

Cahier des charges

ASADI : ASsistant ADministratif Intelligent

Groupe:

L3W1

Encadrant:

David Janisek

Auteurs:

Djaffer Abdel Malik, Gourmelen Thomas, Ponnoussamy Valentin, Traore Ali

Version du document:

1.0

Résumé

Dans le cadre de l'UE Projet Informatique à l'université Paris Cité, nous devons par équipes de quatre tenir un projet tout le long du semestre 6 sous la tutelle d'un encadrant. Ce projet vise à développer un assistant numérique capable de répondre automatiquement aux questions administratives, d'expliquer des procédures et d'offrir un apprentissage interactif via des scénarios immersifs, en s'appuyant sur une base documentaire centralisée. Le présent cahier des charges rédigé en accord avec l'encadrant pose les bases du projet. Il se veut descriptif des objectifs et ambitions de l'application et explique la démarche et le déroulement prévu pour la réalisation du projet.

Sommaire

1	Introduction	3
1.1	Contexte	3
1.2	Technologies	3
1.3	Historique	4
2	Description de la demande	5
2.1	Objectifs	5
2.2	Produit du projet	5
2.3	Fonctions du projet	5
2.4	Critères d'acceptabilité et de réception	7
3	Contraintes	9
4	Déroulement du projet	10
4.1	Planification	10
4.2	Ressources	11
5	Annexes	13
6	Glossaire	14
7	Références	14
8	Index	15

1 Introduction

1.1 Contexte

Les procédures administratives et réglementaires peuvent être complexes et parfois difficiles à comprendre, tant pour les nouveaux collaborateurs que pour les utilisateurs ayant besoin d'informations claires et accessibles. Un assistant numérique, capable de répondre automatiquement à des questions administratives, d'expliquer des procédures et de proposer un mode d'apprentissage interactif, contribuerait à simplifier ces tâches tout en favorisant une intégration plus rapide des nouveaux membres d'une équipe.

En s'appuyant sur un ensemble de ressources documentaires (fichiers, textes réglementaires, sites web, etc.), cet assistant permettrait de centraliser et de rendre accessible une base de connaissances, tout en offrant un apprentissage ludique et personnalisé grâce à des scénarios interactifs et des questions adaptées.

1.2 Technologies

Ce projet repose sur l'utilisation de plusieurs technologies permettant de concevoir une application web intégrant un assistant intelligent basé sur l'intelligence artificielle.

1.2.a Backend et Intelligence Artificielle

Pour le développement du backend et de la partie liée à l'intelligence artificielle, nous utiliserons Python. Ce langage de programmation est en effet largement employé dans les domaines du traitement des données et du machine learning. De plus, il offre un grand nombre de bibliothèques spécialisées dans ces domaines.

Pour simplifier le développement, nous adopterons le framework Django. Ce dernier facilite la gestion de la connexion entre le frontend et le backend, tout en permettant un développement plus rapide, sécurisé et facilement maintenable.

1.2.b Gestion et traitement des documents

La gestion et le traitement des documents impliquent l'utilisation de plusieurs technologies différentes :

- **Pipeline de traitement des documents** : Un pipeline est mis en place pour traiter des fichiers non structurés (PDF, images, documents Word) et les convertir en données exploitables. Ce processus suit plusieurs étapes : importation, extraction de texte, nettoyage, structuration et indexation.
- **Reconnaissance optique de caractères (OCR)** : La bibliothèque Tesseract est utilisée pour extraire le texte à partir de documents complexes, tels que des PDF ou des images.
- **RAG (Retrieval-Augmented Generation)** : Le RAG est une approche qui combine un moteur de recherche de documents avec un modèle de langage génératif, afin de fournir des réponses précises et contextualisées. Cette technologie permet de rechercher les informations pertinentes dans la base de documents, puis de générer des réponses en s'appuyant sur un modèle de langage.

