

# Dixit : Agent intelligent d'accès à la recherche

Combler le fossé entre les chercheurs et le savoir scientifique grâce à l'IA conversationnelle et l'intégration de données ouvertes.



# La Vision Derrière Dixit

Dixit est un agent conversationnel intelligent conçu pour révolutionner la manière dont les étudiants et les chercheurs accèdent aux travaux universitaires. En exploitant les dépôts en libre accès et le traitement avancé du langage naturel, Dixit simplifie la découverte de thèses, de dissertations et d'articles scientifiques.

La plateforme transforme les requêtes de recherche complexes en résultats exploitables, éliminant ainsi les barrières traditionnelles à la découverte de connaissances académiques.

## Principales Fonctionnalités

- Extraction en temps réel des dépôts ouverts
- Compréhension intelligente des requêtes
- Réponses structurées et citées
- Traitement LLM local pour la confidentialité





# Rencontrez l'équipe de développement



## Amine

### Lead Agent / Architecture

Responsable du cerveau de l'agent, Amine définit la structure et la logique fondamentale de Dixit.

- Définir le scope fonctionnel
- Concevoir le schéma d'architecture
- Définir le flux agentique
- Décider quand la base de connaissances suffit ou qu'un outil est nécessaire
- Centraliser les décisions techniques



## Imène

### Backend Orchestrateur

Le moteur technique, Imène assure la fluidité des interactions et la gestion des processus en arrière-plan.

- Créer l'API /chat
- Implémenter la logique décisionnelle
- Gérer l'historique conversationnel
- Charger la base de connaissances (KB)
- Appeler MCP / Ollama
- Gérer les erreurs et les logs



## Marielle

### Knowledge Base & Logique Métier

La mémoire intelligente, Marielle structure et enrichit la base de connaissances de Dixit, assurant la pertinence des réponses. Très valorisé car souvent négligé.

- Concevoir la structure de la KB
- Rédiger les contenus métier
- Définir ce que l'agent sait et ce qui déclenche un outil
- Tester les réponses sans scraping



## Aude

### MCP & Tools / Scraping

Les mains de l'agent, Aude développe les outils et les capacités de scraping, garantissant un accès éthique et structuré aux données. Qualité > quantité.

- Implémenter le serveur MCP
- Créer 1-2 outils maximum
- Réaliser le scraping éthique des données
- Nettoyer et structurer les données extraites
- Gérer les erreurs d'outils



## Laetitia

### Frontend & UX

La vitrine du projet, Laetitia conçoit une interface utilisateur intuitive et esthétique pour une expérience fluide.

- Développer l'interface du chatbot
- Assurer une UX fluide
- Gérer les états de chargement et d'erreur
- Assurer la communication API
- Mettre en page de manière propre et cohérente

# Architecture Technique



## Couche Frontend

Interface basée sur JavaScript offrant un accès réactif et convivial aux requêtes de recherche et à l'affichage des résultats.



## Orchestrateur Backend

Serveur Python gérant le routage des requêtes, les interrogations de la base de connaissances et la coordination entre les composants du système.



## Moteur IA

Framework Ollama hébergeant des modèles de langage locaux (Qwen2.5 7B / Mistral 8B) pour un raisonnement intelligent et la génération de réponses.



## Intégration MCP

Protocole de Contexte de Modèle (MCP) permettant une intégration modulaire des outils, séparant les processus de prise de décision de ceux d'exécution.

# Architecture du Flux de Travail du Système



## Soumission de la Requête

L'étudiant soumet sa question de recherche via l'interface



## Analyse

Le backend interroge la base de connaissances interne pour des informations pertinentes



## Récupération des Données

Le serveur MCP active les outils de scraping pour récupérer les données de thèse en temps réel



## Traitement

Le LLM local analyse les données récupérées et génère une réponse structurée



## Livraison de la Réponse

L'utilisateur reçoit une réponse citée et organisée avec des références de source



# Stratégie de Scraping & Gestion des Données



## Données Réelles, Pas de Maquettes

Dixit extrait des données de recherche authentiques des dépôts en libre accès, garantissant aux utilisateurs un contenu académique actuel et vérifiable.

## Principes de Scraping Éthique

- Respect des directives robots.txt
- Concentration sur les sources publiques et sans authentification
- Extraction ciblée depuis arXiv et HAL
- Conformité aux conditions d'utilisation des dépôts

## Organisation du Lac de Données

Système de stockage structuré : données brutes, ensembles de données traités et métadonnées complètes pour une récupération efficace.

# Stratégie de Sélection du Modèle Linguistique

Modèle	Paramètres	Cas d'Utilisation	Performance
Qwen2.5 0.5B	500M	Développement & Test	Cycles d'itération rapides
Qwen2.5 7B	7B	Production (Option A)	Raisonnement et vitesse équilibrés
Minstral 8B	8B	Production (Option B)	Contexte académique amélioré

La phase de développement utilise Qwen2.5 0.5B pour le prototypage rapide, tandis que le déploiement en production cible des modèles plus grands (7B-8B paramètres) pour offrir des capacités de raisonnement sophistiquées essentielles aux requêtes de recherche académique.

# Intelligence Modulaire : L'Avantage MCP

## Séparation des préoccupations

Le Protocole de Contexte de Modèle (MCP) découple la prise de décision de l'exécution. L'IA détermine *quoi* rechercher, tandis que des scripts spécialisés gèrent *comment* le récupérer.

## Extensibilité

De nouvelles sources de données et outils peuvent être intégrés sans modifier la logique principale de l'IA, permettant une expansion rapide des fonctionnalités.

## Fiabilité

Les composants isolés réduisent les défaillances à l'échelle du système. Les problèmes de scraping n'affectent pas le moteur de raisonnement, et vice versa.



# Sprint de Sept Jours : Compétences et Réalisations

## Jours 1-2 : Fondations

Conception de l'architecture, sélection de la pile technologique et attribution des rôles d'équipe.  
Configuration initiale du dépôt et de l'environnement de développement.

## Jours 5-6 : Intégration et Tests

Configuration du serveur MCP, tests de bout en bout du flux de travail et validation du pipeline de données.  
Optimisation des performances et gestion des erreurs.

1

2

3

4

## Jours 3-4 : Développement Principal

Implémentation de l'orchestrateur backend, développement du module de scraping et création de l'interface frontend. Intégration d'Ollama avec des modèles de test.

## Jour 7 : Affinage

Tests finaux, achèvement de la documentation et préparation de la présentation. Planification du déploiement et définition de la feuille de route future.

# Perspectives Futures & Conclusions

## Améliorations Prévues

- Extension à d'autres référentiels académiques (PubMed, IEEE Xplore)
- Support multilingue pour la recherche internationale
- Visualisation avancée du réseau de citations
- Moteur de recommandation de recherche collaborative
- Fonctionnalité d'exportation pour les gestionnaires de références

## Points Clés à Retenir

Dixit démontre l'application pratique de l'IA agentique dans les contextes académiques, combinant les capacités modernes des LLM avec des pratiques éthiques de données pour démocratiser l'accès à la recherche.



- ❏ **Impact du projet :** Livraison réussie d'un agent intelligent fonctionnel en sept jours, démontrant des capacités de prototypage rapide et l'excellence du développement collaboratif.