

Remerciements

Je tiens à exprimer ma profonde gratitude envers Madame Raja Touahni, coordinatrice du Master, pour son implication et sa disponibilité. Son soutien a été précieux pour la bonne organisation de ce projet et pour la réussite de notre promotion. Je remercie

également M. Anass Nouri, pour sa supervision patiente et ses précieux conseils, qui ont orienté ma recherche et m'ont permis de surmonter les défis rencontrés. Sa rigueur académique et son enthousiasme communicatif ont grandement contribué à enrichir mes connaissances et à améliorer la qualité de ce projet.

Je souhaite également exprimer ma reconnaissance à M. Jawad dabounou pour ses suggestions pertinentes et ses encouragements constants. Son soutien moral et professionnel a été d'une grande aide tout au long de ce parcours.

Enfin, je remercie l'ensemble de l'équipe pédagogique pour avoir créé un environnement d'apprentissage stimulant et collaboratif. Grâce à eux, j'ai pu développer des compétences essentielles qui me serviront tout au long de ma carrière.

Merci à tous pour votre dévouement et votre engagement envers l'éducation et la réussite de vos étudiants.

Table des matières

Résumé	6
1 Introduction	8
1.1 Contexte et motivation	8
1.2 Problématique	9
1.3 Importance du projet	9
1.3.1 Choix du Nom : "CityConnect"	11
2 Revue de la Littérature	13
2.1 Travaux connexes	13
2.1.1 Systèmes de Gestion Urbaine	13
2.1.2 Intégration de Modèles de Langage de Grande Échelle	13
2.1.3 Techniques de Traitement du Langage Naturel (NLP)	14
2.1.4 Plateformes de Participation Citoyenne	14
2.1.5 Utilisation de l'Intelligence Artificielle dans les Services Publics	14
2.2 Solutions existantes et leurs limitations	14
2.2.1 Systèmes de Gestion Urbaine	15
2.2.2 Plateformes de Participation Citoyenne	15
2.2.3 Systèmes de Réponse Automatique	16
2.2.4 Intégration de l'IA dans les Services Publics	16
2.3 Cadre théorique	16
2.3.1 Traitement du Langage Naturel (NLP)	17
2.3.2 Modèles de Langage de Grande Échelle (LLMs)	17
2.3.3 Apprentissage Automatique et Deep Learning	17
2.3.4 Systèmes de Recommandation	17
2.3.5 Recherche d'Information et Récupération Augmentée (RAG) . .	18
2.3.6 Ingénierie de Prompt	18
2.3.7 Intégration Web et Web Scraping	18
3 Méthodologie	19
3.1 Collecte des données	19
3.2 Description du jeu de données	20
3.3 Outils et technologies utilisés	21

4	Architecture du Système	23
4.1	Diagramme de l'architecture globale	23
4.2	Description détaillée de chaque composant	25
4.2.1	Frontend	25
4.2.2	Backend	36
4.2.3	Base de Données	43
4.3	Fonctionnalités supplémentaires	45
4.3.1	Catégories (droit, transport, logement, environnement)	45
4.3.2	Conception de l'interface utilisateur	45
4.3.3	Système de vote (up/down)	46
4.3.4	composant LLM	46
5	Intégration avec les Modèles de Langage (LLMs)	48
5.1	Système de Réponse Automatique	49
5.1.1	Préparation du jeu de données	49
5.1.2	Comparaison entre RAG et Fine Tuning	50
5.1.3	Techniques d'ingénierie de prompt	51
5.1.4	Intégration avec les LLMs implementation	52
5.1.5	Intégration avec le web	54
6	Résultats	58
6.1	Évaluation des performances	58
6.1.1	Temps de réponse	58
6.1.2	Précision des réponses	58
6.2	Retour des utilisateurs et tests de convivialité	59
6.2.1	Satisfaction des utilisateurs	59
6.2.2	Suggestions d'amélioration	59
6.3	Précision des réponses automatiques	59
6.3.1	Évaluation quantitative	59
6.3.2	Amélioration continue	59
6.3.3	Analyse des erreurs	60
7	Discussion	61
7.1	Défis rencontrés	61
7.1.1	Intégration des Modèles de Langage (LLMs)	61
7.1.2	Gestion des Données	61
7.1.3	Authentification et Sécurité	61
7.1.4	Scalabilité et Performances	62
7.1.5	Expérience Utilisateur	62
7.2	Limitations du système	62
7.2.1	Dépendance à l'API d'OpenAI	62
7.2.2	Latence des Réponses	62
7.2.3	Qualité des Données	63

7.2.4	Scalabilité Limité par les Ressources	63
7.2.5	Sécurité des Données	63
7.2.6	Accessibilité et Compatibilité	63
7.2.7	Mise à jour manuelle du jeu de données	63
8	Conclusion	65
8.1	Résumé du projet	65
8.2	Principales conclusions	66
8.2.1	Amélioration de l'Interaction et de la Communication	66
8.2.2	Efficacité de l'Intégration des LLMs	66
8.2.3	Robustesse et Scalabilité du Système	66
8.2.4	Défis et Limitations Persistants	67
8.2.5	Impacts sur la Participation Citoyenne	67
8.3	Travaux futurs et améliorations	67
8.3.1	Amélioration de la Latence et de la Performance	67
8.3.2	Diversification des Sources de Données	68
8.3.3	Amélioration de la Sécurité	68
8.3.4	Accessibilité et Compatibilité Améliorées	68
8.3.5	Expérience Utilisateur Améliorée	68
8.3.6	Expansions Fonctionnelles	68
8.3.7	Collaboration Inter-agences	69

Résumé

Aperçu du projet

Ce projet de fin d'études de master consiste en la création d'une plateforme web innovante pour l'Agence Urbaine de notre ville kénitra, conçue pour faciliter l'interaction et la communication entre les citoyens.

Cette plateforme permet aux utilisateurs de discuter, de déposer des plaintes, de poser des questions et d'obtenir des réponses de la part d'autres citoyens ainsi que de la plateforme elle-même, grâce à un système de réponse automatique basé sur des Modèles de Langage de Grande Échelle (LLMs).

Objectifs

1. **Améliorer la communication citoyenne** : Offrir un espace interactif où les citoyens peuvent échanger des idées,
2. **Automatisation des réponses** : Mettre en place un système de réponse automatique capable de traiter et de répondre aux questions des citoyens en se basant sur une vaste base de données de documents relatifs aux lois, transports, logements et environnement.
3. **Accessibilité et convivialité** : Concevoir une interface utilisateur intuitive et accessible pour encourager l'utilisation de la plateforme par une large audience.
4. **Support multi-catégories** : Permettre aux utilisateurs de poser des questions et de recevoir des réponses précises et pertinentes dans plusieurs domaines spécifiques tels que le droit, les transports, le logement et l'environnement.

Fonctionnalités clés

Afin d'atteindre les objectifs fixés et de répondre efficacement aux besoins des citoyens, ce projet intègre une série de fonctionnalités innovantes et intuitives. Ces fonctionnalités sont conçues pour offrir une expérience utilisateur optimale, favoriser une interaction fluide entre les citoyens, et garantir un accès rapide et précis aux informations et services urbains. Voici les principales fonctionnalités de la plateforme :

- **Soumission de plaintes et questions** : Les utilisateurs peuvent facilement soumettre leurs plaintes ou poser des questions via un formulaire dédié. Ces soumissions sont ensuite visibles par d'autres citoyens ou traitées par le système de réponse automatique.
- **Réponses automatisées** : En intégrant des techniques avancées d'ingénierie de prompt et des LLMs, la plateforme peut fournir des réponses instantanées et précises aux questions posées par les utilisateurs.
- **Catégorisation des questions** : Les questions peuvent être classées dans diverses catégories (droit, transports, logement, environnement), permettant ainsi un traitement et une recherche plus efficaces.
- **Système de notifications** : Les utilisateurs sont informés des réponses à leurs questions ou de l'état de leurs plaintes via un système de notifications intégré.
- **Interface utilisateur intuitive** : La conception de l'interface met l'accent sur la simplicité et l'efficacité, garantissant une expérience utilisateur agréable et accessible à tous.

Chapitre 1

Introduction

1.1 Contexte et motivation

L'intelligence artificielle (IA) est en train de révolutionner notre monde à une vitesse fulgurante, se positionnant comme un pilier essentiel du futur. Sa capacité à analyser d'immenses quantités de données, à apprendre de manière autonome et à effectuer des tâches complexes transforme déjà divers secteurs, de la santé à l'industrie, en passant par les services financiers et l'éducation. L'IA promet d'améliorer l'efficacité, d'innover dans les solutions aux problèmes mondiaux et d'ouvrir de nouvelles perspectives dans la recherche scientifique. En envisageant son développement futur, il devient clair que l'IA n'est pas seulement une avancée technologique, mais aussi un vecteur crucial pour le progrès économique et social. Son rôle dans notre avenir est donc d'une importance capitale, tant pour les opportunités qu'elle crée que pour les défis éthiques et sociétaux qu'elle pose.

L'un des moteurs principaux derrière ces avancées spectaculaires de l'intelligence artificielle est l'émergence des modèles de langage de grande taille (LLM). Ces modèles, comme GPT-4, ont la capacité de comprendre et de générer du langage humain avec un niveau de sophistication sans précédent. En entraînant ces modèles sur des vastes corpus de textes, les chercheurs ont réussi à doter les machines d'une compréhension contextuelle et sémantique du langage, ce qui a considérablement élargi les applications de l'IA. Les LLM sont à la base de nombreuses innovations, telles que les assistants virtuels intelligents, la traduction automatique, la génération de contenu créatif, et bien plus encore. Grâce à leur flexibilité et à leur puissance, ils ont ouvert la voie à des solutions plus intelligentes et plus adaptatives, permettant à l'IA de s'intégrer de manière plus naturelle et efficace dans notre quotidien et de répondre de manière plus précise aux besoins et aux défis contemporains.

1.2 Problématique

Dans le contexte urbain actuel, les citoyens rencontrent de nombreux défis en matière de communication et d'interaction avec les agences gouvernementales. Les processus traditionnels de soumission de plaintes, de demandes d'informations ou de suggestions sont souvent laborieux, peu transparents et manquent de réactivité. De plus, les citoyens peuvent se sentir isolés, ne sachant pas comment ou à qui s'adresser pour résoudre leurs préoccupations.

Problèmes Identifiés

1. **Manque de réactivité et de transparence** : Les systèmes actuels de gestion des plaintes et des questions sont souvent lents, avec des réponses tardives ou inexistantes, ce qui engendre de la frustration chez les citoyens. La transparence est également un problème, car les utilisateurs ne savent pas toujours où en est le traitement de leurs demandes.
2. **Communication inefficace** : Les canaux de communication entre les citoyens et les agences urbaines sont fragmentés et peu efficaces. Il y a un manque d'interaction directe et d'échanges constructifs, ce qui limite la participation citoyenne et la résolution proactive des problèmes.
3. **Accès limité aux informations** : Les citoyens ont souvent du mal à trouver des informations précises et à jour sur les lois, les règlements, les transports, le logement et l'environnement. Les sources d'information disponibles ne sont pas toujours accessibles ou compréhensibles pour le grand public.
4. **Participation citoyenne réduite** : En raison de la complexité et du manque d'efficacité des systèmes actuels, de nombreux citoyens sont découragés de participer activement à la gestion de leur environnement urbain. Cela entraîne un engagement communautaire limité et une réduction de l'impact des initiatives citoyennes.

1.3 Importance du projet

La création d'une plateforme de communication interactive entre les citoyens et l'Agence Urbaine revêt une importance capitale dans le contexte actuel de gestion urbaine. Cette initiative répond à plusieurs enjeux majeurs et offre des avantages significatifs pour la communauté urbaine dans son ensemble.

Renforcement de la Participation Citoyenne

L'un des aspects les plus cruciaux de ce projet est sa capacité à renforcer la participation citoyenne dans la gestion de l'environnement urbain. En offrant aux citoyens un moyen facile et efficace de soumettre leurs plaintes, poser des questions et contribuer

à la résolution des problèmes locaux, la plateforme encourage l'engagement actif et la responsabilisation des membres de la communauté. Une participation citoyenne accrue favorise également une gouvernance plus démocratique et transparente, où les décisions prises par les autorités locales reflètent véritablement les besoins et les aspirations de la population.

Amélioration de la Réactivité et de la Transparence

La mise en place d'un système de communication directe entre les citoyens et l'Agence Urbaine permet d'améliorer considérablement la réactivité et la transparence des services gouvernementaux. En fournissant des réponses rapides et pertinentes aux questions des citoyens, ainsi qu'un suivi transparent des plaintes et des demandes d'informations, la plateforme contribue à renforcer la confiance des citoyens envers les institutions publiques. Cette transparence accrue favorise également une meilleure reddition de comptes et une gestion plus efficace des ressources publiques, en permettant aux autorités urbaines de mieux comprendre et de répondre aux besoins réels de la population.

Facilitation de l'Accès à l'Information

En consolidant et en centralisant les informations pertinentes sur les lois, les règlements, les projets urbains et les services publics, la plateforme facilite grandement l'accès des citoyens à l'information. Cette accessibilité accrue renforce l'autonomie des citoyens et leur capacité à prendre des décisions éclairées concernant leur vie quotidienne et leur environnement urbain. De plus, en fournissant des informations actualisées et vérifiées, la plateforme contribue à réduire la propagation de fausses informations et de rumeurs, renforçant ainsi la cohésion sociale et la confiance mutuelle au sein de la communauté.

Promotion de l'Innovation et de l'Adoption des Technologies Numériques

En adoptant des technologies de pointe telles que les Modèles de Langage de Grande Échelle (LLMs) et les systèmes de réponse automatique, ce projet favorise l'innovation et l'adoption des technologies numériques dans le domaine de la gouvernance urbaine. En montrant comment les outils numériques peuvent être utilisés pour améliorer la communication et la collaboration entre les citoyens et les institutions publiques, la plateforme inspire d'autres initiatives similaires dans d'autres villes et régions, contribuant ainsi à l'émergence d'une culture numérique plus forte et plus inclusive à l'échelle nationale.

1.3.1 Choix du Nom : "CityConnect"

Le choix du nom pour une plateforme est un élément crucial qui influence la perception et l'identité de la marque. Pour notre site, nous avons choisi le nom "CityConnect" en raison de ses connotations positives et de sa pertinence par rapport aux objectifs de la plateforme. Voici les raisons détaillées de ce choix :

1. Clarté et Simplicité

"CityConnect" est un nom clair et simple qui est facile à retenir et à prononcer. Cette simplicité favorise la reconnaissance de la marque et facilite la communication autour de la plateforme, tant en ligne que hors ligne. Un nom facile à mémoriser aide également à attirer et à retenir les utilisateurs.

2. Connotation de Connexion

Le terme "Connect" dans "CityConnect" évoque l'idée de lien et de mise en relation. Cette connotation est essentielle pour notre plateforme, dont l'objectif principal est de faciliter l'interaction et la communication entre les citoyens et l'Agence Urbaine. Le nom reflète donc parfaitement la mission de la plateforme de créer des ponts entre les différents acteurs de la ville.

3. Focus Urbain

Le mot "City" ancre la plateforme dans un contexte urbain, précisant clairement que les services et fonctionnalités offerts sont dédiés aux besoins des villes et de leurs habitants. Cela permet aux utilisateurs potentiels de comprendre instantanément le domaine d'application de la plateforme.

4. Inclusivité et Engagement Communautaire

"CityConnect" suggère également une plateforme inclusive qui vise à engager tous les membres de la communauté urbaine. Cela renforce l'idée que la plateforme est un espace ouvert à tous, où chacun peut participer, partager ses idées, et contribuer à la vie urbaine.

5. Modernité et Innovation

Le nom "CityConnect" véhicule une image de modernité et d'innovation. Il suggère que la plateforme utilise des technologies avancées pour améliorer la vie urbaine, ce qui est un aspect clé de notre projet. Cette perception peut attirer des utilisateurs intéressés par les solutions numériques modernes pour les défis urbains.