1.2.c Base de données vectorielle

La base de données vectorielle remplace une base SQL traditionnelle pour la gestion des contenus et des requêtes documentaires. Cette technologie permet de stocker et d'indexer les vecteurs extraits des documents et des requêtes des utilisateurs, facilitant une recherche rapide et précise des informations pertinentes.

- ▶ **Milvus** : Milvus est une base de données vectorielle open source conçue pour les applications GenAI.
- ▶ **Faiss** : Faiss est une bibliothèque pour la recherche de similitude efficace et le regroupement de vecteurs denses. Il contient des algorithmes qui recherchent dans des ensembles de vecteurs de toute taille.

1.2.d Interface utilisateur

- ▶ **HTML/CSS** : L'interface utilisateur reposera sur des technologies classiques du développement web.

1.3 Historique

L'histoire de l'intelligence artificielle commence en 1950 avec l'idée qu'une machine pourrait penser par elle-même. Cette idée a été introduite par Alan Turing.

Le premier tournant de l'intelligence artificielle intervient en 2010 avec le traitement automatique du langage naturel. Cette évolution a été rendue possible grâce à l'essor du *Deep Learning*. Certaines innovations, comme *Word2Vec*, ont permis de représenter les mots sous forme de vecteurs, établissant ainsi une relation contextuelle.

Les modèles de langage modernes, tels que GPT, ont permis la génération de texte. Cependant, le principal défaut de ces nouveaux modèles est leur imprécision factuelle.

Pour pallier ce défaut, la méthode RAG a été développée par *Facebook AI* en 2020. Cette méthode a permis de combiner un moteur de recherche documentaire avec un modèle de génération de texte.

2 Description de la demande

2.1 Objectifs

L'objectif du projet ASADI est de concevoir une application web intégrant une intelligence artificielle capable de :

- ▶ Répondre à des questions administratives en exploitant diverses sources documentaires (textes, PDF, sites web, etc.) et en citant ses sources.
- ▶ Expliquer les procédures à l'aide d'explications détaillées, de guides étape par étape et de supports visuels (schémas, graphiques).
- ▶ Proposer un mode d'apprentissage interactif, basé sur des scénarios immersifs, des quiz et un système de suivi personnalisé, afin de renforcer les connaissances des utilisateurs.

ASADI vise à simplifier l'accès aux informations administratives en centralisant les démarches et en facilitant leur assimilation, notamment pour les nouveaux collaborateurs.

2.2 Produit du projet

ASADI est une application web qui utilise une IA avancée pour optimiser la gestion et la compréhension des démarches administratives. Elle repose sur un moteur de génération augmentée par récupération (RAG) permettant d'exploiter diverses ressources (textes, PDF, DOCX, sites web, images, etc.) afin de générer des réponses fiables et contextualisées.

Les principales fonctionnalités incluent :

- ▶ ChatBot IA intelligent :
 - Réponses aux questions administratives basées sur une base de connaissances centralisée.
 - Citations systématiques des sources pour garantir la transparence.
 - Prise en charge de multiples formats d'entrée (texte, documents, etc.).
- ▶ Mode d'apprentissage interactif :
 - Quiz dynamiques pour évaluer la compréhension des utilisateurs.
 - Scénarios pratiques où l'utilisateur applique ses connaissances.
 - Feedback personnalisé en fonction des performances.
- ▶ Interface ergonomique et sécurisée :
 - Création et gestion de comptes utilisateurs.
 - Navigation fluide avec une organisation claire des informations.
- ▶ Gestion évolutive du contenu :
 - Mise à jour et enrichissement de la base de connaissances avec de nouvelles ressources.
 - Ajout et modification des documents de référence.

L'architecture modulaire d'ASADI permettra l'intégration de nouvelles fonctionnalités à l'avenir.

