Rapport de projet: Création d'un Chatbot pour la HES-SO

1. Introduction de la problématique du projet

La HES-SO, avec plus de 10 000 enseignant·es, fait face à une demande croissante de formation et de gestion administrative via son centre de développement professionnel, DevPro. DevPro offre une variété de cours et assure la gestion des attestations didactiques en collaboration avec les différentes directions. L'afflux constant de questions administratives et de demandes d'information pose un défi pour l'administration, qui cherche à soulager cette pression.

La problématique centrale est donc la suivante : comment améliorer l'expérience utilisateur sur le site DevPro tout en réduisant la charge administrative liée aux demandes d'information récurrentes ? La solution identifiée est la création d'un chatbot basé sur l'intelligence artificielle (IA), capable de répondre aux questions fréquentes des utilisateur·rices et de pointer vers les ressources appropriées.

2. Solution et technologies choisies selon les critères de la HES

Critères principaux :

- **Technologies open-source** : Permet de réduire les coûts et de garantir la flexibilité et la transparence du code.
- **Solution self-hosted** : Important pour des raisons de contrôle des données et de respect de la vie privée.
- **IA customizable** : L'outil doit permettre des ajustements spécifiques à l'institution, tels que l'entraînement sur des données propres à la HES-SO.
- **Utilisation de technologies d'Embeddings/RAG (Retrieval-Augmented Generation)** : Essentiel pour que le chatbot puisse répondre aux questions en recherchant dans une base de connaissances existante et en générant des réponses précises à partir des documents fournis.

Solutions considérées :

- **BotPress** : Une solution open-source avec des fonctionnalités IA. Self-hosted, mais limitée en personnalisation IA avancée.
- **Rasa** : Une plateforme open-source pour construire des chatbots, mais nécessite une configuration complexe pour le traitement du langage naturel (NLP) avancé.
- **MindsDB** : Technologie émergente qui facilite la construction de modèles IA en temps réel. Permet l'utilisation d'Embeddings et d'un modèle RAG, ce qui la rend idéale pour une intégration avec les bases de données existantes de la HES.

Choix final : MindsDB pour le backend d'IA, combiné à Next.js pour la gestion du frontend du site web.

3. Choix de la technologie MindsDB en backend et Next.js en frontend

3.1 MindsDB : Pourquoi cette solution et comment cela fonctionne

3.1.1 Qu'est-ce que MindsDB ?

MindsDB est une plateforme d'IA qui permet de créer, déployer, et ajuster des modèles IA en temps réel en se connectant à des sources de données variées comme des bases de données, des vector stores, ou des applications. Elle propose un ensemble d'outils simples, compatibles avec des technologies standards comme SQL, pour faciliter l'intégration de l'IA dans les applications existantes.

3.1.2 Pourquoi MindsDB ?

MindsDB se distingue par sa capacité à créer des agents d'IA personnalisables, en intégrant des compétences comme le text-to-SQL, la recherche sémantique ou les modèles de langage naturel (LLM). Elle permet d'utiliser des technologies comme les embeddings pour améliorer les réponses du chatbot en combinant génération de texte et recherche dans des documents préexistants, une approche adaptée au besoin de DevPro d'analyser des documents pédagogiques et administratifs.

3.1.3 Licence de MindsDB

MindsDB utilise deux types de licences :

- **Elastic License 2.0** : Cette licence permet une utilisation gratuite de la plateforme à condition que le logiciel ne soit pas fourni sous forme de service hébergé. Cela correspond parfaitement au projet, car la solution sera auto-hébergée sur les serveurs de la HES-SO.
- **MIT License** : Utilisée pour certaines intégrations, cette licence libre permet une large utilisation, modification, et distribution du logiciel, compatible avec les exigences open-source du projet.

3.2 Next.js : Qu'est-ce que c'est et pourquoi cette solution ?

Next.js est un framework pour React qui permet de créer des applications web modernes avec un rendu côté serveur (SSR), idéal pour les performances et le SEO. Il offre une grande flexibilité, notamment en ce qui concerne l'intégration avec des API, la gestion des données en temps réel, et l'optimisation des interfaces utilisateurs. Il est également open-source, et peut facilement s'intégrer à MindsDB via des requêtes API pour afficher les réponses du chatbot en direct.

