative, le système offre des réponses contextuelles et adaptées aux requêtes des utilisateurs. Cette approche peut garantir une compréhension approfondie des textes sacrés, tout en permettant la génération de réponses nouvelles et inattendues, en raison de la forte capacité des LLM à raisonner et à comprendre les nuances des textes. Si ce projet est réalisé correctement en optimisant les performances du système RAG, il pourrait ouvrir la voie à de nouvelles découvertes, enrichir l'étude des textes religieux et proposer des applications innovantes.