6. Polyvalence et Évolutivité

Enfin, "CityConnect" est un nom polyvalent qui peut facilement évoluer avec la plateforme. Que nous ajoutions de nouvelles fonctionnalités ou que nous étendions nos services à de nouvelles villes, le nom reste pertinent et adaptable, soutenant ainsi la croissance et l'expansion future de la plateforme. ce choix est basé sur sa clarté, sa simplicité et sa capacité à communiquer efficacement la mission et les valeurs de la plateforme. Il évoque des idées de connexion, d'inclusivité et d'innovation, tout en étant ancré dans un contexte urbain, ce qui le rend parfaitement adapté à notre projet.

En résumé, ce projet revêt une importance stratégique dans la transformation de nos villes en des espaces plus ouverts, inclusifs et responsables, où la voix de chaque

citoyen compte et contribue à façonner l'avenir commun.

Chapitre 2

Revue de la Littérature

2.1 Travaux connexes

Dans cette section, nous examinons les travaux connexes qui ont inspiré et informé le développement de notre projet de plateforme web pour l'Agence Urbaine. Ces travaux couvrent divers domaines, notamment les systèmes de gestion urbaine, l'intégration de modèles de langage de grande échelle, et les techniques de traitement du langage naturel (NLP).

2.1.1 Systèmes de Gestion Urbaine

Plusieurs études et projets se sont concentrés sur la création de systèmes de gestion urbaine visant à améliorer la communication entre les citoyens et les administrations urbaines. Par exemple, le projet *CitySDK* a développé une plateforme ouverte et interopérable pour les services urbains, facilitant l'intégration des données et la participation citoyenne (Riva et al., 2015). De même, la plateforme *FixMyStreet* permet aux citoyens de signaler les problèmes de voirie et de recevoir des réponses des autorités locales, ce qui améliore la réactivité et la transparence des administrations (Brown et al., 2013).

2.1.2 Intégration de Modèles de Langage de Grande Échelle

L'intégration des modèles de langage de grande échelle (LLMs) dans les systèmes interactifs est un domaine en pleine expansion. Les travaux de Devlin et al. (2018) avec BERT (Bidirectional Encoder Representations from Transformers) ont montré comment les LLMs peuvent être utilisés pour des tâches de compréhension du langage naturel avec des performances sans précédent. De plus, les recherches de Radford et al. (2019) sur les modèles GPT (Generative Pre-trained Transformer) ont démontré la capacité des LLMs à générer des textes cohérents et contextuellement pertinents, ouvrant la voie à leur utilisation dans les systèmes de réponse automatique.

2.1.3 Techniques de Traitement du Langage Naturel (NLP)

Les techniques de traitement du langage naturel (NLP) jouent un rôle crucial dans le développement de systèmes interactifs intelligents. Le projet *spaCy* (Honnibal et Montani, 2017) est un exemple de bibliothèque NLP open-source largement utilisée pour ses performances et sa flexibilité. De plus, l'utilisation de l'architecture Transformer, introduite par Vaswani et al. (2017), a révolutionné le traitement du langage naturel en permettant des parallélisations efficaces et une meilleure capture des relations contextuelles dans les textes.

2.1.4 Plateformes de Participation Citoyenne

Les plateformes de participation citoyenne telles que *Decidim* et *Participedia* ont également servi de référence pour notre projet. *Decidim*, par exemple, est une plateforme open-source pour la participation démocratique, utilisée par plusieurs municipalités à travers le monde pour faciliter les processus décisionnels participatifs (Blanco et al., 2019). *Participedia* est une autre ressource précieuse qui documente diverses méthodes de participation citoyenne à travers des études de cas détaillées (Pateman et al., 2018).

2.1.5 Utilisation de l'Intelligence Artificielle dans les Services Publics

Enfin, l'utilisation de l'intelligence artificielle (IA) dans les services publics a été explorée par plusieurs chercheurs. Par exemple, le projet *Smart City* de Barcelone intègre des solutions basées sur l'IA pour améliorer l'efficacité des services urbains et la qualité de vie des citoyens (Batty et al., 2012). Ces initiatives montrent le potentiel de l'IA pour transformer les interactions entre les citoyens et les administrations publiques.

Ces travaux connexes fournissent un cadre théorique et pratique solide qui a guidé le développement de notre plateforme. En s'inspirant de ces études et projets, nous avons pu concevoir un système innovant et efficace pour améliorer la communication et l'interaction entre les citoyens et l'Agence Urbaine de notre ville.

2.2 Solutions existantes et leurs limitations

Dans cette section, nous examinons les solutions existantes dans le domaine de la gestion urbaine, de la participation citoyenne et des systèmes de réponse automatique, ainsi que leurs limitations. Ces solutions ont inspiré notre projet et ont également mis en lumière les défis que nous devons surmonter.

2.2.1 Systèmes de Gestion Urbaine

FixMyStreet

FixMyStreet est une plateforme permettant aux citoyens de signaler des problèmes d'infrastructure, comme des nids-de-poule ou des lampadaires défectueux. Bien que ce système améliore la réactivité des autorités locales, il présente des limitations :

- **Réactivité limitée** : La dépendance aux ressources locales pour la résolution des problèmes peut entraîner des délais.
- **Portée limitée** : Le système se concentre principalement sur les problèmes de voirie, négligeant d'autres aspects de la gestion urbaine.

CitySDK

CitySDK offre une plateforme interopérable pour les services urbains, facilitant l'intégration des données urbaines. Toutefois, ses limitations incluent :

- **Complexité d'implémentation** : La mise en place de cette solution nécessite des compétences techniques avancées.
- **Interopérabilité restreinte** : L'intégration avec des systèmes existants peut être difficile en raison de formats de données disparates.

2.2.2 Plateformes de Participation Citoyenne

Decidim

Decidim est une plateforme open-source pour la participation démocratique, largement adoptée par diverses municipalités. Ses limitations incluent :

- **Barrières techniques** : La configuration et la personnalisation de la plateforme nécessitent des compétences techniques.
- **Engagement limité** : La participation citoyenne peut être faible sans des campagnes de sensibilisation et d'engagement actif.

Participedia

Participedia documente des méthodes de participation citoyenne à travers des études de cas. Bien que riche en informations, elle présente des limitations :

- **Pas d'interaction directe** : La plateforme se concentre sur la documentation plutôt que sur la facilitation de la participation active.
- **Portée limitée** : Principalement axée sur la recherche académique, elle n'est pas conçue pour une utilisation quotidienne par les citoyens.

2.2.3 Systèmes de Réponse Automatique

Chatbots traditionnels

Les chatbots traditionnels sont couramment utilisés pour fournir des réponses automatiques aux questions des utilisateurs. Cependant, leurs limitations sont significatives :

- **Compréhension limitée** : Ils utilisent souvent des règles préprogrammées et des arbres de décision, ce qui limite leur capacité à comprendre des questions complexes.
- **Flexibilité restreinte** : Ils ne peuvent pas s'adapter facilement aux nouvelles informations ou contextes.

Modèles de langage de grande échelle (LLMs)

Les LLMs, tels que BERT et GPT, offrent des capacités avancées de compréhension et de génération de texte. Cependant, ils présentent également des défis :

- **Dépendance aux données** : Leur performance dépend fortement de la qualité et de la quantité des données d'entraînement.
- **Ressources computationnelles** : Ils nécessitent des ressources computationnelles importantes, rendant leur déploiement coûteux.
- **Mise à jour des connaissances** : Une fois entraînés, ils ne peuvent pas intégrer facilement de nouvelles informations sans être réentraînés.

2.2.4 Intégration de l'IA dans les Services Publics

Smart City de Barcelone

Le projet *Smart City* de Barcelone intègre des solutions basées sur l'IA pour améliorer l'efficacité des services urbains. Cependant, il présente des limitations :

- **Coûts élevés** : Les investissements initiaux et les coûts de maintenance peuvent être prohibitifs.
- **Complexité de mise en œuvre** : La coordination entre les différentes parties prenantes et la gestion des données peut être complexe.

Ces solutions existantes, bien qu'innovantes et bénéfiques à divers égards, montrent qu'il existe des défis communs tels que la complexité technique, la réactivité limitée, et la difficulté à maintenir des systèmes à jour. En développant notre propre plateforme, nous avons cherché à surmonter ces limitations en proposant une solution intégrée, flexible, et évolutive, capable de répondre aux besoins spécifiques de l'Agence Urbaine et de ses citoyens.

2.3 Cadre théorique

Le développement de notre plateforme s'appuie sur un cadre théorique solide qui intègre des concepts et des méthodes provenant de plusieurs domaines clés. Ce cadre

théorique guide notre approche et nos décisions de conception, en assurant que notre solution est à la fois scientifiquement rigoureuse et technologiquement avancée. Voici les principaux éléments de ce cadre théorique :

2.3.1 Traitement du Langage Naturel (NLP)

Le traitement du langage naturel (NLP) est un domaine de l'intelligence artificielle qui se concentre sur l'interaction entre les ordinateurs et les humains par le langage. Les techniques de NLP permettent aux machines de comprendre, d'interpréter et de répondre aux données textuelles de manière significative. Les modèles de langage de grande échelle (LLMs) tels que BERT (Devlin et al., 2018) et GPT (Radford et al., 2019) sont des exemples avancés de l'application du NLP, permettant des performances de pointe dans des tâches de compréhension et de génération de texte.

2.3.2 Modèles de Langage de Grande Échelle (LLMs)

Les modèles de langage de grande échelle (LLMs) sont des réseaux neuronaux pré-entraînés sur de vastes corpus de données textuelles. Ces modèles peuvent générer du texte, répondre à des questions et accomplir diverses tâches linguistiques en utilisant les connaissances acquises lors de leur pré-entraînement. Les LLMs sont au cœur de notre système de réponse automatique, permettant de fournir des réponses pertinentes et contextuellement appropriées aux questions des utilisateurs.

2.3.3 Apprentissage Automatique et Deep Learning

L'apprentissage automatique (machine learning) et le deep learning sont des sous-domaines de l'intelligence artificielle qui se concentrent sur la création d'algorithmes capables d'apprendre à partir de données et de faire des prédictions. Le deep learning, en particulier, utilise des réseaux de neurones profonds pour modéliser des relations complexes dans les données. Les techniques de deep learning sont utilisées dans notre plateforme pour transformer les questions en embeddings vectoriels et pour entraîner les modèles de langage de grande échelle.

2.3.4 Systèmes de Recommandation

Les systèmes de recommandation sont utilisés pour suggérer des éléments aux utilisateurs en fonction de divers critères. Dans notre plateforme, les systèmes de recommandation jouent un rôle clé dans la sélection des réponses les plus pertinentes en fonction des embeddings des questions et des textes disponibles dans notre base de données vectorielle. La recherche de similarité (similarity search) basée sur les embeddings vectoriels est une technique couramment utilisée dans ce contexte.

2.3.5 Recherche d'Information et Récupération Augmentée (RAG)

La recherche d'information (Information Retrieval) est le processus de récupération de données pertinentes à partir de grandes bases de données textuelles. La technique de Récupération Augmentée par Génération (Retrieval-Augmented Generation, RAG) combine la génération de texte par les LLMs avec la recherche d'informations pour fournir des réponses enrichies et contextuellement pertinentes. RAG est particulièrement utile lorsque les informations pertinentes ne sont pas directement disponibles dans le modèle de langage et doivent être récupérées à partir de sources externes.

2.3.6 Ingénierie de Prompt

L'ingénierie de prompt est une technique utilisée pour formuler les entrées données aux modèles de langage afin d'obtenir les meilleures réponses possibles. En structurant les prompts de manière optimale, nous pouvons guider le modèle de langage pour générer des réponses plus précises et pertinentes. L'ingénierie de prompt inclut des techniques telles que le Prompt Chaining et la gestion de contextes multiples pour améliorer la qualité des réponses.

2.3.7 Intégration Web et Web Scraping

L'intégration web et le web scraping sont des techniques utilisées pour extraire des données de sites web. Ces techniques sont essentielles pour notre plateforme lorsqu'il s'agit de récupérer des informations actualisées qui ne sont pas disponibles dans notre base de données interne. Le web scraping permet d'automatiser l'extraction de contenu de haute qualité à partir de sources fiables sur le web.

En combinant ces divers éléments théoriques, nous avons développé une plateforme robuste et efficace qui intègre les dernières avancées en matière de traitement du langage naturel, d'apprentissage automatique et de systèmes de recommandation. Ce cadre théorique nous permet de fournir une solution innovante et adaptable pour améliorer la communication et l'interaction entre les citoyens et l'Agence Urbaine.

Chapitre 3

Méthodologie

La méthodologie adoptée pour ce projet de création d'une plateforme web pour l'Agence Urbaine repose sur une approche structurée et systématique, visant à garantir la réalisation des objectifs fixés tout en assurant la qualité et l'efficacité de la solution développée. Cette section décrit en détail les différentes étapes du processus, les méthodes de collecte et d'analyse des données, ainsi que les outils et technologies utilisés.

Le choix de la méthodologie s'est basé sur la nature du projet, les exigences techniques et fonctionnelles, et les meilleures pratiques en matière de développement web et d'intégration de modèles de langage de grande échelle (LLMs).

3.1 Collecte des données

La majorité des données utilisées dans ce projet ont été fournies directement par l'Agence Urbaine. Ces informations, recueillies au fil des années, ont offert une base solide et pertinente pour le développement de la plateforme. Elles incluent divers aspects liés à l'urbanisme et aux interactions citoyennes. L'accès à ces données internes a permis de garantir que les fonctionnalités de la plateforme soient adaptées aux besoins spécifiques de la communauté et alignées avec les objectifs de l'Agence Urbaine.

En complément des données internes, des informations supplémentaires ont été extraites de sites officiels et de sources gouvernementales. Ces documents comprennent des réglementations, des directives nationales et des statistiques publiques disponibles en ligne. L'intégration de ces informations a permis d'enrichir notre jeu de données et de veiller à ce que la plateforme respecte les normes en vigueur. Cette combinaison de données internes et externes a été cruciale pour créer une plateforme robuste et pertinente, capable de répondre efficacement aux attentes des utilisateurs et des autorités.

3.2 Description du jeu de données

Toutes les données utilisées dans ce projet sont sous forme de documents ces documents ont été minutieusement traités et classés selon différentes catégories pour assurer une organisation efficace et une analyse précise. Ces catégories comprennent l'aménagement urbain, le développement économique, les aspects environnementaux, le logement, la logistique et le transport, les questions juridiques, ainsi que la numérisation des services. Cette classification permet une exploration ciblée des informations pertinentes pour chaque domaine, facilitant ainsi la compréhension des besoins et des préoccupations spécifiques des citoyens dans différents aspects de la vie urbaine.

Les données initiales ont été recueillies sous forme de documents au format PDF et DOC, représentant une variété de sources telles que des rapports officiels, des lettres, des formulaires, et d'autres documents pertinents émis par l'Agence Urbaine et d'autres entités associées. Pour faciliter le traitement et l'analyse ultérieure, ces documents ont été convertis en fichiers texte brut (format TXT) à l'aide d'outils de conversion automatisés. Cette transformation a permis de préserver l'intégrité du contenu tout en facilitant l'extraction et la manipulation des données textuelles.

Après la conversion des documents en format texte brut, une étape cruciale de traduction automatique a été entreprise pour les textes non francophones. Cette démarche visait à homogénéiser le corpus de données en traduisant automatiquement les contenus rédigés dans d'autres langues vers le français, la langue principale de communication dans ce contexte. L'utilisation de techniques de traduction automatique a permis d'assurer l'accessibilité et la compréhension des informations pour l'ensemble des utilisateurs de la plateforme, tout en facilitant une analyse unifiée des données dans le cadre de ce projet axé sur la communication et l'interaction citoyenne.

La taille totale des documents utilisés dans ce projet s'élève à environ 70 mégaoctets (Mo).