2.3 Fonctions du projet

Le produit ASADI (ASsistant ADministratif Intelligent) repose sur plusieurs fonctions principales, conçues pour répondre aux besoins des utilisateurs en matière d'assistance administrative et d'apprentissage interactif. Ces fonctions sont détaillées ci-dessous :

1. Fonction de réponses aux questions administratives :

- ▶ L'objectif de cette fonctionnalité est de fournir des réponses précises et documentées aux questions de l'utilisateur.
- ▶ Description :
 - L'IA analyse la requête de l'utilisateur et recherche les informations correspondantes dans une base de connaissances (textes réglementaires, documents PDF, sites web...).
 - Elle génère une réponse synthétique et structurée, en citant les sources utilisées pour garantir la fiabilité.
 - Possibilité d'ajouter du contenu multimédia (ex : images, liens externes) pour illustrer la réponse.

2. Fonction d'explication des procédures administratives :

- ▶ L'objectif de cette fonctionnalité est de guider l'utilisateur dans la compréhension et l'exécution des démarches administratives.
- ▶ Description :
 - Présentation des étapes détaillées des procédures (ex : demande de visa, inscription à un service public...).
 - Explication claire avec des instructions pas à pas.
 - Intégration d'illustrations et de schémas interactifs pour une meilleure compréhension.
 - Possibilité d'afficher des cas spécifiques en fonction des besoins de l'utilisateur.

3. Fonction d'apprentissage interactif :

- ▶ L'objectif de cette fonctionnalité permet de tester les utilisateurs et de renforcer leurs connaissances administratives.
- ▶ Description :
 - Propose des quiz interactifs pour vérifier la compréhension des informations.
 - Génère des scénarios administratifs réalistes où l'utilisateur doit prendre des décisions.
 - Offre un feedback personnalisé en fonction des réponses de l'utilisateur.

4. Fonction de gestion et mise à jour de la base de connaissances :

- ▶ L'objectif de cette fonctionnalité est de maintenir une base de connaissances toujours à jour avec les dernières réglementations. Nous laisserons la possibilité d'ajout, de suppression et de modification seulement aux administrateurs désignés, excluant ainsi les utilisateurs lambda.
- ▶ Description :
 - Collecte et analyse de nouvelles ressources documentaires pour enrichir le contenu.
 - Possibilité d'ajouter et modifier les documents de référence uniquement par les administrateurs, garantissant ainsi l'intégrité et la fiabilité des informations contenues dans la base de connaissances.

5. Fonction de création et de connexion de compte

- ▶ L'objectif de cette fonctionnalité permet à un utilisateur de s'inscrire et de se connecter pour accéder aux fonctionnalités du site.
- ▶ Description:
 - L'accès au site est réservé aux utilisateurs connectés.
 - **Création de compte :**
 - L'utilisateur doit fournir un identifiant unique et un mot de passe sécurisé.
 - Une validation du mot de passe (critères de complexité) peut être requise.
 - Un email de confirmation peut être envoyé pour activer le compte.
 - **Connexion (Administrateur compris) :**
 - L'utilisateur saisit son identifiant et son mot de passe pour accéder à son compte.
 - Une option « Mot de passe oublié » permet de réinitialiser le mot de passe via email.

2.4 Critères d'acceptabilité et de réception

La validation du projet ASADI repose sur un ensemble de critères garantissant la conformité du produit aux attentes définies. Ces critères s'appliquent à la fois aux fonctionnalités de l'application, à sa performance, et à son ergonomie.

1. Critères d'acceptabilité

L'application ASADI sera considérée comme conforme aux attentes si elle satisfait les critères suivants :

