

RENDU INTERMÉDIAIRE DU PROJET CHATBOT

Dossier Intermédiaire – Assistant Éducatif pour Aide aux Devoirs

1. Introduction

Contexte et Problématique de l'Incubateur

Dans un monde où l'éducation joue un rôle crucial, l'accès à des ressources pédagogiques de qualité est un enjeu majeur.

De nombreux élèves peinent à obtenir l'aide nécessaire pour comprendre leurs cours ou s'entraîner efficacement. Cette situation contribue également à des problématiques telles que le décrochage scolaire (source), le manque de motivation ou les inégalités d'accès aux ressources qui aggravent ces difficultés d'apprentissage.

Cette difficulté est accentuée par des enseignants souvent surchargés ne pouvant consacrer suffisamment de temps à chaque étudiant.

Malgré la diversité des solutions disponibles, beaucoup manquent d'adaptabilité et d'interactivité, laissant ainsi les besoins individuels des élèves partiellement répondus.

Il devient donc indispensable de repenser les approches traditionnelles et d'intégrer des outils innovants pour accompagner chaque élève dans son apprentissage.

Solution Proposée et Bénéfices Clés

Pour répondre à cette problématique, une plateforme éducative intégrant un assistant intelligent est proposée. Basé sur une intelligence artificielle avancée, cet assistant vise à aider les élèves à comprendre leurs cours et à s'entraîner à travers des exercices interactifs et personnalisés.

En s'appuyant sur des outils comme l'API OpenAI et les frameworks modernes tels que LangChain, cette solution exploite pleinement le potentiel des technologies avancées pour rendre l'apprentissage plus interactif et accessible à un large public.

De cette manière, l'apprentissage devient non seulement plus engageant, mais également plus efficace. En parallèle, cette plateforme allège la charge des enseignants, en leur fournissant des outils conçus pour maximiser l'impact pédagogique.