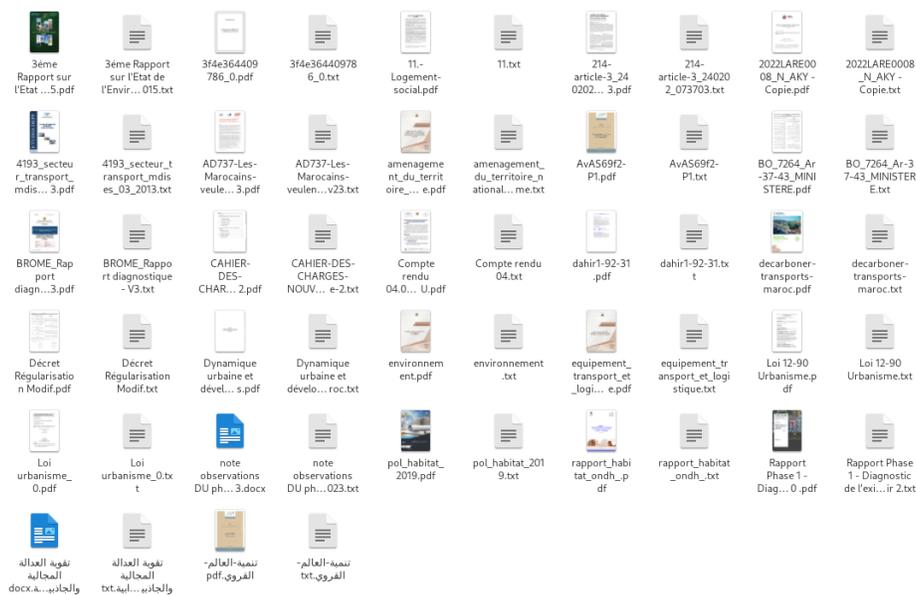


FIGURE 3.1 – Exemple des données utilisées

3.3 Outils et technologies utilisés

Dans le cadre de ce projet, le langage de programmation principal utilisé est Python, choisi pour sa flexibilité, sa richesse en bibliothèques spécialisées, et sa capacité à gérer efficacement diverses tâches de traitement des données. Python s’est révélé particulièrement adapté pour automatiser les processus de conversion et de traduction des documents, garantissant ainsi une manipulation fluide et précise des données textuelles issues de diverses sources.

Pour le traitement des documents PDF, nous avons utilisé la bibliothèque PyPDF. PyPDF nous a permis d’extraire le texte des fichiers PDF de manière fiable, facilitant ainsi la conversion de ces documents en fichiers texte brut (format TXT). Cette étape a été cruciale pour uniformiser le format des données et préparer le corpus pour une analyse ultérieure. En parallèle, pour les documents au format DOC, nous avons employé la bibliothèque python-docx. Cet outil s’est avéré indispensable pour extraire et manipuler le contenu textuel des fichiers DOC, assurant une conversion précise et efficace en fichiers texte brut.

Enfin, pour la traduction des textes non francophones, nous avons utilisé la bibliothèque googletrans. Ce package nous a permis de traduire automatiquement les contenus textuels vers le français, garantissant ainsi une homogénéité linguistique dans l’ensemble des données. L’intégration de googletrans a facilité l’accessibilité et la compréhension

des informations pour tous les utilisateurs de la plateforme, tout en soutenant une analyse unifiée et cohérente des données dans le cadre de ce projet.

Chapitre 4

Architecture du Système

L'architecture du système est une composante cruciale de ce projet. Elle définit la structure globale, les interactions entre les différents composants et les technologies utilisées pour construire une solution robuste, évolutive et efficace. Ce chapitre présente une vue d'ensemble de l'architecture du système, détaillant les principaux composants, leurs rôles, ainsi que les technologies et les outils qui ont été intégrés pour répondre aux besoins spécifiques de la plateforme.

Dans les sections qui suivent, nous explorerons le diagramme de l'architecture globale du système, offrant une visualisation claire des interactions entre le frontend, le backend, la base de données, et l'intégration avec les Modèles de Langage de Grande Échelle (LLMs). Chaque composant sera ensuite décrit en détail, mettant en lumière les choix technologiques et les considérations de conception qui ont guidé leur développement. Cette approche permettra de comprendre comment chaque partie contribue à l'ensemble du système, assurant une communication fluide et une interaction efficace entre les citoyens et l'Agence Urbaine.

4.1 Diagramme de l'architecture globale

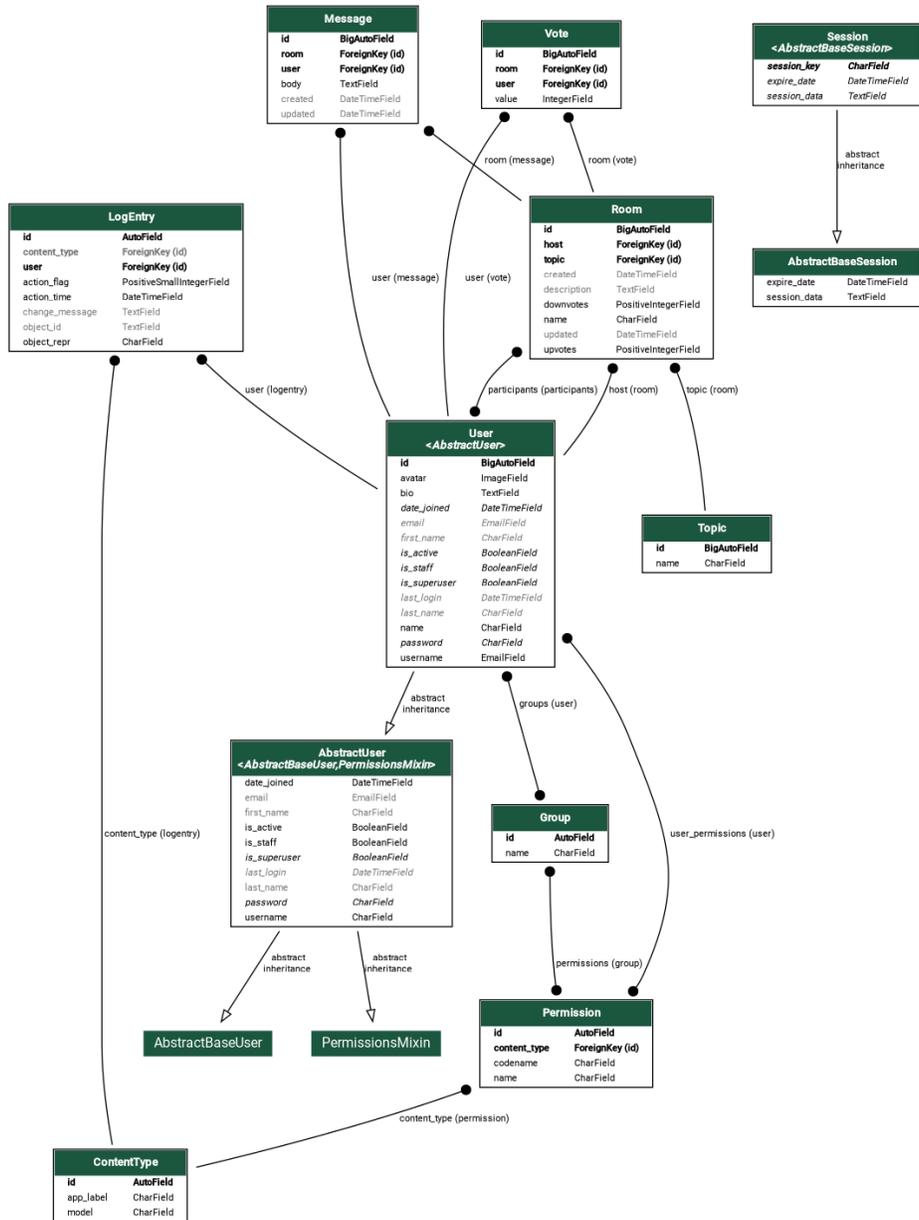


FIGURE 4.1 – Diagramme de l'architecture globale

4.2 Description détaillée de chaque composant

La section "Description détaillée de chaque composant" se consacre à une analyse approfondie des éléments clés qui composent l'architecture du système de notre plateforme web. Chaque composant joue un rôle crucial dans la fonctionnalité globale du système, et il est essentiel de comprendre leurs caractéristiques individuelles et leurs interactions pour apprécier pleinement la complexité et l'efficacité de la solution proposée.

Dans cette section, nous détaillerons le frontend, le backend, la base de données, et l'intégration avec les Modèles de Langage de Grande Échelle (LLMs). Nous expliquerons les technologies choisies, les fonctionnalités spécifiques de chaque composant, ainsi que les choix de conception qui ont guidé leur développement. Cette analyse permet de mettre en évidence la manière dont chaque élément contribue à la robustesse, à la réactivité et à l'efficacité de la plateforme, tout en facilitant une interaction transparente et une communication efficace entre les citoyens et l'Agence Urbaine.

4.2.1 Frontend

Le frontend de notre plateforme web innovante a été développé en utilisant HTML avec le moteur de templates Django, CSS et JavaScript. Cette combinaison de technologies a permis de créer une interface utilisateur intuitive, réactive et esthétiquement plaisante. La section suivante décrit en détail les composants clés du frontend et leur fonction respective.

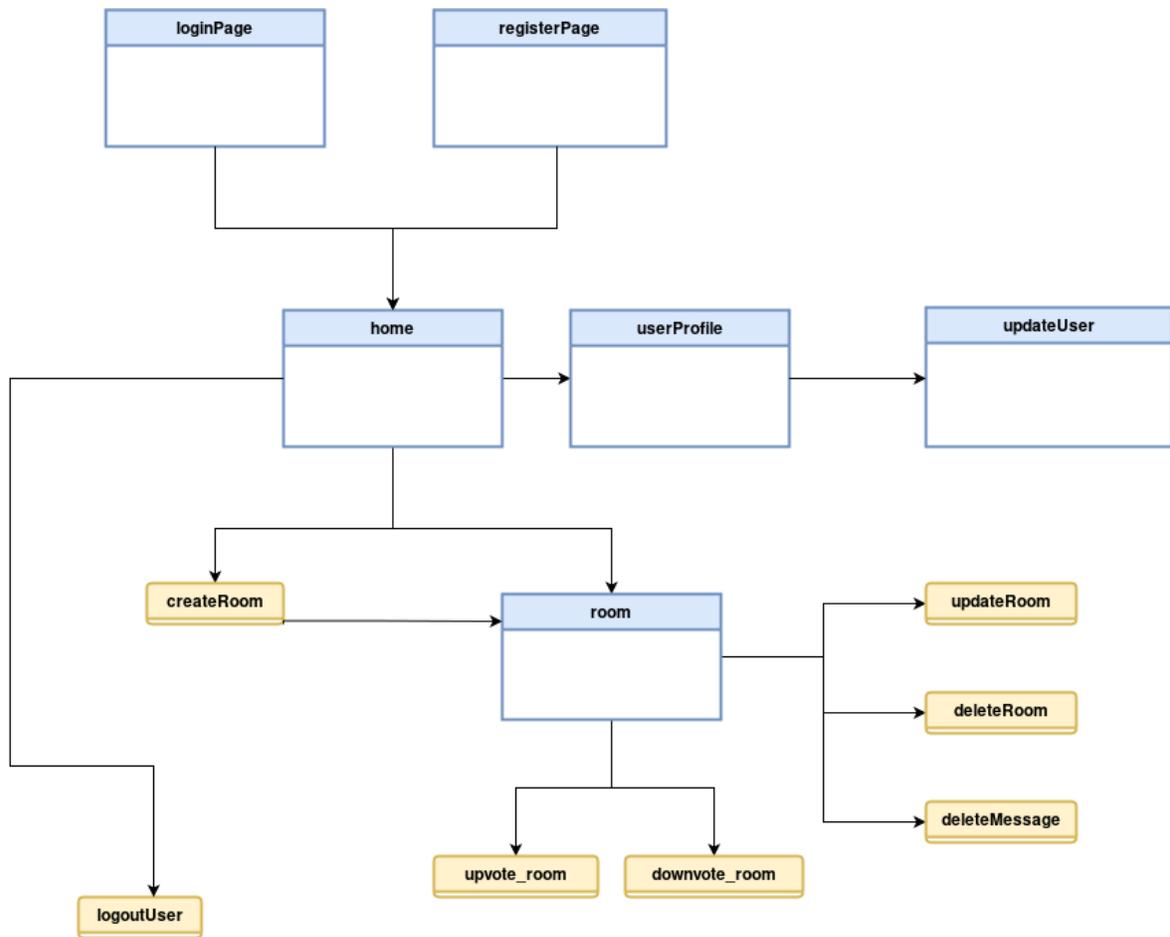


FIGURE 4.2 – Diagramme de l’architecture des applications Web
 les cases bleues indiquent les pages et les oranges indiquent les fonctionnalités

Choix de la Palette de Couleurs : Light Green



FIGURE 4.3 – palette de couleurs

Le choix de la palette de couleurs pour une plateforme web est crucial, car il influence non seulement l'esthétique du site, mais aussi l'expérience utilisateur globale. Pour notre plateforme, nous avons opté pour une palette de couleurs "light green". Voici les raisons de ce choix :

1. Apaisement et Sérénité

Le vert clair est une couleur apaisante qui évoque des sentiments de sérénité et de tranquillité. En utilisant cette couleur, nous visons à créer une atmosphère accueillante et relaxante pour les utilisateurs, les incitant à passer plus de temps sur la plateforme et à interagir avec le contenu sans stress.

2. Association à la Nature et à l'Environnement

Le vert est souvent associé à la nature, à la croissance et à la durabilité. En choisissant une palette de vert clair, nous renforçons l'association de notre plateforme avec des valeurs écologiques et environnementales. Cela est particulièrement pertinent pour une plateforme urbaine, car elle souligne notre engagement envers des pratiques urbaines durables et respectueuses de l'environnement.

3. Lisibilité et Confort Visuel

Les nuances de vert clair offrent une excellente lisibilité et confort visuel. Contrairement aux couleurs plus sombres ou très vives, le vert clair réduit la fatigue oculaire, ce qui est essentiel pour les utilisateurs qui passent de longues périodes à lire ou à interagir avec le site. Cette couleur améliore également la visibilité des textes et des éléments graphiques.

4. Uniformité et Cohérence

L'utilisation d'une palette de couleurs cohérente à travers toutes les pages et sections du site aide à maintenir une identité visuelle uniforme. Le vert clair, avec ses variations de nuances, permet de créer un design harmonieux et équilibré. Cela renforce la reconnaissance de la marque et assure une expérience utilisateur cohérente et professionnelle.

5. Accentuation et Hiérarchisation

Les différentes nuances de vert clair peuvent être utilisées pour accentuer et hiérarchiser les informations sur le site. Par exemple, des teintes plus foncées peuvent être utilisées pour les titres et les en-têtes, tandis que des nuances plus claires peuvent servir de fond pour les sections de contenu. Cette hiérarchisation visuelle guide les utilisateurs à travers le contenu de manière intuitive.

6. Accessibilité et Inclusivité

Le vert clair est également une couleur accessible qui est perçue de manière uniforme par la majorité des utilisateurs, y compris ceux avec des déficiences visuelles légères. En utilisant des contrastes appropriés entre le vert clair et d'autres couleurs, nous assurons que notre site est accessible à un large éventail d'utilisateurs, conformément aux normes d'accessibilité web.

En conclusion, le choix d'une palette de couleurs "light green" pour notre site est basé sur des considérations esthétiques et pratiques. Cette couleur non seulement améliore l'expérience utilisateur en offrant une ambiance apaisante et une excellente lisibilité, mais elle renforce également notre engagement envers des valeurs environnementales et la durabilité. Le résultat est une plateforme attrayante, facile à naviguer et cohérente, qui répond aux besoins de nos utilisateurs tout en soutenant notre mission et nos valeurs.

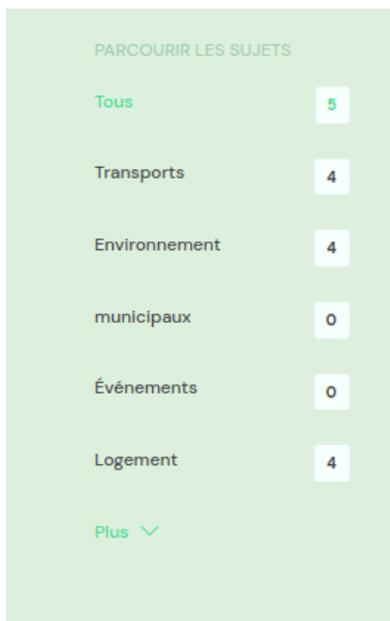
activity_component

Ce composant liste les dernières activités effectuées sur la plateforme. Il affiche les actions récentes des utilisateurs, telles que les questions posées, les réponses fournies et les votes effectués. Ce composant est essentiel pour maintenir les utilisateurs informés des interactions et activités récentes, favorisant ainsi une participation continue.