- ▶ **Pertinence et exactitude des réponses :**
 - Le programme doit systématiquement fournir une réponse à toute requête de l'utilisateur.
 - La réponse doit être pertinente, c'est à dire que la réponses doit être :
 - Adaptée à la question posée.
 - Basé sur des informations issues de la base de connaissances et des documents référencés.
 - Accompagnée de la citation de source afin de garantir la fiabilité de la réponse.
- ▶ **Suivi de la progression de l'utilisateur :**
 - L'application doit permettre d'enregistrer et d'analyser l'évolution de l'utilisateur dans le mode d'apprentissage interactif.
 - Le suivi inclut :
 - Historique des quiz et scénarios réalisé.
 - Affichage des scores et résultats obtenus.
- ▶ **Historique des interactions :**
 - L'application doit permettre d'enregistrer les interactions entre l'utilisateur et l'IA.
 - Cet historique doit permettre :
 - De consulter les questions et les réponses fournies.
 - D'offrir à l'utilisateur la possibilité de supprimer ou de télécharger son historique.
- ▶ **Compatibilité avec divers formats de documents :**
 - L'application doit être capable d'exploiter tout type de documents administratifs, notamment :
 - Texte brut : .txt.
 - Documents bureautiques : .pdf,.docx,.xlsx.
 - Formats web : HTML
 - Images : contenant du texte (via OCR)
 - Le programme doit pouvoir analyser et extraire les informations pertinentes de ces documents sans altération du contenu original.

► **Ajout des documents dans la base de connaissances :**

- L'utilisateur doit pouvoir :
 - Ajouter de nouveaux documents à la base de connaissances.
 - Supprimer des documents existants.
- La base de connaissances doit être évolutive et permettre l'intégration automatique ou manuelle de nouvelles ressources.

► **Ergonomie et expérience utilisateur :**

- L'interface doit être intuitive, permettant une navigation fluide et claire.
- Les temps de chargement des réponses ne doivent pas être longues en condition normales d'utilisation.(au plus 10 secondes)

2. Critères de réception

La réception de l'application sera validée après une série de tests garantissant son bon fonctionnement et sa conformité :

► **Test fonctionnels :**

- **Réponses de l'IA** : Doit fournir une réponse pertinente avec citation des sources
- **Suivi de l'utilisateur** : Enregistrement et affichage des résultats des quiz et scénarios.
- **Historique des interactions** : Accès et suppression possibles.
- **Gestion des documents** : Ajout et exploitation correcte des nouveaux fichier.

► **Tests de compatibilité et performance :**

- **Comptabilité** : L'IA doit traiter correctement les formats .pdf, .docx, .xlsx, .txt, HTML.
- **Performance** : Le temps de réponse ne doit pas dépasser au plus 10 secondes.

► **Tests de sécurité :**

- **Authentification** : Vérification de la gestion des comptes et des accès.
- **Protection des données** : Sécurisation des documents et suppression sécurisée des données.

Une fois ces critères remplis, la recette finale sera validée.

3 Contraintes

Dans le cadre du développement de notre assistant numérique basé sur la technologie RAG, plusieurs contraintes doivent être prises en compte afin d'assurer la faisabilité et la viabilité du projet.

Le projet étant réalisé dans un cadre étudiant sans budget alloué, l'utilisation exclusive de technologies gratuites et open-source est impérative. Ainsi, les solutions cloud, les API payantes et les infrastructures coûteuses sont exclues, et les modèles d'IA utilisés seront issus de l'open-source afin d'éviter toute dépense superflue.

Le développement devra être réalisé dans un délai imparti de 12 semaines, avec une livraison prévue la semaine du 28 avril 2025. Ce planning inclut plusieurs étapes clés :

- ▶ La semaine du 24 février : Fin de la conception et remise de la documentation.
- ▶ La semaine du 7 avril : Dernière phase de développement, incluant l'analyse des ressources, la mise en place de l'assistant et du mode interactif, ainsi que les tests et optimisations.
- ▶ La semaine du 21 avril : Finalisation de l'intégration et remise du cahier de recette.
- ▶ La semaine du 28 avril : Préparation de la soutenance, incluant un rapport de projet, un wiki et des diapositives.

L'absence de budget impose l'utilisation de matériel personnel pour le développement, nécessitant :

- ▶ Des ordinateurs adaptés à l'exécution locale du projet et à l'IDE choisi.
- ▶ Une connexion Internet stable, essentielle pour accéder aux API et aux outils collaboratifs.
- ▶ Une base de données vectorielle open-source auto-hébergée, permettant le stockage et la recherche efficace des documents administratifs indexés et des données générées.