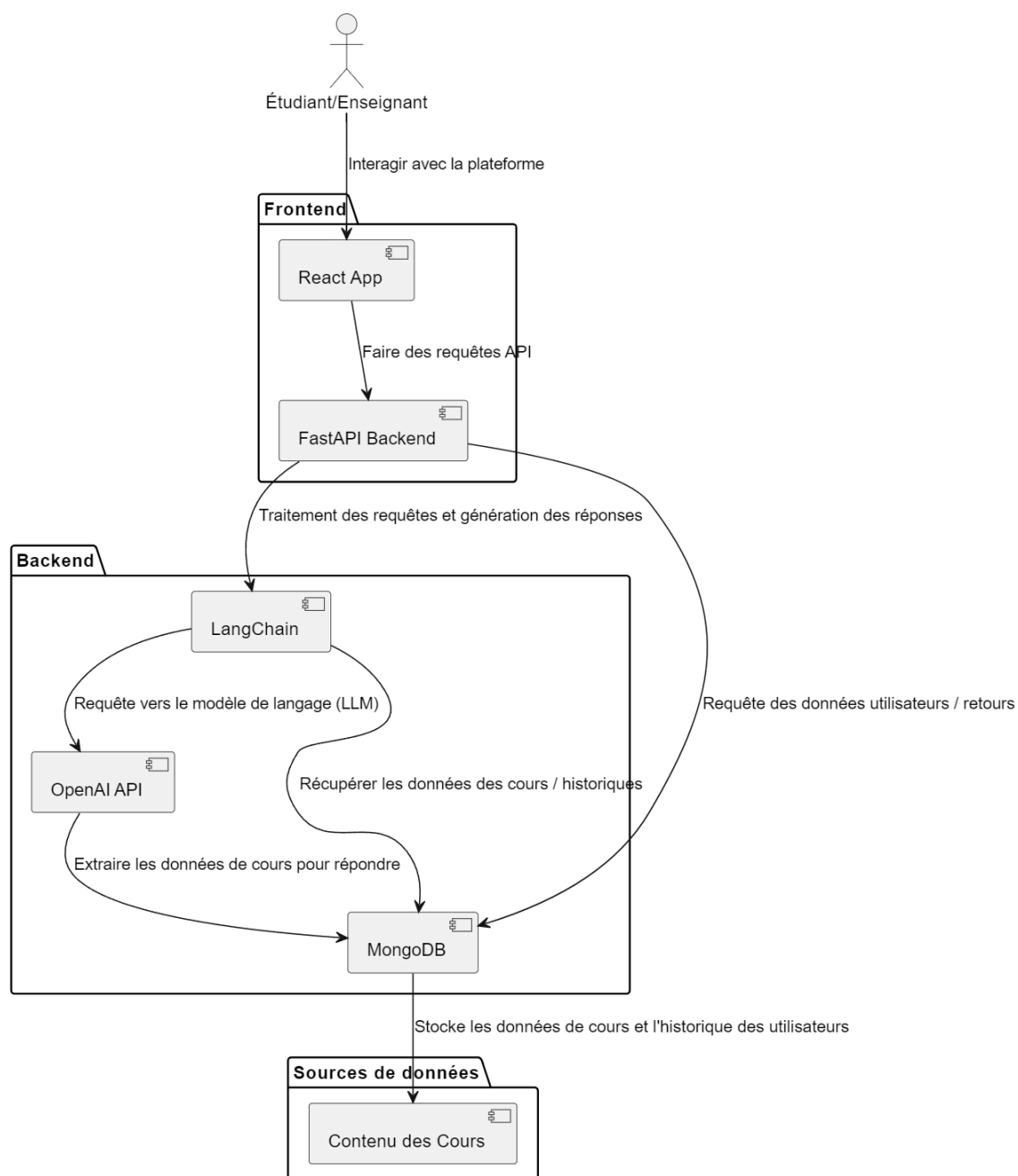
Objectifs Principaux du Chatbot

1. **Aide à la Compréhension** : Fournir des explications claires et adaptées aux élèves pour les aider à assimiler leurs cours.
2. **Entraînement personnalisé** : Générer des exercices alignés aux besoins de chaque élève, en tenant compte de son niveau et de ses préférences.
3. **Accessibilité 24/7** : Offrir une aide continue et intuitive, disponible à tout moment.

2. Documentation Technique et Vision

Architecture Technique

Schéma d'Architecture :



Choix Technologiques :

- **LangChain** : Intégration et gestion des prompts pour un dialogue adaptatif.
- **FastAPI** : Framework léger et rapide pour développer les APIs backend.
- **API OpenAI** : Moteur IA performant pour générer des réponses.
- **Swagger** : Documentation interactive pour simplifier les tests et les déploiements.
- **React** pour frontend
- **MongoDB** pour BDD

Justification des choix Technologiques :

- **React** : Choisi pour créer des interfaces utilisateur dynamiques et interactives, permettant une expérience fluide et réactive sans rechargement de la page, avec une forte réutilisabilité des composants et un large écosystème de bibliothèques.
- **MongoDB** : Une base de données NoSQL flexible et scalable. Elle permet de stocker des données non structurées comme les cours, exercices et historiques des utilisateurs, et pour intégrer la recherche vectorielle des documents.
- **OpenAI** : Utilisé pour sa capacité à générer des réponses naturelles, cohérentes et informatives en langage naturel. Elle semble être la plus performante vis à vis de nos besoins dans le cadre de ce projet.
- **LangChain** : Permet l'intégration facile des modèles OpenAI dans un flux de travail de recherche augmentée par génération (RAG), facilitant la récupération d'informations contextuelles et la génération de réponses personnalisées.
- **FastAPI** : Un framework Python performant, permettant de construire des API rapides et asynchrones avec une gestion automatique de la validation, de la documentation interactive et de la sécurité, idéal pour un backend scalable et réactif.

Besoins

Utilisateurs Cibles :

- **Élèves du collège au lycée** : Cherchant un support scolaire interactif et personnalisé.
- **Parents ou tuteurs** : Souhaitant une solution fiable pour améliorer l'autonomie de leurs enfants.

Fonctionnalités Essentielles :

1. **Réponse aux questions** : Comprendre et répondre aux questions liées aux cours.
2. **Génération d'exercices** : Proposer des exercices , incluant des quiz, des problèmes et questions, avec un feedback immédiat pour guider les étudiants.
3. **Historique et suivi** : Sauvegarde des sessions d'apprentissage pour permettre une révision ultérieure et identifier les progrès réalisés.

Planning (Roadmap sur 4 semaines)

1. Semaine 1 :

- Mise en place de l'environnement de développement ;
- Intégration initiale de l'API OpenAI via LangChain ;
- Intégration de la BDD MongoDB ;
- Développement des endpoints FastAPI ;