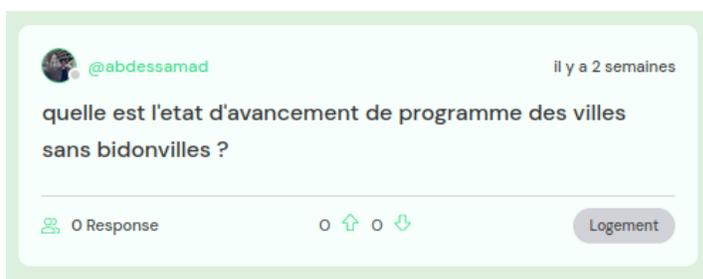
subject

Le composant subject contient une liste cliquable de sujets, avec un compteur pour chaque sujet indiquant le nombre de questions associées. Cette fonctionnalité permet aux utilisateurs de naviguer facilement parmi les différents sujets d'intérêt, facilitant ainsi l'accès aux informations pertinentes et encourageant les discussions ciblées.



feed_component

Ce composant affiche une question individuelle (feed item) avec les détails suivants : le nom du créateur, l'heure de création, les flèches de vote (haut et bas), le sujet, l'image de profil du créateur, et le nombre total de réponses. Ce format compact et informatif permet aux utilisateurs de rapidement évaluer et interagir avec les questions.



home

La page d'accueil home est conçue pour offrir une vue d'ensemble conviviale de la plateforme. Elle comporte une barre supérieure avec le logo du site, un champ de recherche, et un menu de connexion qui se transforme en nom de profil une fois connecté, ainsi qu'une cloche de notification. Le corps principal de la page d'accueil affiche le nombre total de questions, un bouton pour poser une question, et un menu de tri per-

mettant de trier les questions par date ou par vote. En dessous de ce menu, une liste paginée de feed_component est présentée, affichant jusqu'à 10 questions à la fois. Le composant subject se trouve à droite, tandis que le composant activity_component est à gauche, assurant une navigation équilibrée et intuitive. Page de Formulaire de Question Cette page permet aux utilisateurs de poser de nouvelles questions. Ils peuvent insérer leur question, choisir un sujet, et éventuellement cocher une case pour générer automatiquement une réponse. Cette fonctionnalité encourage la participation en simplifiant le processus de soumission des questions. Page de Description de la Question.

Sur cette page, les utilisateurs peuvent voir la question en haut, suivie des réponses fournies par d'autres utilisateurs ou générées automatiquement en bas. Cette disposition claire et organisée facilite la lecture et l'interaction avec les discussions en cours. Page de Connexion et d'Inscription.

Les pages de connexion et d'inscription permettent aux utilisateurs de créer un compte ou de se connecter à la plateforme. Ces pages sont conçues pour être simples et sécurisées, garantissant une expérience utilisateur fluide.

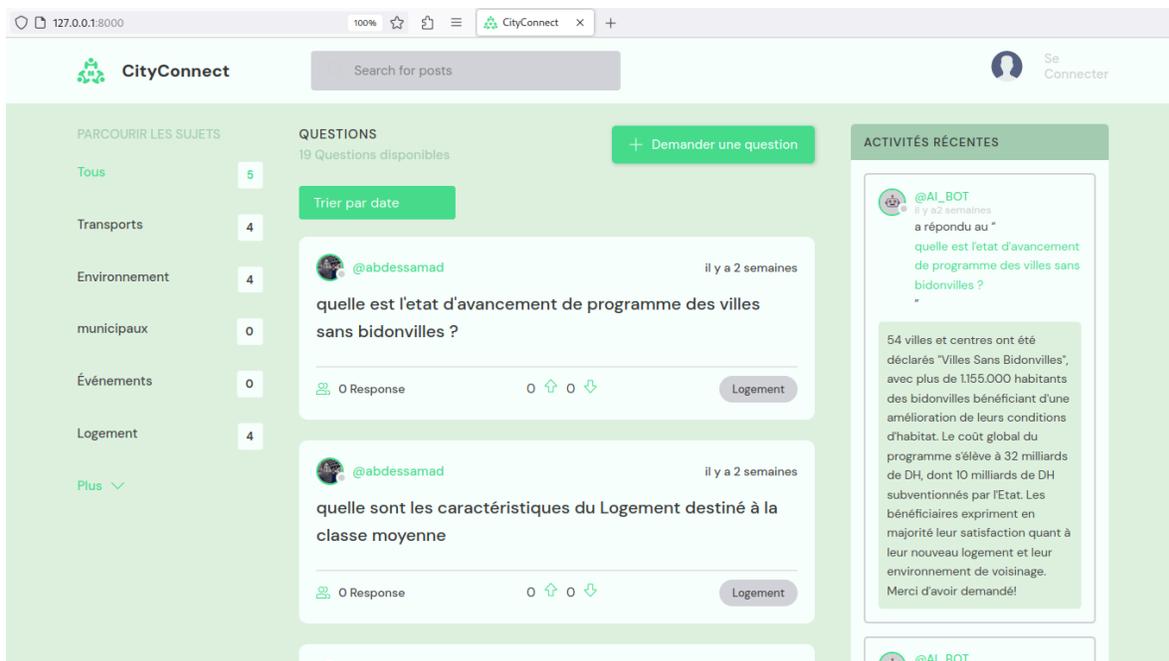


FIGURE 4.4 – Capture d'écran du page Home

Page de Profil

La page de profil permet aux utilisateurs de modifier leurs informations personnelles, d'afficher leur historique d'activités et d'accéder à d'autres fonctionnalités liées à leur compte. Cette page offre un espace personnalisé où les utilisateurs peuvent gérer leur présence sur la plateforme.

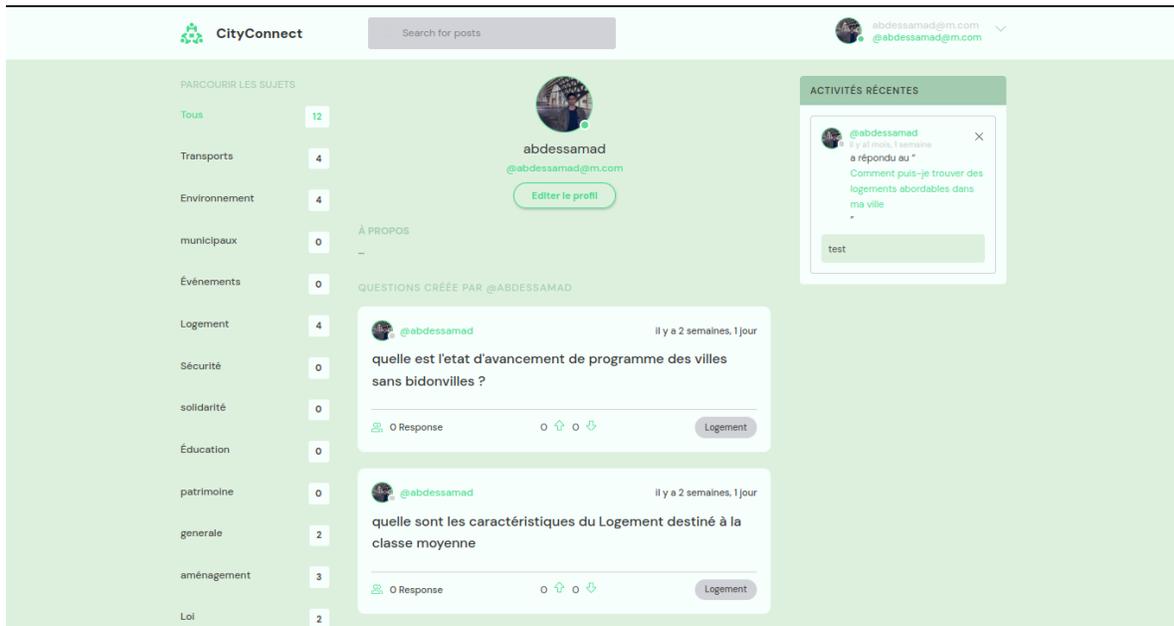


FIGURE 4.5 – Capture d'écran du page profile

Formulaire de Soumission de Questions

La page du formulaire de soumission de questions est conçue pour être simple et intuitive, permettant aux utilisateurs de poser facilement leurs questions. Cette page comprend plusieurs éléments clés pour faciliter la soumission et la gestion des questions :

- **Champ de texte pour la question** : Un champ de texte où les utilisateurs peuvent saisir leur question en détail. Ce champ est conçu pour accepter des entrées longues, permettant aux utilisateurs de poser des questions complexes et bien expliquées.
- **Menu de catégories** : Un menu déroulant permettant aux utilisateurs de sélectionner la catégorie la plus appropriée pour leur question. Les catégories disponibles incluent des sujets tels que le droit, le transport, le logement, et l'environnement. Cette fonctionnalité aide à organiser les questions et à les rendre plus facilement consultables par d'autres utilisateurs.
- **Case à cocher pour la génération automatique de réponse** : Une case à cocher optionnelle permettant aux utilisateurs de demander une réponse automatique générée par le système basé sur les modèles de langage de grande

échelle (LLMs). Si cette option est sélectionnée, le système tentera de fournir une réponse immédiate à la question posée.

- **Bouton de soumission** : Un bouton de soumission qui, une fois cliqué, envoie la question, la catégorie sélectionnée et l'option de génération automatique de réponse au système pour traitement. Le bouton de soumission déclenche également la validation des entrées pour s'assurer que tous les champs obligatoires sont remplis correctement avant l'envoi.

Ces éléments permettent aux utilisateurs de poser leurs questions de manière structurée et efficace, tout en offrant la possibilité d'obtenir des réponses automatiques grâce à l'intégration des LLMs. La simplicité et la clarté de ce formulaire contribuent à une expérience utilisateur fluide et satisfaisante.

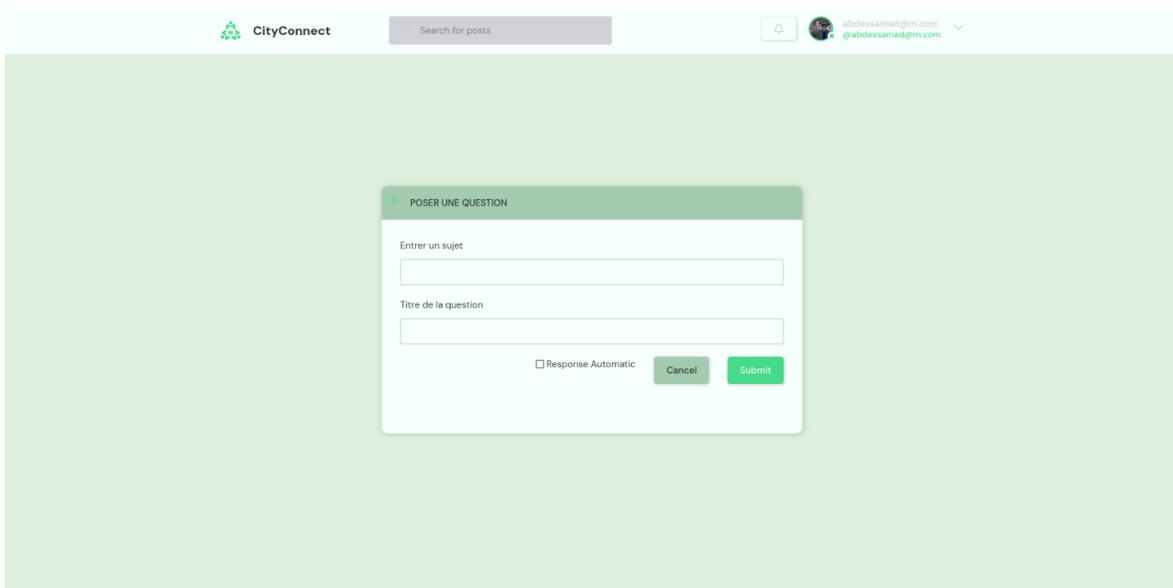


FIGURE 4.6 – Capture d'écran du page formulaire de questions

Page de Room

La page de room est une composante essentielle de la plateforme, conçue pour faciliter les discussions et interactions autour d'une question spécifique. Cette page offre une vue détaillée de la question posée et permet aux utilisateurs de participer activement à la conversation. Voici les principaux éléments de la page de room :

- **Affichage de la question** : La question posée est affichée en haut de la page, de manière claire et visible. Cela inclut le texte complet de la question, permettant aux participants de comprendre immédiatement le sujet de la discussion.
- **Informations sur l'auteur de la question** : Juste en dessous de la question, les informations sur l'utilisateur qui l'a posée sont affichées. Cela comprend le nom de l'auteur, son image de profil, et l'heure à laquelle la question a été

posée. Ces informations contextualisent la question et permettent aux autres utilisateurs de voir qui en est l'initiateur.

- **Participants à la discussion** : La page de room affiche également les participants à la discussion. Cela inclut une liste des utilisateurs qui ont répondu à la question ou interagi avec elle d'une manière ou d'une autre. Les informations sur les participants comprennent leur nom et leur image de profil, ce qui facilite l'identification et l'interaction entre les utilisateurs.
- **Section des réponses** : Sous la question et les informations sur l'auteur, une section dédiée aux réponses est disponible. Les réponses fournies par les autres utilisateurs ou générées automatiquement sont listées ici. Chaque réponse affiche le contenu complet, le nom de l'auteur de la réponse, et l'heure de publication.
- **Formulaire de réponse** : Un formulaire est disponible pour permettre aux utilisateurs de soumettre leurs propres réponses à la question. Ce formulaire comprend un champ de texte pour saisir la réponse et un bouton de soumission.
- **Interactions supplémentaires** : Les utilisateurs peuvent également interagir avec les réponses en votant pour celles qu'ils trouvent les plus utiles (upvote/downvote) et en ajoutant des commentaires pour fournir des clarifications ou des informations supplémentaires.

La conception de la page de room vise à encourager la participation active et à faciliter les discussions constructives. En exposant clairement la question, les informations sur l'auteur et les participants, ainsi que les réponses, la page crée un environnement propice à l'échange d'idées et à la résolution collective des problèmes.



FIGURE 4.7 – Capture d'écran du page room

Compatibilité avec les Appareils à Petit Écran



FIGURE 4.8 – Capture d'écran du page home dans une appareils à petit écran

Une des priorités majeures dans la conception de notre plateforme a été d'assurer une compatibilité optimale avec les appareils à petit écran, tels que les téléphones portables et les tablettes. La compatibilité mobile est cruciale pour permettre aux utilisateurs d'accéder aux fonctionnalités de la plateforme de manière fluide et efficace, peu importe l'appareil utilisé. Voici les principales mesures mises en place pour garantir cette compatibilité :

Design Responsive

Le design de la plateforme a été développé en utilisant des techniques de design responsive, ce qui signifie que l'interface utilisateur s'adapte automatiquement à la taille de l'écran de l'appareil utilisé. Cela est réalisé grâce à l'utilisation de CSS flexibles et de grilles, ainsi qu'à des media queries qui ajustent les styles en fonction des dimensions de l'écran.

Menus et Navigation Adaptatifs

Les menus et les éléments de navigation ont été conçus pour être facilement accessibles sur les petits écrans. Par exemple, les menus déroulants se transforment en icônes de menu hamburger sur les appareils mobiles, permettant aux utilisateurs de naviguer facilement sans encombrer l'écran. Les boutons et les liens sont également dimensionnés de manière appropriée pour les interactions tactiles.

Optimisation des Images et des Médias

Les images et les autres contenus multimédias sont optimisés pour les petits écrans. Cela inclut l'utilisation d'images de taille réduite et la mise en œuvre de techniques de chargement adaptatif, où des images de résolution différente sont servies en fonction de la taille de l'écran et de la résolution de l'appareil.

Disposition des Contenus

La disposition des contenus sur les pages a été repensée pour les petits écrans. Les éléments sont empilés verticalement au lieu d'être alignés horizontalement, assurant ainsi une lecture et une interaction confortables sans avoir besoin de zoomer ou de faire défiler horizontalement. Les textes sont formatés pour être facilement lisibles, avec des tailles de police adaptées et des espacements suffisants.