Le projet devra répondre à plusieurs exigences en matière de qualité, de sécurité et d'ergonomie :

- ▶ Structuration et qualité des données : Les documents doivent être clairs, organisés et exploitables par l'IA pour garantir des réponses pertinentes.
- ▶ Modularité du code : Le développement devra être modulaire, facilitant ainsi la maintenance, la réutilisation et l'ajout de nouvelles fonctionnalités sans impacter l'ensemble du projet.
- ▶ Sécurité et confidentialité : Certaines informations administratives pouvant être sensibles, le projet devra respecter les réglementations en vigueur, notamment le RGPD (protection des données personnelles).
- ▶ Ergonomie et accessibilité : L'interface utilisateur devra être intuitive, fluide et accessible aux différents profils d'utilisateurs (nouveaux collaborateurs, étudiants, etc.).
- ▶ Conformité de l'application web : L'application devra respecter les fonctionnalités définies, offrir une expérience utilisateur optimale, et garantir performance et fiabilité. Des tests et validations seront menés pour s'assurer de son adéquation avec les attentes.

En résumé, ces contraintes et une gestion agile avec des sprints courts et des objectifs intermédiaires assureront un produit fonctionnel, optimisé et aligné avec les ressources disponibles et les exigences techniques du projet tout en garantissant le respect des échéances.

4 Déroulement du projet

4.1 Planification

Afin de garantir le bon déroulement de ce projet, une restriction temporelle est nécessaire. La durée imposée pour celui-ci s'étend sur 12 semaines, combinant à la fois la phase de conception et celle de développement.

Pour mieux visualiser la temporalité du projet, nous avons répertorié l'ensemble des tâches à accomplir, réparties sur chaque semaine de travail.

Cette approche a pour objectif d'assurer une progression régulière, tout en minimisant les risques de ne pas finaliser le produit dans les délais impartis.

Semaine	Date	Tâche	Commentaire
N°1	27 Janvier	Définition des objectifs	
N°2	3 Février	Analyse des besoins	
N°3	10 Février	Spécification	Rendu du cahier des charges
		Vacances	
N°4	24 Février	Conception	Rendu du cahier de recette
N°5	3 Mars	Développement	Rendu de a conception détaillée
N°6	10 Mars	Développement	
	17 Mars	Semaine des CC	
N°7	24 Mars	Développement	
N°8	31 Mars	Développement	
N°9	7 Avril	Développement	
N°10	14 Avril	Intégration	
N°11	21 Avril	Recette	
N°12	28 Avril	Pré-soutenance	
	5 Mai		
	12 Mai	Examens	
	19 Mai	Examens	Rendu du rapport, wiki, diapositives sonorisées.
Du 27 Mai	au 31 Mai	Soutenances	

Fig. 1. – Tableau de planification du projet sur 12 semaines

4.1.a Phase de Conception

4.1.b Phase de Développement

- ▶ **Écriture des tests unitaires** : Après chaque implémentation, nous testerons la fonction pour nous assurer qu'elle répond bien aux exigences définies. Nous validerons son efficacité sur des cas pratiques et recueillerons les retours des utilisateurs afin d'ajuster les fonctionnalités et l'ergonomie si nécessaire.
- ▶ **Implémentation des fonctionnalités** : Nous développerons les différentes fonctions en nous assurant qu'elles passent les tests préalablement définis.
- ▶ **Vérification de la conformité** : Après chaque implémentation, nous vérifierons que la fonction répond bien aux exigences spécifiées, en confrontant son comportement aux attentes définies dans les spécifications fonctionnelles.
- ▶ **Correction et amélioration** : Enfin, nous procéderons à la correction des éventuels bugs détectés et à l'amélioration du code si nécessaire, afin d'optimiser la stabilité et les performances du logiciel.

