

Questions Classic

Pourquoi avoir choisi ce projet ?

Parce qu'il répond à un vrai besoin métier et allie chatbot, intelligence artificielle et intégration technique, ce qui m'intéresse fortement.

Quelles sont les limites de votre travail ?

L'intelligence du chatbot peut encore être enrichie, et certaines fonctionnalités avancées comme la supervision ou l'analyse prédictive n'ont pas été développées.

Quelle est votre valeur ajoutée ?

La modélisation métier du recouvrement, l'automatisation des recommandations et l'intégration complète d'une interface intuitive avec une base de données réelle.

Quels outils avez-vous utilisés et pourquoi ?

Rasa pour la couche NLP, Spring Boot pour le back-end, Angular pour le front-end, PostgreSQL pour la base de données, car ils sont performants, bien documentés et compatibles entre eux.

Quel est le besoin métier auquel répond votre application ?

Faciliter le travail des agents de recouvrement en centralisant l'information et en leur proposant des recommandations automatiques adaptées aux dossiers.

Comment fonctionne votre solution de manière globale ?

L'agent interagit avec le chatbot via une interface web. Le chatbot comprend la demande, exécute des actions via des Custom Actions en Python, interroge la base PostgreSQL, puis retourne la réponse à l'utilisateur.

Quelle est la particularité de votre chatbot par rapport à d'autres ?

Il intègre des règles métier complexes spécifiques au recouvrement et recommande automatiquement les procédures à suivre selon le profil du client.

Avez-vous respecté les objectifs fixés au départ ?

Oui, toutes les fonctionnalités principales ont été implémentées et validées.

Quels sont les scénarios utilisateurs les plus représentatifs ?

Recherche client, obtention des créances, recommandation automatique des modules, ajout ou modification d'un bien, suivi conversationnel.

Pourquoi avoir choisi Rasa plutôt qu'un autre framework comme Dialogflow ?

Pour sa flexibilité, son approche open-source, son contrôle total sur les données et ses possibilités d'extension avec Python.

Pourquoi avoir utilisé Spring Boot / Angular / PostgreSQL ?

Spring Boot pour sa robustesse côté back-end et son utilisation de JPA, facilitant le changement de base de données selon les besoins des clients lors du déploiement local. Angular pour un front moderne et dynamique, et PostgreSQL pour sa fiabilité et sa puissance relationnelle.

Comment s'effectue la communication entre vos différentes couches ?

L'interface Angular interagit avec Spring Boot via des API REST. Spring Boot communique avec Rasa et PostgreSQL selon les besoins métier, tandis que Rasa peut également interagir directement avec la base de données lorsque le cas d'usage l'exige.

Comment avez-vous structuré votre base de données ?

Elle est relationnelle, avec des tables pour les clients, créances, biens, cautions, garanties, utilisateurs, et conversations.

Quelle méthodologie de travail avez-vous adoptée ? Pourquoi ?

La méthode agile Scrum avec 5 sprints, pour une meilleure organisation, des livraisons progressives et une bonne répartition des tâches.

Combien de sprints avez-vous réalisés ? Que contenait chaque sprint ?

Nous avons réalisé 5 sprints. Le premier a été consacré à la création des intents et à la configuration de base du chatbot. Le deuxième sprint a porté sur le développement des actions personnalisées et la gestion des réponses contextuelles. Le troisième sprint a permis de mettre en place l'authentification, l'interface agent et l'interface administrateur. Le quatrième sprint s'est concentré sur la recommandation automatisée des procédures contentieuses. Enfin, le cinquième sprint a introduit le tableau de bord administratif et les fonctionnalités de paramétrage du site.

Quels tests avez-vous mis en place et pourquoi ?

Tests unitaires sur le back-end avec JUnit, tests NLU et stories sur Rasa, tests d'intégration avec Postman, pour garantir la fiabilité du système.

Comment avez-vous géré le travail en binôme ?

En répartissant les tâches selon nos compétences, tout en assurant une communication régulière et des validations croisées.

Quels ont été les principaux blocages rencontrés ?

La complexité des règles métier et l'interprétation des fichiers métier Excel. On a surmonté ça avec des réunions régulières et des diagrammes explicatifs.

Comment avez-vous surmonté un problème technique majeur ?

En isolant le bug, en testant chaque couche indépendamment, et en recherchant des solutions via la documentation ou la communauté.

Avez-vous rencontré des limites imposées par les outils utilisés ?

Oui, Rasa nécessite un bon entraînement pour éviter les confusions d'intention, et certaines intégrations demandent beaucoup de configuration.

Comment avez-vous sécurisé l'accès à l'application ?

Par un système d'authentification avec contrôle de rôle, hachage des mots de passe, et vérification des accès côté API.

Quelles mesures de performance ou d'optimisation avez-vous envisagées ? Nettoyage des données NLU, réduction des appels inutiles en base, et chargement paresseux pour le front-end.

Comment sont structurés vos intents et vos actions ?

Les intents couvrent les besoins métier (client, créance, bien...), et les actions personnalisées exécutent les traitements ou accèdent à la base.

Avez-vous utilisé des entités ou des slots conditionnels ?

Oui, plusieurs réponses dépendent du type de client ou de la nature de la demande, via des slots dynamiques.

Comment gérez-vous les cas d'incompréhension ou d'ambiguïté dans les messages ?

Via le fallback, la reformulation de la demande, ou des réponses guidées avec boutons cliquables.

Comment avez-vous testé votre chatbot ?

Avec rasa shell, rasa interactive, rasa test, et des conversations simulées depuis l'interface.

Que pourriez-vous améliorer si vous aviez plus de temps ?

Une meilleure personnalisation des scénarios, l'ajout de feedback utilisateur, et une interface encore plus intuitive.

Comment ce projet pourrait-il être mis en production ?

Le projet sera adapté en fonction des besoins spécifiques du client, puis déployé localement sur son infrastructure, avec une configuration sécurisée et une connexion aux bases de données réelles.

Y a-t-il des fonctionnalités que vous auriez voulu développer mais que vous avez écartées ?

Oui, nous aurions souhaité permettre à l'administrateur de modifier les règles de recommandation via l'interface sans toucher au code, mais cela nécessitait une couche de configuration trop complexe à intégrer dans le temps imparti.

Qu'avez-vous appris grâce à ce projet ?

Ce projet m'a permis d'approfondir mes compétences techniques, d'améliorer ma collaboration en binôme, de mieux comprendre le métier du recouvrement contentieux, et de faire mes premiers pas concrets dans le domaine de l'intelligence artificielle.

Que feriez-vous différemment si c'était à refaire ?

Prévoir plus de temps pour le testing, mieux anticiper la modélisation métier, et structurer les données d'entraînement dès le début.

Quel a été votre rôle précis dans le binôme ?

Modélisation métier, développement back-end et chatbot, intégration Rasa, et suivi de la qualité des livrables.

Ce projet vous a-t-il aidé à définir votre orientation professionnelle ?

Oui, il a renforcé mon intérêt pour les systèmes intelligents et les architectures logicielles avancées. Je souhaite approfondir dans ce domaine.