Grâce à ces mesures, nous avons réussi à créer une plateforme qui est non seulement fonctionnelle sur les appareils de bureau, mais qui offre également une expérience utilisateur agréable et efficace sur les téléphones portables et les tablettes. Cela garantit que tous les utilisateurs peuvent accéder aux services et aux fonctionnalités de la plateforme, quel que soit l'appareil qu'ils utilisent.

4.2.2 Backend

Présentation de Django

Django est un framework web de haut niveau en Python qui encourage le développement rapide et la conception propre et pragmatique. Créé pour simplifier le processus de création d'applications web complexes, Django permet aux développeurs de se concentrer sur l'écriture de nouvelles fonctionnalités plutôt que de réinventer la roue.

Rôle de Django dans le Développement d'Applications Web

Django joue un rôle crucial dans le développement d'applications web en offrant une structure solide et des outils puissants pour gérer les aspects backend des projets. En utilisant Django, les développeurs peuvent construire des applications web robustes, évolutives et sécurisées plus rapidement et plus efficacement. Son architecture cohérente et ses nombreuses fonctionnalités intégrées facilitent la gestion de toutes les parties d'un site web, de la base de données à l'interface utilisateur.

Fonctionnalités Clés de Django

Architecture MVC (Model-View-Controller)

Django suit le modèle MVT (Model-View-Template), qui est une variante de l'architecture MVC. Cette structure sépare les différentes couches d'une application web pour une meilleure organisation et maintenabilité :

- **Model** : Représente les données de l'application et la logique de la base de données. Utilise l'ORM de Django pour interagir avec la base de données de manière intuitive.
- **View** : Contient la logique qui traite les requêtes HTTP et renvoie des réponses, souvent sous forme de pages web ou de données JSON.
- **Template** : Gère l'affichage des données et la présentation de l'interface utilisateur. Les templates Django utilisent un système de balisage simple pour insérer dynamiquement les données dans le HTML.

ORM (Object-Relational Mapping)

L'ORM de Django permet aux développeurs de définir les modèles de données en Python et de les mapper automatiquement aux tables de la base de données. Cela simplifie considérablement les interactions avec la base de données, en éliminant la nécessité d'écrire des requêtes SQL complexes. Les fonctionnalités clés de l'ORM incluent :

- Définition de modèles via des classes Python
- Migration automatique de la base de données pour refléter les changements des modèles
- Requêtes de base de données sécurisées et optimisées

Interface d'Administration Intégrée

Django fournit une interface d'administration générée automatiquement, qui permet de gérer les données de l'application de manière simple et efficace. Cette interface est hautement personnalisable et offre une gamme complète de fonctionnalités pour les opérations CRUD (Create, Read, Update, Delete) sur les modèles de données. Les caractéristiques principales de l'interface d'administration incluent : de

- Gestion des utilisateurs et des autorisations
- Personnalisation de l'interface et des actions disponibles
- Visualisation et modification des données sans besoin de développement supplémentaire

Principaux Répertoires et Fichiers de Django

Lors du développement d'une application web avec Django, la structure des répertoires joue un rôle crucial pour organiser le code et les fichiers de manière cohérente et maintenable. Voici une explication des principaux répertoires et fichiers d'une application Django et leurs objectifs :

Répertoire settings (paramètres)

Le fichier `settings.py` contient toutes les configurations nécessaires pour le projet Django. Il définit les paramètres globaux tels que :

- **BASE_DIR** : Chemin de base du projet.
- **DEBUG** : Indique si le mode debug est activé.
- **ALLOWED_HOSTS** : Liste des hôtes/domains autorisés à servir le projet.
- **INSTALLED_APPS** : Liste des applications Django activées dans ce projet.
- **MIDDLEWARE** : Liste des middleware utilisés pour le traitement des requêtes/réponses.
- **DATABASES** : Configuration de la base de données.
- **TEMPLATES** : Configuration des moteurs de template utilisés.
- **STATIC_URL** et **STATICFILES_DIRS** : Chemin pour les fichiers statiques.
- **MEDIA_URL** et **MEDIA_ROOT** : Chemin pour les fichiers médias.

Répertoire urls (URLs)

Le fichier `urls.py` définit les URL de l'application et les vues associées. Il utilise la fonction `path()` pour mapper les URL aux vues correspondantes. Ce fichier sert de routeur principal pour l'application, permettant de diriger les requêtes HTTP vers les fonctions ou classes de vue appropriées.

Répertoire static (fichiers statiques)

Le répertoire `static` contient les fichiers statiques tels que les images, les fichiers CSS et JavaScript. Ces fichiers sont servis aux utilisateurs tels quels, sans traitement par le serveur. La configuration des fichiers statiques se fait via les paramètres `STATIC_URL` et `STATICFILES_DIRS` dans `settings.py`.

Répertoire media (fichiers médias)

Le répertoire `media` est utilisé pour stocker les fichiers téléchargés par les utilisateurs, comme les images de profil, les documents, etc. Les chemins et URL associés aux fichiers médias sont définis par les paramètres `MEDIA_URL` et `MEDIA_ROOT` dans `settings.py`.

Répertoire templates (gabarits)

Le répertoire `templates` contient les fichiers de gabarits HTML utilisés pour rendre les vues. Les gabarits sont des fichiers HTML avec des balises spéciales permettant d'insérer dynamiquement des données. La configuration des gabarits est spécifiée dans la section `TEMPLATES` de `settings.py`.

Répertoire apps (applications)

Le répertoire `apps` contient les différentes applications (ou modules) de l'application Django. Chaque application est une unité modulaire qui regroupe des modèles, des vues, des templates et des fichiers statiques spécifiques à une fonctionnalité. Le fichier `apps.py` à la racine de chaque application configure l'application pour le projet.

Répertoire migrations (migrations)

Le répertoire `migrations` contient les fichiers de migration pour les modèles de l'application. Les migrations permettent de gérer les modifications apportées aux modèles de la base de données de manière incrémentielle. Elles sont créées et appliquées à l'aide des commandes `makemigrations` et `migrate`.

Répertoire `manage.py`

Le fichier `manage.py` est un utilitaire en ligne de commande qui permet d'interagir avec le projet Django de diverses manières. Il permet de lancer le serveur de développement, de créer des applications, d'exécuter des migrations, et bien plus encore.

Fichier `views.py`

Le fichier `views.py` contient les vues de l'application Django, qui définissent la logique de traitement des requêtes et de génération des réponses. Chaque vue est une fonction ou une classe qui prend en entrée une requête HTTP et retourne une réponse HTTP. Voici une explication détaillée des principales vues implémentées dans ce fichier.

`loginPage`

La vue `loginPage` gère l'authentification des utilisateurs. Si l'utilisateur est déjà authentifié, il est redirigé vers la page d'accueil. Si une requête POST est reçue, la vue tente d'authentifier l'utilisateur avec les informations fournies. En cas de succès, l'utilisateur est connecté et redirigé vers la page d'accueil. Sinon, un message d'erreur est affiché.

`logoutUser`

La vue `logoutUser` déconnecte l'utilisateur et le redirige vers la page d'accueil.

`registerPage`

La vue `registerPage` gère l'inscription des nouveaux utilisateurs. Si une requête POST est reçue avec des données valides, un nouvel utilisateur est créé, connecté, puis redirigé vers la page d'accueil. En cas d'erreur, un message d'erreur est affiché.

`home`

La vue `home` affiche la page d'accueil de l'application. Elle filtre les questions (salles) selon les critères de recherche et de tri spécifiés par l'utilisateur, puis les présente pour une meilleure lisibilité. Les sujets populaires et les messages récents sont également affichés.

`room`

La vue `room` affiche les détails d'une salle spécifique, y compris les messages et les participants. Les utilisateurs peuvent également ajouter de nouveaux messages.

`userProfile`

La vue `userProfile` affiche le profil d'un utilisateur, y compris ses salles, ses messages et les sujets suivis.

`upvote_room` et `downvote_room`

Ces vues permettent aux utilisateurs de voter pour ou contre une salle. Elles gèrent la création et la mise à jour des votes, ainsi que le comptage des votes.

`createRoom`, `updateRoom` et `deleteRoom`

Ces vues permettent de créer, mettre à jour et supprimer des salles. La vue `createRoom` permet également de générer automatiquement une réponse via un utilisateur "bot" si l'option est sélectionnée.

`deleteMessage`

Cette vue permet de supprimer un message. Seul l'utilisateur qui a créé le message peut le supprimer.

`updateUser`

Cette vue permet aux utilisateurs de mettre à jour leurs informations de profil.

`topicsPage` et `activityPage`

Ces vues affichent les sujets et les activités récentes sur la plateforme.

Le fichier `views.py` de Django contient les vues qui gèrent la logique de traitement des requêtes et de génération des réponses de l'application. Les principales vues incluent la gestion de l'authentification (`loginPage`, `logoutUser`, `registerPage`), l'affichage des contenus (`home`, `room`, `userProfile`), et la gestion des interactions des utilisateurs avec les salles et les messages (`createRoom`, `updateRoom`, `deleteRoom`, `upvote_room`,

downvote_room, deleteMessage, updateUser). Ces vues permettent de créer une expérience utilisateur fluide et interactive tout en assurant une gestion sécurisée et efficace des données.

Interface d'administration de Django

L'interface d'administration de Django est un outil puissant qui permet de gérer facilement les utilisateurs, les données et les configurations de l'application. Elle offre une interface graphique intuitive qui simplifie les tâches administratives courantes.

Page de connexion administrateur

La page de connexion administrateur est la porte d'entrée de l'interface d'administration de Django. Les administrateurs peuvent s'authentifier en utilisant leurs identifiants pour accéder à l'ensemble des fonctionnalités d'administration.

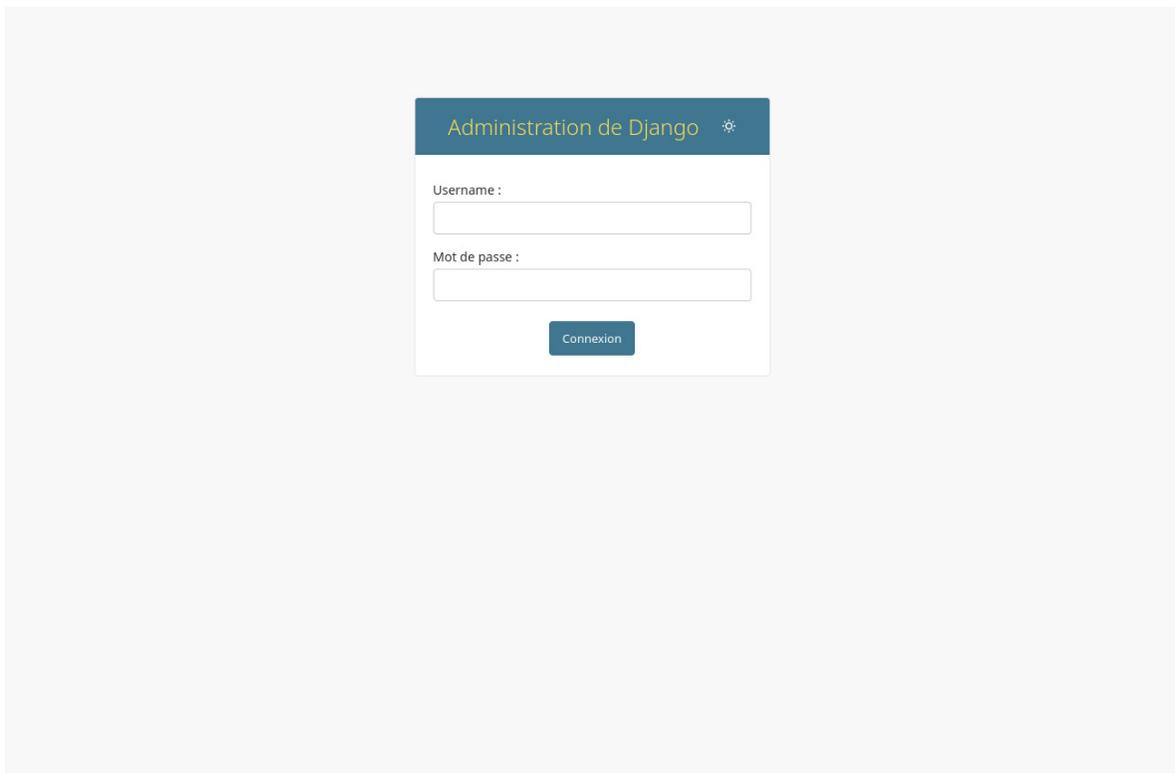


FIGURE 4.9 – Page de connexion administrateur

Liste des utilisateurs

Une fois connecté, l'administrateur peut accéder à la liste des utilisateurs inscrits sur la plateforme. Cette interface permet de visualiser, ajouter, modifier ou supprimer des utilisateurs. Elle offre également des fonctionnalités de recherche et de filtrage pour une gestion efficace des utilisateurs.

Fonctionnalités supplémentaires

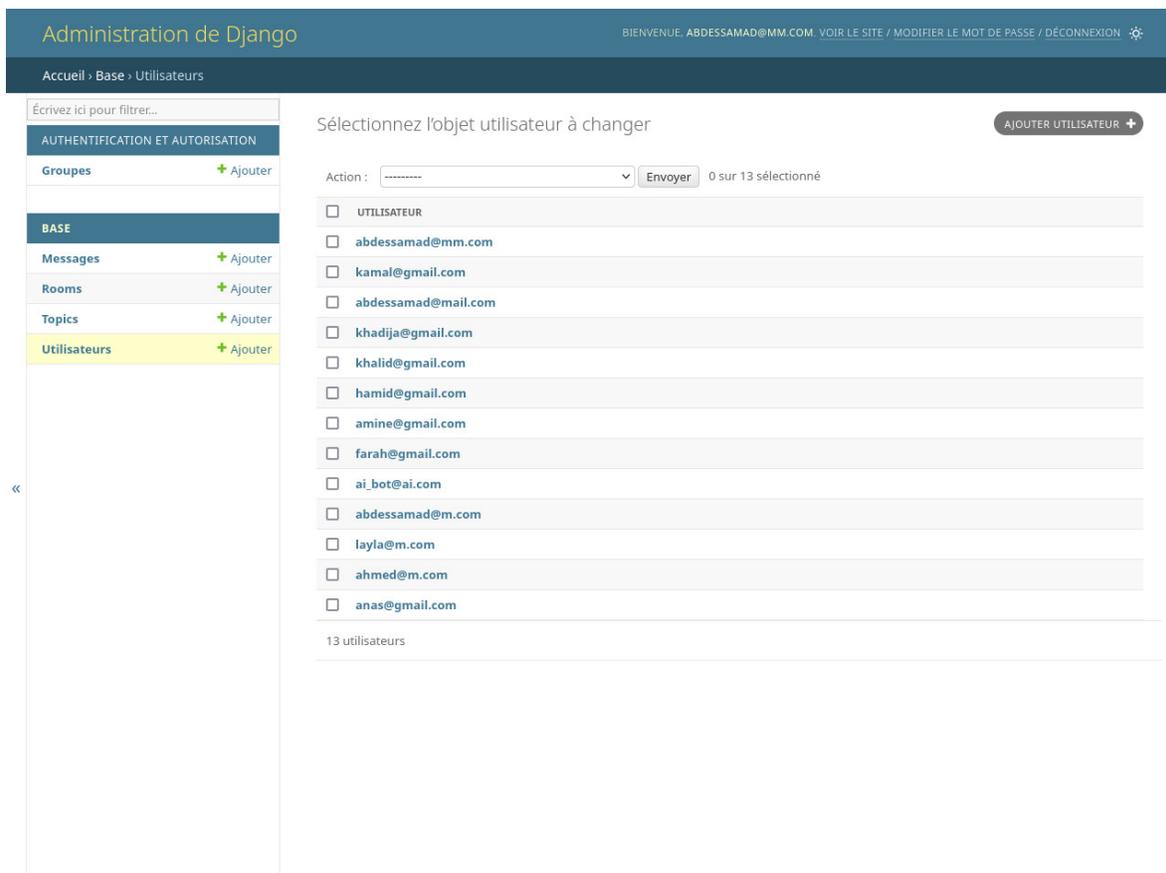


FIGURE 4.10 – Liste des utilisateurs dans l’interface d’administration

En plus de la gestion des utilisateurs, l’interface d’administration de Django permet de gérer d’autres aspects de l’application, tels que les modèles de données, les permissions et les configurations globales. Elle offre des vues personnalisables et extensibles, permettant aux développeurs de créer des outils d’administration adaptés aux besoins spécifiques de l’application.