2. Semaine 2 :

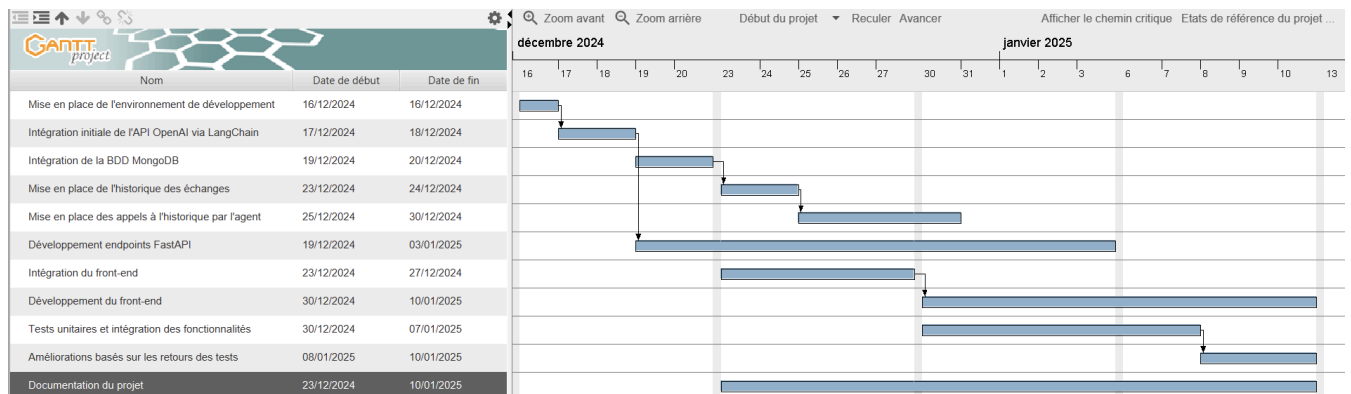
- Développement des endpoints FastAPI ;
- Intégration du front end React ;
- Mise en place de l'historique des échanges ;
- Mise en place des appels à l'historique par l'agent ;
- Documentation du projet ;

3. Semaine 3 :

- Développement des endpoints FastAPI ;
- Mise en place des appels à l'historique par l'agent ;
- Développement du front-end ;
- Tests unitaires et intégration des fonctionnalités ;
- Documentation du projet ;

4. Semaine 4 :

- Développement du front-end ;
- Tests unitaires et intégration des fonctionnalités ;
- Améliorations basées sur les retours des tests ;
- Documentation du projet ;



Risques Principaux :

- Gestion des coûts liés aux appels API OpenAI.
- Hallucinations du modèle si il n'a pas les réponses
- Limitations techniques des modèles IA dans des contextes complexes (exercices de mathématiques par exemple).

3. Prototype

API : Endpoints Principaux (Pour le moment)

1. **/ask** : Endpoint pour poser une question à l'assistant.
 - **Entrée** : Question en texte libre.
 - **Sortie** : Réponse générée par l'IA.
2. **/generate-exercice** : Endpoint pour demander un exercice.
 - **Entrée** : Sujet ou chapitre.
 - **Sortie** : Exercice + correction.
3. **/history** : Endpoint pour récupérer l'historique.
 - **Entrée** : Identifiant utilisateur.
 - **Sortie** : Historique des interactions.

Code : Structure Projet

```
|—api/          # Gestion des routes et endpoints de l'API
| |—endpoints/  # Endpoints spécifiques par fonctionnalité
| |   |—chat.py  # Endpoint pour les fonctionnalités de chat
| |   |—router.py # Router principal regroupant tous les endpoints
|—core/         # Configuration et éléments centraux de l'application
|—models/       # Modèles de données Pydantic
|   |—chat.py   # Modèles pour les requêtes/réponses de chat
|—services/     # Services métier
|   |—llm_service.py # Service d'interaction avec le LLM
|—tests/        # Test unitaire sur le code
|   |—test_mongo_service.py # test lié à la base de donnée
|—utils/        # Utilitaires et helpers
|—main.py       # Point d'entrée de l'application
```

Liens :

- **GitHub (partie LangChain)** : https://github.com/Xolitor/Agent_conv_edu.git
- **Github (partie React)**: <https://github.com/Xolitor/chatbot-frontend.git>

4. Vision Future & Conclusion

Améliorations Prévues

1. **Amélioration de la Data Quality** : Pour fournir des réponses plus pertinentes on souhaite ajouter des documents sur lesquels il pourra chercher les informations pertinentes. Les réponses seront plus précises et adaptées aux questions des utilisateurs.
2. **Personnalisation des réponses** : Fournir des réponses en utilisant des références culturelles pertinentes en fonction de leur centre d'intérêt (comme des exemples issus de jeux vidéos, par rapport au sport ou de la pop culture) pour rendre les concepts étudiés plus accessibles.

Perspectives d'Évolution

- **Expansion Internationale** : Adaptation à différents systèmes éducatifs et langues.
- **Proposer l'outil aux établissements éducatifs** : Faire adopter aux différents établissements comme outil complémentaire aux professeurs pour suivre la progression de leurs élèves et identifier les points à travailler avec leurs élèves.
- **Gamification** : Ajout de mécanismes ludiques pour augmenter l'engagement.
- **Suivi de progression intelligent** : Fournir des rapports personnalisés sur les performances et les lacunes, à destination des élèves, parents et enseignants.
- **Planification flexible** : Proposer des emplois du temps sur demande et adaptés aux élèves pour s'exercer et réviser des notions de cours.

Prochaines Étapes

1. Finaliser la liste des endpoints
2. Mettre en place des databases de cours pour que l'agent puisse chercher les informations
3. Suite à la mise en place de l'interface front end, il faut désormais la faire évoluer: personnaliser, développer et rendre plus attractif.
4. Tests utilisateurs étendus avec élèves.