La semaine d'intégration sera dédiée à l'assemblage de l'ensemble du code du projet afin d'obtenir une version exécutable. Cette étape nécessitera de revoir et de corriger les derniers éléments non encore pris en compte, ainsi que de vérifier une dernière fois que le projet répond bien aux exigences du client.

4.2 Ressources

Identifier les types de questions et de procédures administratives à traiter. Centraliser les documents réglementaires et ressources documentaires à intégrer dans l'assistant.

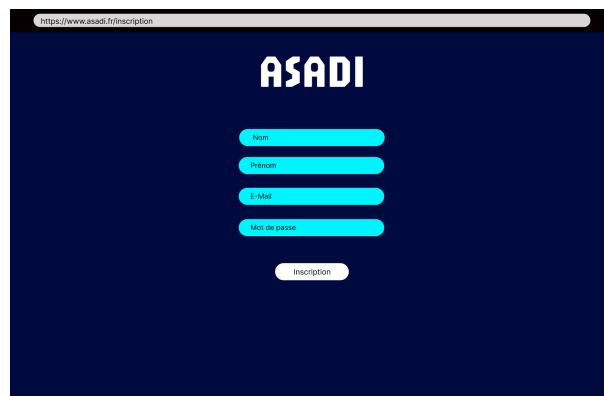
4.2.b Développement de l'assistant :

Enrichir la base de connaissances de l'assistant avec les documents identifiés. Écrire les prompts adaptés à chaque situation. Concevoir le mode d'apprentissage interactif en développant des scénarios de mise en situation et un système de création de questions dynamiques. Intégrer des fonctionnalités de suivi et de progression pour les utilisateurs.

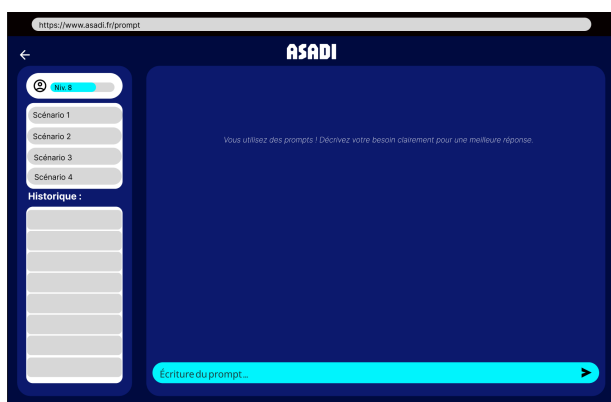
5 Annexes



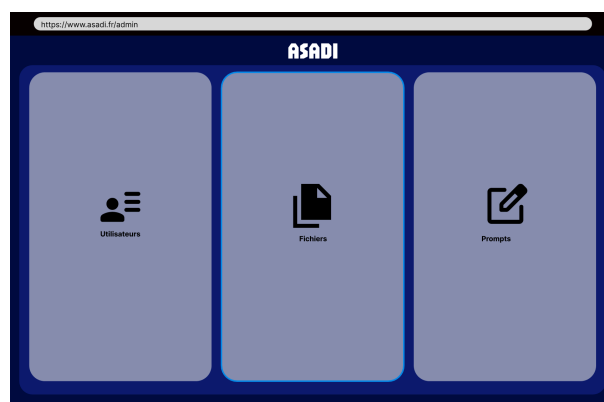
Écran de login (User) pour se connecter.



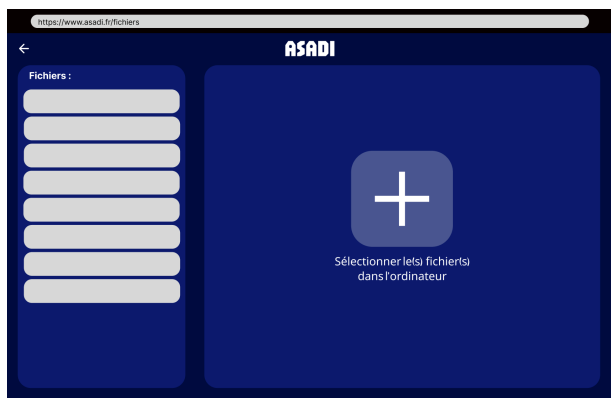
Écran d'inscription (User) pour créer un compte.