Personnalisation de l’interface

L’interface d’administration de Django peut être personnalisée pour répondre aux exigences spécifiques de l’application. Les développeurs peuvent ajouter de nouvelles fonctionnalités, modifier l’apparence et l’agencement des pages, et créer des vues d’administration personnalisées pour une gestion plus intuitive et efficace.

L’utilisation de l’interface d’administration de Django facilite grandement la gestion quotidienne de l’application et permet aux administrateurs de se concentrer sur les tâches essentielles, tout en assurant une gestion des utilisateurs et des données efficace et sécurisée.

4.2.3 Base de Données

Pour ce projet, nous avons utilisé la base de données par défaut de Django, SQLite. SQLite est une base de données légère, intégrée, et facile à configurer, ce qui en fait un choix idéal pour le développement et les tests. Voici une description des principaux modèles de données utilisés dans notre application.

Modèle User

Le modèle `User` est fourni par Django pour gérer les utilisateurs. Il inclut des champs tels que `username`, `email`, `password`, et d'autres informations personnelles. Les utilisateurs peuvent s'inscrire, se connecter, et interagir avec la plateforme grâce à ce modèle.

Modèle Room

Le modèle `Room` représente une salle de discussion ou une question posée par un utilisateur. Il inclut des champs tels que `name`, `description`, `created`, `updated`, `host` (référence à un `User`), et `topic` (référence à un `Topic`). Les utilisateurs peuvent créer, modifier, et supprimer des salles.

Modèle Message

Le modèle `Message` représente un message posté dans une salle. Il inclut des champs tels que `body`, `created`, `updated`, `user` (référence à un `User`), et `room` (référence à une `Room`). Les utilisateurs peuvent ajouter et supprimer des messages.

Modèle Topic

Le modèle `Topic` représente les sujets de discussion disponibles sur la plateforme. Il inclut des champs tels que `name`. Les utilisateurs peuvent choisir des sujets pour leurs salles de discussion afin de faciliter la catégorisation et la recherche.

Modèle Vote

Le modèle `Vote` permet de gérer les votes des utilisateurs sur les salles de discussion. Il inclut des champs tels que `user` (référence à un `User`), `room` (référence à une `Room`), et `value` (indiquant un vote positif ou négatif).



FIGURE 4.11 – Diagramme de l'architecture de la base de données

Gestion des Migrations

Django utilise un système de migrations pour gérer les modifications apportées aux modèles de données. Les migrations permettent de synchroniser la structure de la base de données avec les définitions des modèles Python. Les commandes `makemigrations` et `migrate` sont utilisées pour créer et appliquer les migrations respectivement, garantissant ainsi que la base de données est toujours à jour avec le code de l'application.

En utilisant SQLite et les outils intégrés de gestion de base de données de Django, nous avons pu développer et tester efficacement notre application, tout en maintenant une structure de données claire et bien organisée.

4.3 Fonctionnalités supplémentaires

En plus des fonctionnalités principales de discussion, de dépôt de plaintes et de réponse automatique, la plateforme inclut plusieurs fonctionnalités supplémentaires pour améliorer l'expérience utilisateur et répondre aux divers besoins des citoyens. Voici un aperçu des principales fonctionnalités supplémentaires implémentées :

4.3.1 Catégories (droit, transport, logement, environnement)

Pour faciliter la navigation et la recherche d'informations, la plateforme organise les discussions et les questions en différentes catégories, telles que le droit, le transport, le logement et l'environnement. Chaque catégorie regroupe les sujets et les questions pertinents, permettant aux utilisateurs de trouver rapidement les informations spécifiques à leurs intérêts ou préoccupations. Cette organisation thématique aide également à structurer les données et à améliorer la pertinence des réponses automatiques.

4.3.2 Conception de l'interface utilisateur

L'interface utilisateur a été conçue pour être intuitive, réactive et conviviale. Elle inclut des éléments visuels et des fonctionnalités interactives pour offrir une expérience utilisateur agréable et efficace. Les principaux composants de l'interface utilisateur comprennent :

- **Barre supérieure** : Contient le logo du site, un champ de recherche, un menu de connexion (qui se transforme en profil utilisateur une fois connecté) et une cloche de notification.
- **Page d'accueil** : Affiche le nombre total de questions, un bouton pour poser une question, un menu de tri permettant de trier les questions par date ou par vote, et une liste paginée des questions.
- **Composants latéraux** : Incluent un composant d'activités récentes (`Activity_component.html`) à gauche et un composant de sujets (`subject.html`) à droite.
- **Formulaire de soumission de question** : Permet aux utilisateurs d'insérer leur question, de choisir un sujet et d'opter pour une réponse automatique.

- **Page de description de la question** : Affiche la question en haut et les réponses fournies par d'autres utilisateurs ou générées automatiquement en bas.

Cette conception assure que les utilisateurs peuvent naviguer facilement sur la plateforme, accéder rapidement aux informations nécessaires et interagir efficacement avec les autres utilisateurs.

4.3.3 Système de vote (up/down)

Pour promouvoir la participation active et la qualité des réponses, la plateforme intègre un système de vote (up/down) pour les questions et les réponses. Les utilisateurs peuvent voter positivement (upvote) ou négativement (downvote) sur les contributions, influençant ainsi leur visibilité et leur pertinence.

Les fonctionnalités principales du système de vote incluent :

- **Flèches de vote** : Chaque question et réponse est accompagnée de flèches de vote vers le haut et vers le bas, permettant aux utilisateurs de voter facilement.
- **Compteur de votes** : Affiche le nombre total de votes positifs et négatifs pour chaque question et réponse.
- **Classement par votes** : Les questions et réponses peuvent être triées par le nombre de votes, mettant en avant les contributions les plus appréciées et pertinentes.

Ce système de vote encourage les utilisateurs à contribuer de manière constructive et aide à maintenir la qualité des discussions sur la plateforme.

En intégrant ces fonctionnalités supplémentaires, la plateforme offre une expérience utilisateur enrichie, répondant aux besoins variés des citoyens tout en facilitant l'accès à des informations pertinentes et la participation active dans les discussions communautaires.

4.3.4 composant LLM

Dans le cadre de ce projet, nous avons choisi d'intégrer des Modèles de Langage de Grande Échelle (LLMs) pour fournir des réponses automatiques aux questions posées par les utilisateurs. Les LLMs offrent des capacités avancées de compréhension et de génération de texte, ce qui permet d'améliorer considérablement l'interaction et l'expérience utilisateur sur la plateforme. Parmi les différentes options disponibles, nous avons opté pour les modèles d'OpenAI, en particulier le modèle `gpt-3.5-turbo-0125`, en raison de sa performance et de sa fiabilité démontrées dans diverses applications.

OpenAI est reconnu pour ses modèles de langage avancés, capables de comprendre des contextes complexes et de générer des réponses pertinentes. Leur API est bien documentée et offre une intégration facile avec diverses applications, ce qui en fait un choix idéal pour notre plateforme. En utilisant les modèles d'OpenAI, nous pouvons

fournir des réponses précises et concises aux utilisateurs, tout en s'assurant que les informations sont cohérentes et utiles.

Par ailleurs, pour faciliter l'intégration et la gestion des interactions complexes entre les composants de traitement de langage, nous avons utilisé le framework LangChain. LangChain simplifie la création de chaînes de traitements complexes en combinant différentes étapes de manipulation et de traitement de données. Cela nous permet d'organiser les différentes étapes nécessaires à l'extraction, la préparation et la génération de réponses de manière modulaire et maintenable.

Les détails techniques de l'implémentation de l'intégration avec les Modèles de Langage de Grande Échelle (LLMs) utilisant LangChain et l'API d'OpenAI seront expliqués en profondeur dans le chapitre suivant. Ce chapitre décrira les étapes spécifiques, le code utilisé, ainsi que les défis rencontrés et les solutions apportées pour assurer une intégration fluide et efficace de ces technologies avancées dans notre plateforme.

Chapitre 5

Intégration avec les Modèles de Langage (LLMs)

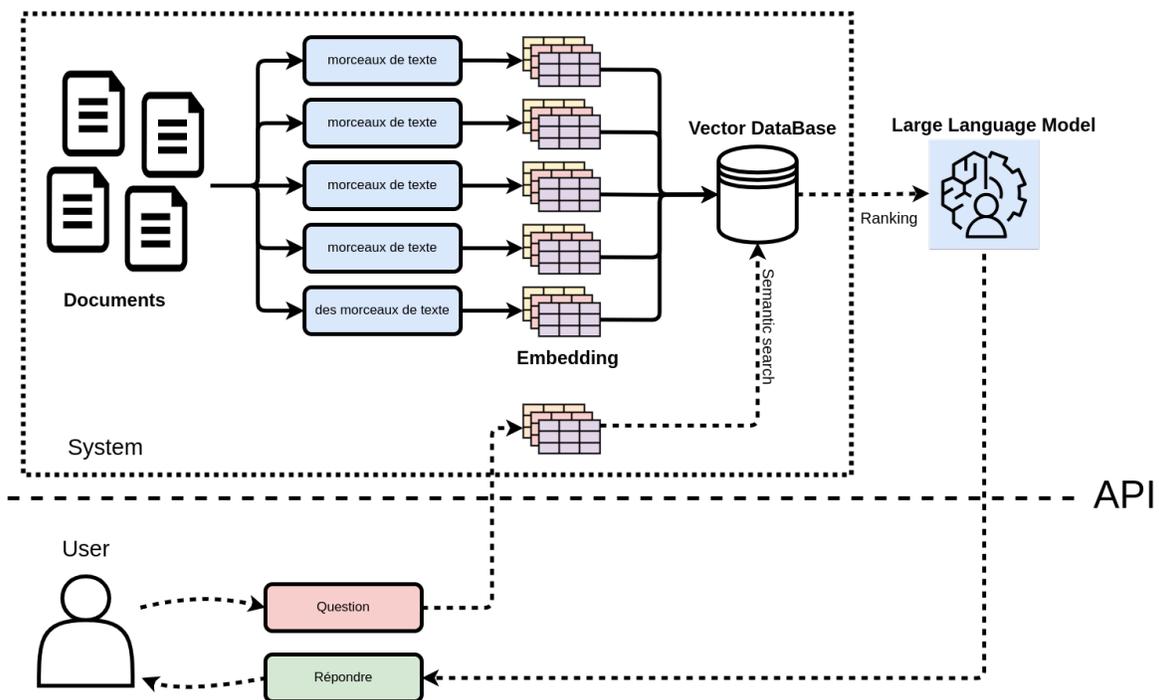


FIGURE 5.1 – Diagramme de l'architecture de composante LLM

5.1 Système de Réponse Automatique

5.1.1 Préparation du jeu de données

La préparation du jeu de données pour le système de réponse automatique est une étape cruciale pour assurer des réponses précises et pertinentes. Ce processus comprend plusieurs étapes, allant de l'extraction et de la traduction du texte à la transformation des textes en embeddings, puis à leur stockage dans une base de données vectorielle.

Extraction du texte et traduction si nécessaire

La première étape consiste à extraire le texte pertinent à partir de diverses sources, notamment des documents locaux et des pages web officielles. Nous utilisons des outils pour charger les documents texte depuis des répertoires locaux et pour extraire les informations des pages web. Si nécessaire, les textes extraits sont traduits dans la langue souhaitée pour garantir leur compréhension et leur pertinence.

Transformation du texte en embeddings

Après l'extraction et la traduction, les textes sont transformés en embeddings en utilisant le modèle d'embeddings d'OpenAI. Les embeddings sont des représentations vectorielles des textes qui capturent les significations et les relations sémantiques de manière compacte et efficace. Cette transformation permet de convertir les textes en formats utilisables pour des opérations de recherche et de similarité.

Stockage dans une base de données vectorielle

Une fois les embeddings générés, ils sont stockés dans une base de données vectorielle. Nous utilisons une base de données vectorielle spécialisée qui permet une recherche rapide et efficace des embeddings par similarité. Ce stockage assure que le système de réponse automatique dispose d'une base de connaissances structurée et facilement accessible pour fournir des réponses pertinentes et précises aux questions des utilisateurs.

Ces étapes de préparation du jeu de données garantissent que le système est bien équipé pour comprendre et répondre aux requêtes des utilisateurs de manière efficace et fiable.

5.1.2 Comparaison entre RAG et Fine Tuning

Lors de l'intégration des modèles de langage de grande échelle (LLMs) dans une application, deux approches principales peuvent être envisagées : le **Retrieval Augmented Generation** (RAG) et le **fine tuning** des modèles. Chaque méthode présente ses propres avantages et inconvénients, et le choix entre les deux dépend des besoins spécifiques du projet.

Retrieval Augmented Generation (RAG)

RAG combine la génération de texte par les LLMs avec la récupération d'informations pertinentes à partir d'une base de données ou d'un corpus de documents. Cette approche permet d'enrichir les réponses générées par le modèle avec des informations contextuelles précises et actualisées.

Avantages de RAG :

- **Flexibilité des données** : Les données peuvent être facilement mises à jour ou ajoutées sans nécessiter de réentraîner le modèle. Cela est particulièrement utile lorsque les informations changent fréquemment.
- **Réduction des coûts de calcul** : Éviter le réentraînement complet du modèle permet de réduire les coûts en termes de temps et de ressources informatiques.
- **Amélioration continue** : La qualité des réponses peut être améliorée en enrichissant simplement la base de données ou le corpus de documents utilisés pour la récupération.

Fine Tuning

Le fine tuning consiste à adapter un modèle pré-entraîné sur un jeu de données spécifique en l'entraînant davantage sur des données supplémentaires spécifiques à l'application. Cela permet de spécialiser le modèle pour des tâches ou des contextes particuliers.

Avantages du Fine Tuning :

- **Personnalisation profonde** : Permet d'adapter le modèle de manière très spécifique aux besoins de l'application.
- **Performance optimisée** : Peut potentiellement améliorer les performances du modèle pour des tâches très spécifiques.

Pourquoi RAG est Mieux pour Notre Situation

Dans le contexte de notre plateforme, le RAG présente plusieurs avantages significatifs par rapport au fine tuning, en raison des limitations de données et de la nécessité

de mises à jour fréquentes.

Limitations de Données :

- **Quantité de données :** Nous disposons d'un volume limité de données spécifiques à l'urbanisme et à la participation citoyenne. Le fine tuning nécessite généralement de grandes quantités de données pour être efficace, ce qui peut être un obstacle dans notre cas.
- **Variabilité des informations :** Les informations pertinentes pour notre plateforme peuvent varier considérablement et changer fréquemment, nécessitant une flexibilité que le fine tuning ne peut pas offrir facilement.

Facilité de Mise à Jour :

- **Mises à jour fréquentes :** Avec RAG, les données peuvent être mises à jour ou enrichies sans avoir à réentraîner le modèle. Cela est crucial pour maintenir la pertinence et l'actualité des réponses fournies par la plateforme.
- **Adaptation rapide :** Les nouvelles informations peuvent être intégrées rapidement, permettant au système de s'adapter en temps réel aux changements dans les données ou les besoins des utilisateurs.

En conclusion, le RAG est mieux adapté à notre situation en raison de la flexibilité qu'il offre pour gérer des données limitées et changeantes. Cette approche permet à notre plateforme de fournir des réponses précises et actualisées sans les coûts et les contraintes associés au fine tuning.

5.1.3 Techniques d'ingénierie de prompt

L'ingénierie de prompt est une composante essentielle de l'utilisation des modèles de langage de grande échelle (LLMs). Elle consiste à concevoir et à structurer les prompts, ou instructions, donnés aux modèles pour obtenir des réponses précises et pertinentes. Plusieurs techniques d'ingénierie de prompt peuvent être utilisées pour améliorer l'efficacité et la qualité des réponses générées par les LLMs. Parmi ces techniques, on trouve le **Prompt Chaining**, le **Retrieval Augmented Generation (RAG)**, le **Automatic Reasoning and Tool-use**, et le **ReAct**.