Pourquoi Next.js ?

- **Rendu côté serveur** : Améliore les performances, essentiel pour une navigation fluide sur le site DevPro.
- **Facilité d'intégration** : Compatible avec les API de MindsDB et permet d'ajuster rapidement l'interface en fonction des retours utilisateur·rices.
- **Modularité** : Le framework permet de gérer différents composants du site web tout en assurant une expérience utilisateur optimale.

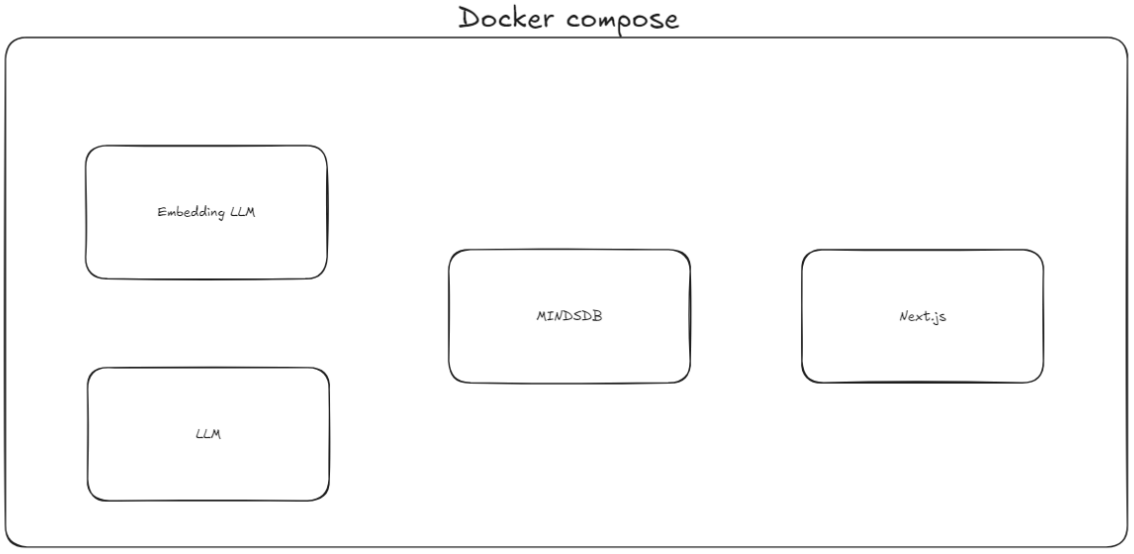
4. System Design : Backend + Frontend + RAG/Embeddings

Architecture du système

Le système proposé repose sur une architecture à deux niveaux :

1. **Frontend (Next.js)** : Gère l'interface utilisateur, notamment le formulaire de questions pour le chatbot. Il est responsable de l'affichage des réponses du chatbot et de la navigation fluide entre les différentes pages liées.

2. **Backend (MindsDB)** : C'est ici que se situent les principaux traitements d'IA. Le backend intègre un modèle RAG qui permet d'extraire des informations pertinentes à partir des documents de la HES-SO (comme les conditions d'obtention des attestations). Le backend répond aux questions des utilisateur·rices en effectuant des recherches dans la base de données et en générant une réponse adéquate.



Fonctionnement du RAG avec Embeddings

MindsDB utilise des modèles d'embeddings pour comprendre le contexte des questions posées par les utilisateur·rices. Lorsqu'une question est soumise via le frontend, elle est envoyée à MindsDB où elle est transformée en vecteur numérique. Ce vecteur est comparé aux vecteurs des documents disponibles (par exemple, le règlement sur les attestations) pour trouver les informations les plus pertinentes. Une fois ces informations trouvées, un modèle de génération de texte (comme GPT) génère une réponse complète et cohérente.

Autres technologies envisagées pour l'intégration IA

- **Langchain** : Une option pour la gestion de modèles d'IA conversationnelle.
- **OpenAI GPT-4** : Peut être utilisé pour des générateurs de texte plus complexes en combinaison avec MindsDB.
- **Rasa** : Potentiel remplacement si des besoins spécifiques en NLP se révèlent plus importants.