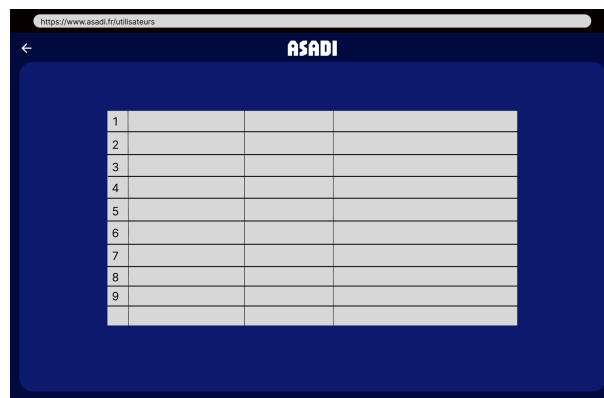
Écran de prompt (User & Admin) pour rédiger des prompts.



Écran de menu (Admin) permettant de naviger dans l'application.



Écran de chargement des document (Admin) pour déposer des fichiers administratifs.



Écran avec la liste des users (Admin) pour visualiser les utilisateurs.

6 Glossaire

- ▶ Deep learning = Le Deep Learning ou apprentissage profond est une technique de machine learning reposant sur le modèle des réseaux neurones: des dizaines voire des centaines de couches de neurones sont empilées pour apporter une plus grande complexité à l'établissement des règles.
- ▶ Utilisation du RAG : Lorsqu'une question est posée, le système récupère d'abord les documents pertinents et les utilise pour générer des réponses factuelles et fiables. Cette approche innovante est aujourd'hui utilisée dans les assistants intelligents, la synthèse de contenu et les FAQ automatisées, révolutionnant la manière dont les informations sont traitées et présentées.

7 Références

- ▶ Source de la définition du Deep learning
- ▶ Pour la rédaction de cette documentation, nous avons utilisé le modèle d'intelligence artificielle ChatGPT. Ce modèle nous a aidés sur plusieurs points :
 - Reformulation de phrases et correction de fautes d'orthographe : Nous avons utilisé ChatGPT pour nous assister dans la rédaction de certaines phrases, afin d'améliorer la clarté du texte.
 - Recherche : ChatGPT nous a permis d'explorer différentes technologies que nous devons utiliser dans ce projet. Il nous a également aidés à mieux comprendre certains concepts abstraits.

Toutefois, l'ensemble des droits de propriété intellectuelle de ce document demeure exclusivement attribué à ses auteurs.

8 Index

A

Application web	3, 5
Apprentissage interactif	5
Apprentissage ludique	3

B

Backend	3
Base de connaissance	5, 6
Base de données vectorielle	3, 9

C

Conception	10, 11
------------	--------

D

Deep learning	4
Diagramme de cas d'utilisation	11
Diagramme de classes	11
Django	3
Développement	10, 11

E

Ergonomie	11
Extraction	3

F

Faiss	4
Fonctionnalité	11
Framework	3
Frontend	3

G

GenAI	4
Génération augmentée	5
Génération de texte	4

I

Indexation	3
Intelligence artificielle	3, 4, 5, 6

L

Langage naturel	4
-----------------	---

M

Machine learning	3
Maquette	11
Milvus	4
Modèle de langage	3
Méthode agile	11

O

Open-source	9
-------------	---

P

Pipeline	3
Python	3

Q

Quiz	5, 6, 7
------	---------

R

RAG	3, 4, 5, 9
Requêtes	3
Ressources documentaires	3, 5

S

Scénario	3, 5, 7
Scénarios administratifs	6
Scénarios immersifs	5
Sources	5
Suivi personnalisé	5

T

Tesseract	3
Tests unitaires	11
Traitement des données	3

V

Vecteurs	4
----------	---