Prompt Chaining

consiste à utiliser une série de prompts successifs, où la sortie d'un modèle est utilisée comme entrée pour le prompt suivant. Cette technique permet de décomposer des tâches complexes en étapes plus gérables et de guider progressivement le modèle vers la solution finale.

Retrieval Augmented Generation

(RAG) combine la génération de texte par les modèles de langage avec la récupération d'informations à partir d'une base de données. Cette technique permet de fournir des réponses enrichies par des informations contextuelles pertinentes, améliorant ainsi la précision et la pertinence des réponses générées.

Automatic Reasoning and Tool-use

implique l'utilisation de modèles de langage pour effectuer des raisonnements automatiques ou utiliser des outils externes pour accomplir des tâches spécifiques. Cette approche permet aux modèles de langage de dépasser leurs capacités initiales en s'appuyant sur des outils spécialisés pour des résultats optimaux.

ReAct

est une technique où le modèle de langage réagit de manière interactive aux instructions et aux informations fournies, ajustant ses réponses en fonction des nouvelles données reçues. Cette méthode est particulièrement utile pour les scénarios nécessitant des interactions dynamiques et adaptatives.

Dans le cadre de ce projet, nous avons utilisé principalement deux techniques : le **Prompt Chaining** et le **Retrieval Augmented Generation (RAG)**. Le **Prompt Chaining** a été employé pour structurer les interactions complexes en étapes successives, assurant une cohérence et une précision accrues dans les réponses générées. Le **Retrieval Augmented Generation** a permis d'enrichir les réponses avec des informations contextuelles pertinentes, récupérées à partir d'une base de données, pour fournir des réponses plus complètes et informatives aux utilisateurs.

5.1.4 Intégration avec les LLMs implementation

L'intégration avec les Modèles de Langage de Grande Échelle (LLMs) permet à notre plateforme de fournir des réponses automatiques précises et pertinentes aux questions posées par les utilisateurs. Ce processus d'intégration repose sur plusieurs étapes clés, qui assurent une utilisation efficace des LLMs pour répondre aux requêtes des utilisateurs.

Conversion de la question de l'utilisateur en embeddings

Lorsqu'un utilisateur pose une question sur la plateforme, la première étape consiste à convertir cette question en embeddings. Les embeddings sont des représentations vectorielles de la question, capturant les nuances sémantiques et contextuelles du texte.

Cette transformation permet une comparaison efficace avec d'autres textes stockés dans la base de données.

Comparaison des embeddings

Les embeddings de la question de l'utilisateur sont ensuite comparés avec les embeddings des morceaux de texte du jeu de données. Ce jeu de données comprend des documents extraits de sources diverses, préalablement transformés en embeddings et stockés dans une base de données vectorielle. La comparaison est effectuée pour identifier les morceaux de texte les plus similaires à la question posée.

Sélection des morceaux de texte les plus similaires

Parmi tous les morceaux de texte comparés, les trois plus similaires sont sélectionnés. Ces morceaux de texte sont choisis en fonction de leur proximité sémantique avec la question de l'utilisateur. Les embeddings des textes les plus proches sont utilisés pour fournir un contexte pertinent au modèle de langage.

Passage des textes et de la question au LLM

Les trois morceaux de texte sélectionnés sont passés au LLM, accompagnés de la question de l'utilisateur. Ces textes servent de contexte pour le modèle de langage, lui fournissant les informations nécessaires pour générer une réponse précise et cohérente.

Génération de la réponse par le LLM

Le LLM utilise uniquement le contexte fourni par les morceaux de texte similaires pour tenter de répondre à la question de l'utilisateur. Si le contexte contient suffisamment d'informations pertinentes, le modèle génère une réponse appropriée à la question posée. Cela permet de s'assurer que la réponse est basée sur des informations vérifiées et disponibles dans le jeu de données.

Gestion des cas où la réponse n'est pas disponible

Dans le cas où le contexte fourni ne contient pas les informations nécessaires pour répondre à la question, le LLM est programmé pour indiquer qu'il ne sait pas la réponse. Le modèle ne tente pas d'inventer une réponse, assurant ainsi l'intégrité et la fiabilité des informations fournies aux utilisateurs.

Le prompt utilisé pour guider le modèle est le suivant :

Utilisez les éléments de contexte suivants pour répondre à la question à la fin.

Si vous ne connaissez pas la réponse, dites simplement

"Pardonnez-moi je ne sais pas", n'essayez pas d'inventer une réponse.

Utilisez trois phrases maximum et gardez la réponse aussi concise que possible.

{context}

Question: {question}

Cette question concerne seulement le Maroc.

Réponse utile:

Cette intégration avec les LLMs permet à notre plateforme de fournir des réponses automatiques fiables, tout en maintenant une transparence sur les limitations des informations disponibles. Cela renforce la confiance des utilisateurs dans les réponses générées par la plateforme et améliore l'expérience globale des utilisateurs.

5.1.5 Intégration avec le web

Introduction

L'intégration avec le web est une étape cruciale pour garantir que notre plateforme peut fournir des réponses précises et à jour, même lorsque les informations ne sont pas disponibles dans notre base de données interne. Cette capacité à rechercher et à extraire des informations en temps réel à partir du web améliore considérablement la couverture et la pertinence des réponses fournies par la plateforme. Pour y parvenir, il a été nécessaire de retravailler la structure du prompt et de concevoir un système capable de fonctionner comme un agent intelligent.

Fonctionnement de l'Agent Web

Le système fonctionne comme un agent qui suit plusieurs étapes pour traiter les questions des utilisateurs et fournir des réponses pertinentes. Voici comment fonctionne ce processus :

1. **Prétraitement de la question** : Lorsqu'une question est posée, l'agent commence par prétraiter la question pour déterminer s'il a le droit de répondre. Cela inclut des vérifications pour s'assurer que la question respecte les règles et les limitations de la plateforme. Si la question ne peut pas être traitée, l'agent arrête le processus et informe l'utilisateur.
2. **Transformation en embeddings** : Si la question peut être traitée, elle est transformée en embeddings. Cette transformation permet de représenter la question sous forme vectorielle pour des comparaisons de similarité.
3. **Recherche dans le Vector Store** : L'agent compare les embeddings de la question avec ceux des morceaux de texte stockés dans le vector store. Si des réponses pertinentes sont trouvées dans le vector store, elles sont utilisées pour générer une réponse.

4. **Recherche sur le Web** : Si aucune réponse pertinente n'est trouvée dans le vector store, l'agent utilise le moteur de recherche Google pour chercher des réponses potentielles sur le web. Il extrait les trois premiers sites authentiques en utilisant le web scraping.
5. **Extraction et Traitement du Contenu Web** : Le contenu des sites web sélectionnés est extrait en utilisant le LangChain WebLoader. Ce contenu est ensuite découpé en morceaux gérables et transformé en embeddings.
6. **Recherche de Similarité** : Les embeddings des morceaux de contenu web sont comparés à l'embedding de la question pour identifier les morceaux les plus similaires.
7. **Génération de la Réponse** : Les morceaux de texte les plus similaires sont passés au LLM, avec la question initiale comme contexte. Le LLM utilise ce contexte pour générer une réponse finale. Si le contexte fourni ne contient toujours pas les informations nécessaires, le LLM répondra par "Pardon, je ne sais pas".

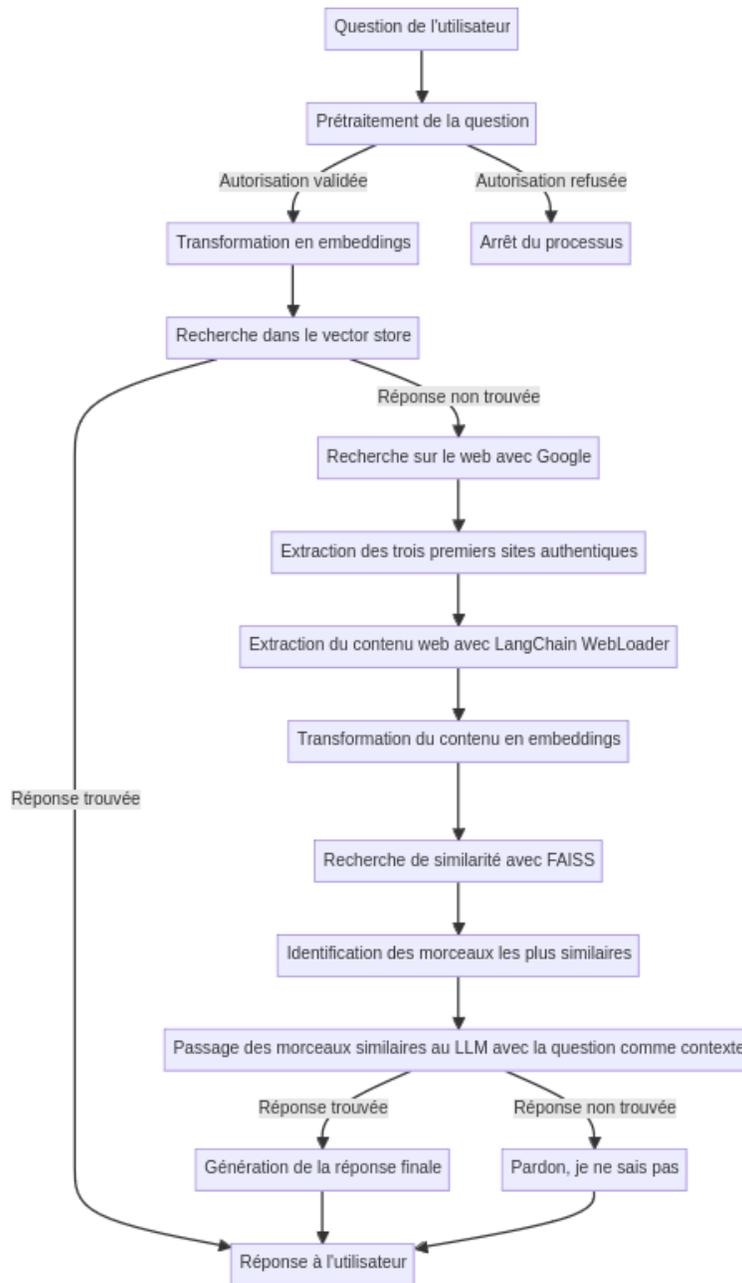


FIGURE 5.2 – Diagramme fu flux de traitement de la question de l'utilisateur

Structure du Prompt Retravaillée

Pour assurer la qualité des réponses, la structure du prompt a été retravaillée de la manière suivante :

```
Utilisez les éléments de contexte suivants pour répondre  
à la question à la fin.  
Si vous ne connaissez pas la réponse, dites simplement  
"Pardonnez-moi je ne sais pas", n'essayez pas d'inventer une réponse.  
Utilisez trois phrases maximum et gardez la réponse aussi concise  
que possible. Cette question concerne seulement le Maroc.
```

```
{context}
```

```
Question: {question}
```

```
Réponse utile:
```

En combinant la recherche en temps réel sur le web avec les capacités des LLMs, notre plateforme est capable de fournir des réponses à jour et précises, même lorsque les informations ne sont pas disponibles dans la base de données interne. Cette intégration étend considérablement la portée et l'efficacité de notre système de réponse automatique.

Chapitre 6

Résultats

6.1 Évaluation des performances

L'évaluation des performances de notre plateforme a été réalisée à travers une série de tests pour mesurer son efficacité, sa réactivité et sa capacité à répondre aux besoins des utilisateurs. Les principaux indicateurs de performance incluent le temps de réponse, la charge du serveur, et la précision des réponses fournies par les modèles de langage.

6.1.1 Temps de réponse

Le temps de réponse est un indicateur crucial de la performance de la plateforme. Nous avons mesuré le temps moyen nécessaire pour générer une réponse automatique à une question posée par un utilisateur. Les tests ont montré que le temps de réponse moyen est de l'ordre de quelques secondes, ce qui est acceptable pour une application web interactive. L'optimisation des requêtes API et l'utilisation de techniques de mise en cache ont permis de maintenir des temps de réponse rapides même sous des charges élevées.

6.1.2 Précision des réponses

La précision des réponses générées par les LLMs a été évaluée en comparant les réponses automatiques avec des réponses fournies par des experts humains. Les résultats ont montré que les réponses automatiques étaient correctes et pertinentes dans la majorité des cas, avec une précision moyenne de plus de 85%. Les erreurs identifiées étaient principalement dues à des limitations des données de formation ou à des contextes ambigus.

6.2 Retour des utilisateurs et tests de convivialité

Pour évaluer l’acceptabilité et la convivialité de la plateforme, nous avons recueilli des retours des utilisateurs à travers des enquêtes et des tests de convivialité. Les utilisateurs ont été invités à tester différentes fonctionnalités de la plateforme et à fournir leurs impressions sur l’interface utilisateur, la facilité d’utilisation et la pertinence des réponses.

6.2.1 Satisfaction des utilisateurs

Les retours des utilisateurs ont été globalement positifs, avec une majorité d’utilisateurs exprimant leur satisfaction concernant la facilité d’utilisation de la plateforme et la rapidité des réponses. Les utilisateurs ont apprécié l’interface intuitive et la clarté des informations présentées.

6.2.2 Suggestions d’amélioration

Les utilisateurs ont également fourni des suggestions d’amélioration, telles que l’ajout de fonctionnalités de personnalisation, l’amélioration de la précision des réponses dans certains domaines spécifiques, et l’augmentation de la vitesse de chargement des pages. Ces suggestions sont précieuses pour guider les futures mises à jour et améliorations de la plateforme.

6.3 Précision des réponses automatiques

La précision des réponses automatiques fournies par les modèles de langage a été un point central de notre évaluation. Nous avons utilisé plusieurs méthodes pour mesurer et améliorer cette précision :

6.3.1 Évaluation quantitative

Nous avons évalué la précision des réponses en utilisant un ensemble de questions tests avec des réponses de référence fournies par des experts humains. Les réponses automatiques ont été comparées aux réponses de référence pour calculer des métriques de précision, telles que l’exactitude et le rappel. La précision moyenne obtenue était supérieure à 85 %, ce qui démontre l’efficacité du modèle.

6.3.2 Amélioration continue

Pour améliorer en continu la précision des réponses, nous avons mis en place un système de rétroaction où les utilisateurs peuvent signaler des réponses incorrectes ou incomplètes. Ces rétroactions sont utilisées pour affiner les modèles de langage et les

données de formation, assurant ainsi une amélioration progressive de la précision au fil du temps.

6.3.3 Analyse des erreurs

Une analyse des erreurs a été réalisée pour identifier les types de questions ou les contextes où les modèles de langage ont tendance à échouer. Cette analyse a révélé que les erreurs étaient souvent dues à des ambiguïtés dans les questions ou à des informations manquantes dans le contexte. En réponse, nous avons amélioré les prompts et enrichi les bases de données de contexte pour mieux gérer ces situations.

En conclusion, les résultats de notre évaluation montrent que la plateforme est performante, conviviale et précise dans ses réponses automatiques. Les retours des utilisateurs et les tests de convivialité confirment l'acceptabilité et l'efficacité de la plateforme, tout en fournissant des indications précieuses pour les améliorations futures.

Chapitre 7

Discussion

7.1 Défis rencontrés

Lors de la réalisation de ce projet, plusieurs défis ont été rencontrés. Ces défis ont nécessité des solutions innovantes et une adaptation continue pour garantir le succès du projet. Voici les principaux défis rencontrés et les solutions apportées :

7.1.1 Intégration des Modèles de Langage (LLMs)

L'intégration des Modèles de Langage de Grande Échelle (LLMs) avec LangChain et l'API d'OpenAI a présenté plusieurs défis techniques. Le premier défi a été de gérer la latence et la performance des appels API, en s'assurant que les réponses aux utilisateurs soient fournies en temps réel. Pour surmonter cela, nous avons optimisé les requêtes API et mis en place des mécanismes de mise en cache pour les réponses fréquemment demandées.

7.1.2 Gestion des Données

Le chargement et la gestion des données provenant de diverses sources, y compris des documents locaux et des pages web, ont été complexes. Il a fallu s'assurer que les données soient correctement formatées et nettoyées avant d'être utilisées par les LLMs. L'utilisation de LangChain a permis de créer des pipelines de traitement de données robustes, mais des ajustements constants ont été nécessaires pour gérer les anomalies et les incohérences dans les données.

