

# Test Technique Développeur Fullstack



## Contexte

Chez **eKonsilio**, les internautes peuvent discuter en direct depuis Live Chat avec nos conseillers, notre solution conversationnelle pour nos clients.

Votre travail consiste à créer l'interface pour consulter et envoyer des messages.

Le code source sera disponible sur un service web d'hébergement et de gestion de développement de logiciels (github, gitlab...).

En plus de votre code, un README expliquant votre réflexion et vos choix serait apprécié.



## Exercice

Votre projet se compose en un mono-repo de 3 solutions :

- L'interface front-end admin pour les agents.
- L'interface front-end de chat accessible pour les visiteurs.
- Un api back-end pour faire transiter les données.

Globalement l'application permettra de :

- Afficher une liste de toutes les conversations
- Autoriser l'agent à sélectionner une conversation
  - Dans la conversation, il y a une liste de tous les messages entre ces deux utilisateurs.
  - En tant qu'utilisateur, vous pouvez saisir et envoyer de nouveaux messages dans cette conversation.
- Les visiteurs sur le chat pourront eux aussi envoyer des messages à l'agent et consulter les messages de la conversation.

Comme votre application peut être utilisée par des milliers d'utilisateurs, assurez-vous de fournir des protections de sécurité robustes.

Évidemment, c'est à vous de créer quelque chose de beau et de joli, vous êtes libre de le concevoir comme vous le souhaitez. L'application sera utilisable à la fois sur mobile et sur desktop.



## Prérequis à respecter

- Une base de données pour stocker les conversations et les messages (ou plus).
  - Au choix **MongoDb** ou **PostgreSql**
- Un **docker compose** pour faciliter le déploiement de la base de données et de tous les services nécessaires.
- Contrainte front-end : **ReactJs**.
- Contrainte back-end : **NodeJs Typescript**.

---

## Bonus 1

Chez eKonsilio, nous disposons d'une armée de conseillers en ligne que nous désignons sous le nom de Genius. Chacun d'entre eux gère ses propres conversations grâce à son compte dédié. Vous avez la possibilité d'améliorer votre application en la rendant multi-utilisateurs et en instaurant un système qui répartit les nouvelles conversations vers les Genius en ligne et disponibles.

## Bonus 2

Des problèmes d'infrastructure peuvent arriver. Parfois, les serveurs tombent en panne. Peut-être pouvez-vous afficher quelque chose d'agréable pour avertir l'utilisateur et le gérer avec élégance.

## Voulez-vous rendre l'application encore meilleure ?

N'hésitez pas à apporter autant d'améliorations que vous le souhaitez. Nous aimons la créativité et les défis techniques.

Si vous êtes à court d'idées, voici quelques réflexions :

Comme notre application est susceptible d'atteindre des utilisateurs de n'importe où, nous devons nous assurer que l'application fonctionne bien. Que pouvez-vous faire pour que cela soit vraiment rapide ?

Nous aimons tous nous détendre après une dure journée de travail. Il serait dommage que nous ne soyons pas suffisamment confiants quant au déploiement automatique à venir. Êtes-vous sûr que tout a été testé minutieusement ?