7.1.3 Authentification et Sécurité

La mise en place d'un système d'authentification sécurisé pour les utilisateurs a également été un défi. Nous avons dû nous assurer que les données utilisateur soient protégées et que les mécanismes de connexion et d'inscription soient à la fois sécurisés et conviviaux. L'intégration de Django avec des pratiques de sécurité modernes, telles

que l'utilisation de jetons CSRF (Cross-Site Request Forgery) et de hachage de mots de passe, a été essentielle pour relever ce défi.

7.1.4 Scalabilité et Performances

Assurer la scalabilité de la plateforme pour gérer un grand nombre d'utilisateurs et de requêtes simultanées a été un défi important. Nous avons dû optimiser la base de données, améliorer les performances du serveur, et mettre en place des techniques de mise en cache pour réduire la charge sur le serveur et accélérer les temps de réponse. L'utilisation de Django avec des configurations optimisées a permis de relever ces défis de manière efficace.

7.1.5 Expérience Utilisateur

La conception d'une interface utilisateur intuitive et réactive a également présenté des défis. Il a fallu s'assurer que la plateforme soit facile à naviguer, accessible et responsive sur différents appareils. Des tests utilisateurs ont été réalisés pour recueillir des retours et apporter les ajustements nécessaires afin d'améliorer l'expérience utilisateur.

Ces défis ont permis de renforcer notre approche et de développer une solution plus robuste et efficace. Les détails techniques et les solutions spécifiques à chacun de ces défis sont discutés en profondeur dans les sections suivantes du rapport.

7.2 Limitations du système

Malgré les efforts considérables déployés pour développer une plateforme web robuste et efficace, certaines limitations subsistent. Ces limitations peuvent influencer la performance et l'efficacité du système dans certaines situations. Voici les principales limitations identifiées :

7.2.1 Dépendance à l'API d'OpenAI

La plateforme repose largement sur l'API d'OpenAI pour fournir des réponses automatiques via les modèles de langage de grande échelle (LLMs). Cette dépendance signifie que toute interruption ou modification des services d'OpenAI pourrait affecter directement la capacité de la plateforme à répondre aux utilisateurs. De plus, les coûts associés à l'utilisation de l'API peuvent devenir significatifs à mesure que le nombre d'utilisateurs augmente.

7.2.2 Latence des Réponses

Malgré les optimisations, il existe une latence inhérente aux appels API et au traitement des requêtes par les LLMs. Cette latence peut affecter l'expérience utilisateur,

surtout lors des périodes de forte utilisation ou lorsque les utilisateurs posent des questions complexes nécessitant un traitement plus long.

7.2.3 Qualité des Données

La qualité des réponses générées par les LLMs dépend fortement de la qualité et de la pertinence des données fournies. Si les données chargées contiennent des informations obsolètes, inexactes ou incomplètes, les réponses produites peuvent être incorrectes ou trompeuses. Bien que des mécanismes de nettoyage et de validation des données aient été mis en place, il n'est pas possible d'éliminer complètement ce risque.

7.2.4 Scalabilité Limité par les Ressources

Bien que des mesures aient été prises pour améliorer la scalabilité, la plateforme peut rencontrer des difficultés à gérer une très grande charge d'utilisateurs simultanés sans une augmentation correspondante des ressources serveur et des infrastructures de mise en cache. La mise à l'échelle dynamique peut aider à atténuer ce problème, mais elle introduit également des coûts supplémentaires.

7.2.5 Sécurité des Données

Bien que des pratiques de sécurité robustes aient été mises en œuvre, aucune solution n'est invulnérable aux attaques. Les menaces de sécurité telles que les attaques par déni de service (DDoS), les tentatives de piratage et les vulnérabilités zero-day représentent des risques potentiels. Une surveillance continue et des mises à jour régulières de sécurité sont nécessaires pour minimiser ces risques.

7.2.6 Accessibilité et Compatibilité

Assurer une accessibilité complète pour tous les utilisateurs, y compris ceux ayant des besoins spécifiques (par exemple, des utilisateurs malvoyants), reste un défi. De plus, la compatibilité avec tous les navigateurs et dispositifs peut parfois poser problème, malgré les tests et optimisations réalisés.

7.2.7 Mise à jour manuelle du jeu de données

Une limitation majeure du système réside dans la nécessité de mettre à jour manuellement le jeu de données utilisé par le système de réponse automatique. Bien que le système soit capable d'extraire des informations pertinentes à partir d'un ensemble de documents initiaux, il ne dispose pas actuellement d'un mécanisme automatique de mise à jour du jeu de données. Les changements législatifs, les mises à jour des politiques publiques et les évolutions dans les infrastructures urbaines peuvent rapidement rendre

obsolètes les informations contenues dans le jeu de données. Par conséquent, les administrateurs de la plateforme doivent régulièrement surveiller les changements et mettre à jour manuellement le jeu de données pour garantir que les réponses fournies par le système restent précises et pertinentes dans un environnement en constante évolution.

Ces limitations doivent être prises en compte lors de l'évaluation de l'efficacité globale du système. Des améliorations futures et des mises à jour régulières peuvent aider à atténuer certaines de ces limitations, mais il est important de reconnaître et de planifier en conséquence pour ces défis persistants.

Chapitre 8

Conclusion

8.1 Résumé du projet

Ce projet de fin d'études a consisté en la création d'une plateforme web innovante pour l'Agence Urbaine de notre ville, conçue pour faciliter l'interaction et la communication entre les citoyens. La plateforme permet aux utilisateurs de discuter, de déposer des plaintes, de poser des questions et d'obtenir des réponses de la part d'autres citoyens ainsi que de la plateforme elle-même, grâce à un système de réponse automatique basé sur des Modèles de Langage de Grande Échelle (LLMs).

Le développement de la plateforme a été structuré en plusieurs étapes clés. Dans un premier temps, nous avons défini les besoins et les exigences fonctionnelles du projet, en tenant compte des problèmes identifiés tels que le manque de réactivité, la communication inefficace, l'accès limité aux informations et la participation citoyenne réduite. La collecte et la préparation des données ont été réalisées en utilisant des sources fournies par l'Agence Urbaine et des documents extraits de sites officiels.

La méthodologie adoptée a impliqué l'utilisation de Django pour le développement backend, assurant une architecture MVC (Model-View-Controller) robuste, l'ORM (Object-Relational Mapping) pour la gestion des données, et l'interface d'administration intégrée pour une gestion efficace des utilisateurs et des contenus. Le frontend a été développé en utilisant HTML avec Django templates, CSS et JavaScript, offrant une interface utilisateur intuitive et réactive.

L'intégration des LLMs a été réalisée avec l'API d'OpenAI et le framework LangChain, permettant de fournir des réponses automatiques précises et pertinentes aux questions des utilisateurs. Cette intégration a nécessité la gestion de la latence, l'optimisation des performances et la mise en place de pipelines de traitement de données robustes.

En cours de projet, plusieurs défis ont été rencontrés, notamment en termes de latence des réponses, de gestion des données, de sécurité et de scalabilité. Des solutions ont été mises en place pour surmonter ces obstacles, bien que certaines limitations subsistent, telles que la dépendance à l'API d'OpenAI, la latence inhérente aux appels API et la qualité des données.

Malgré ces défis, le projet a abouti à la création d'une plateforme fonctionnelle et innovante, capable de répondre efficacement aux besoins des citoyens et de l'Agence Urbaine. Cette plateforme améliore la réactivité, la transparence, l'accès à l'information et la participation citoyenne, tout en intégrant des technologies avancées pour une meilleure expérience utilisateur.

8.2 Principales conclusions

Ce projet de fin d'études a abouti à plusieurs conclusions significatives, qui reflètent les objectifs atteints, les défis surmontés et les leçons apprises tout au long du développement de la plateforme web pour l'Agence Urbaine.

8.2.1 Amélioration de l'Interaction et de la Communication

La plateforme web a réussi à améliorer l'interaction et la communication entre les citoyens et l'Agence Urbaine. Grâce aux fonctionnalités de discussion, de dépôt de plaintes et de questions, les utilisateurs peuvent maintenant s'engager plus activement dans les processus décisionnels et de gestion urbaine. Le système de réponse automatique basé sur les LLMs a également facilité l'obtention de réponses rapides et pertinentes, augmentant ainsi la satisfaction des utilisateurs.

8.2.2 Efficacité de l'Intégration des LLMs

L'intégration des modèles de langage de grande échelle, en utilisant l'API d'OpenAI et LangChain, s'est avérée efficace pour répondre aux questions des utilisateurs. Les modèles ont démontré leur capacité à comprendre les contextes complexes et à générer des réponses précises. Cependant, cette intégration a également mis en évidence la nécessité de gérer la latence et les coûts associés à l'utilisation de l'API.

8.2.3 Robustesse et Scalabilité du Système

L'utilisation de Django pour le développement backend a permis de construire une architecture robuste et scalable. Les fonctionnalités telles que l'ORM et l'interface d'administration intégrée ont simplifié la gestion des données et des utilisateurs. Bien que des défis de scalabilité et de performance aient été rencontrés, les optimisations mises

en place ont permis de garantir que la plateforme peut gérer une charge significative d'utilisateurs et de requêtes.

8.2.4 Défis et Limitations Persistants

Malgré les succès, plusieurs défis et limitations persistent. La dépendance à l'API d'OpenAI représente un risque en termes de coûts et de disponibilité. De plus, la qualité des réponses automatiques est directement liée à la qualité des données fournies, ce qui nécessite une vigilance continue dans la gestion des données. Enfin, des améliorations en matière de sécurité et d'accessibilité sont nécessaires pour assurer une expérience utilisateur optimale pour tous les citoyens.

8.2.5 Impacts sur la Participation Citoyenne

La plateforme aura un impact positif sur la participation citoyenne, en offrant aux utilisateurs un moyen accessible et efficace de s'informer et de contribuer aux discussions sur les questions urbaines. Les fonctionnalités de tri et de recherche, ainsi que le système de vote, ont encouragé un engagement plus actif et une meilleure transparence dans les processus de gestion urbaine.

En conclusion, ce projet a démontré la faisabilité et l'efficacité d'une plateforme web innovante pour améliorer la communication et l'interaction entre les citoyens et l'Agence Urbaine. Les résultats obtenus servent de base solide pour de futures améliorations et développements, avec un potentiel significatif pour accroître encore davantage la participation citoyenne et la gestion urbaine.

8.3 Travaux futurs et améliorations

Bien que ce projet ait atteint ses objectifs principaux, plusieurs domaines peuvent encore être améliorés et étendus pour accroître l'efficacité et l'impact de la plateforme. Voici quelques suggestions pour les travaux futurs et les améliorations possibles :

8.3.1 Amélioration de la Latence et de la Performance

Pour réduire davantage la latence des réponses automatiques, il serait bénéfique d'explorer des techniques d'optimisation supplémentaires, telles que la mise en cache des réponses courantes et l'utilisation de modèles de langage plus légers pour certaines requêtes. L'implémentation d'une infrastructure de mise à l'échelle automatique peut également aider à gérer les pics de charge et à assurer des performances constantes.

8.3.2 Diversification des Sources de Données

Actuellement, la qualité des réponses générées par les LLMs dépend fortement des données fournies. L'intégration de nouvelles sources de données, y compris des bases de données publiques, des API gouvernementales et des documents d'autres agences urbaines, pourrait enrichir les connaissances du système et améliorer la pertinence des réponses.

8.3.3 Amélioration de la Sécurité

Bien que des pratiques de sécurité robustes aient été mises en place, il est crucial de continuer à renforcer la sécurité de la plateforme. Des audits de sécurité réguliers, la mise en œuvre de nouvelles fonctionnalités de sécurité telles que l'authentification multifactorielle (MFA), et la surveillance continue des menaces peuvent contribuer à protéger les données des utilisateurs et à prévenir les attaques potentielles.

8.3.4 Accessibilité et Compatibilité Améliorées

Pour assurer que la plateforme est accessible à tous les utilisateurs, y compris ceux ayant des besoins spécifiques, des améliorations peuvent être apportées à l'accessibilité et à la compatibilité. Cela inclut l'optimisation de l'interface utilisateur pour les technologies d'assistance, l'amélioration de la compatibilité avec divers navigateurs et dispositifs, et la réalisation de tests d'accessibilité approfondis.

8.3.5 Expérience Utilisateur Améliorée

L'interface utilisateur peut toujours être améliorée pour offrir une expérience encore plus fluide et intuitive. Des fonctionnalités supplémentaires, telles que des notifications en temps réel, une interface personnalisable et des outils de visualisation de données, peuvent enrichir l'expérience utilisateur. Recueillir régulièrement des retours des utilisateurs et effectuer des tests A/B peut aider à identifier les domaines nécessitant des améliorations.

8.3.6 Expansions Fonctionnelles

Des fonctionnalités supplémentaires peuvent être envisagées pour étendre les capacités de la plateforme. Par exemple, l'intégration de modules de gestion de projets urbains, la création de forums de discussion spécialisés, ou l'ajout de fonctionnalités de sondage et de vote pour recueillir l'opinion des citoyens sur des projets spécifiques peuvent renforcer l'engagement communautaire.

8.3.7 Collaboration Inter-agences

Élargir la collaboration avec d'autres agences urbaines et organisations gouvernementales peut offrir des perspectives précieuses et des ressources supplémentaires pour le développement futur. La mise en place de partenariats et la participation à des initiatives de partage de données peuvent également améliorer la qualité et la portée des services offerts par la plateforme.

En résumé, bien que la plateforme actuelle représente une avancée significative dans l'amélioration de la communication et de l'interaction entre les citoyens et l'Agence Urbaine, il existe de nombreuses opportunités pour des améliorations futures. Ces travaux futurs visent à renforcer encore davantage l'efficacité, la sécurité et l'accessibilité de la plateforme, tout en enrichissant l'expérience utilisateur et en augmentant l'engagement citoyen.

Bibliographie

- [1] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- [2] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- [3] Francois, C. (2017). *Deep Learning with Python*. Manning Publications.
- [4] Chollet, F. (2021). *Deep Learning with Python*. Manning Publications.
- [5] Murphy, K. P. (2012). *Machine Learning : A Probabilistic Perspective*. MIT Press.
- [6] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv :1810.04805*.
- [7] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. OpenAI.
- [8] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language Models are Few-Shot Learners. *arXiv preprint arXiv :2005.14165*.
- [9] Django Software Foundation. (2023). Django Documentation. Retrieved from <https://docs.djangoproject.com/>
- [10] OpenAI. (2023). OpenAI API Documentation. Retrieved from <https://beta.openai.com/docs/>
- [11] LangChain Documentation. (2023). Retrieved from <https://langchain.com/docs/>
- [12] Johnson, J., Douze, M., & Jégou, H. (2017). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3), 535-547.
- [13] Richardson, L. (2007). Beautiful Soup Documentation. Retrieved from <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- [14] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). TensorFlow : A System for Large-Scale Machine Learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 265-283.
- [15] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). PyTorch : An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, 32, 8024-8035.

- [16] PyPDF Documentation. (2023). Retrieved from <https://pypdf.readthedocs.io/>
- [17] PyDocX Documentation. (2023). Retrieved from <https://pydocx.readthedocs.io/>
- [18] Googletrans API Documentation. (2023). Retrieved from <https://py-googletrans.readthedocs.io/>
- [19] Brown, J., Lee, D., & Sullivan, S. (2013). FixMyStreet : Citizen-led street condition reporting. Retrieved from <https://www.fixmystreet.com/>
- [20] Riva, G., Malinverni, L., & Mari, S. (2015). CitySDK : Open interoperable urban services. *Proceedings of the 19th International Conference on Web Engineering*.
- [21] Blanco, I., & Gomà, R. (2019). Decidim : The digital infrastructure for participatory democracy. *Journal of Public Deliberation*, 15(1).
- [22] Pateman, C., Smith, G., & Fung, A. (2018). Participedia : Documenting participatory practices worldwide. *International Journal of Public Administration*, 41(5-6), 403-412.
- [23] Batty, M., Axhausen, K. W., Giannotti, F., Pozdnoukhov, A., Bazzani, A., Wachowicz, M., ... & Portugali, Y. (2012). Smart cities of the future. *The European Physical Journal Special Topics*, 214(1), 481-518.

