

Call dashboard Homework

L'objectif du test est de construire un dashboard minimaliste de suivi des appels reçus par un agent téléphonique virtuel.

Déployé dans un secrétariat médical, l'agent virtuel reçoit des appels de patients pour différents sujets (prise de rendez-vous, demande de renseignements, etc.).

Le dashboard a pour but de permettre aux gestionnaires de suivre l'activité de l'agent et ses interactions avec les appelants.

L'application, développée avec Next.js App Router, devra comporter deux éléments principaux :

- Une API permettant de servir les données relatives aux appels téléphoniques
- Une page affichant la liste des appels téléphoniques reçus par l'agent

1. API appels téléphoniques

La route `/api/calls` renvoie un tableau d'objets JSON représentant les appels téléphoniques reçus par l'agent. Chaque objet comporte les propriétés suivantes :

- `id` : un identifiant unique de l'appel
- `from` : le numéro de téléphone de l'appelant
- `to` : le numéro de téléphone de l'agent
- `date` : la date et l'heure du début de l'appel au format ISO 8601
- `duration` : la durée de l'appel en secondes
- `subject` : le sujet de l'appel (énumération des valeurs possibles : `appointment`, `information`, `prescription`, `other`)
- `summary` : un résumé de la discussion entre l'agent et l'appelant

Pour simuler une base de données, les données d'appel seront stockées dans un fichier JSON `calls.json` à la racine du projet. Ce fichier (fourni en annexe) contient un tableau d'objets JSON représentant les appels téléphoniques reçus par l'agent.

Attention, la base de données des appels est partagée par plusieurs agents différents. Afin de ne renvoyer que les appels téléphoniques concernant l'agent qui nous intéresse, la route API devra accepter un paramètre `to` correspondant au numéro de téléphone assigné à cet agent. Ce paramètre devra être obligatoire permettra de filtrer les appels téléphoniques retournés.

2. Page de listing des appels

La page devra être accessible à l'adresse `/calls` et devra afficher la liste des appels téléphoniques reçus par l'agent sous la forme d'une liste triée par ordre antichronologique de début d'appel. Chaque élément de la liste devra afficher les informations suivantes :

- La date et l'heure du début de l'appel
- Le numéro de téléphone de l'appelant au format français (ex : 01 98 76 54 32, on considère que tous les appelants ont un numéro de téléphone français)
- La durée de l'appel
- Le sujet de l'appel en français

En cliquant sur un élément de la liste, l'utilisateur devra pouvoir afficher le résumé de la discussion entre l'agent et l'appelant. Le résumé devra être affiché dans une popup modale munie d'un bouton de fermeture.

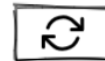
Dans la base de données fournie, l'agent dont on souhaite afficher les appels est identifié par le numéro de téléphone `01 23 45 67 89`.

Une maquette approximative de ce qui est attendu est fournie ci-dessous. Le design final de la page est laissé à votre discrétion.

BONUS : Ajouter un bouton permettant de rafraîchir la liste des appels. Lorsque l'utilisateur clique sur ce bouton, la liste des appels devra être rechargée depuis la route API.

BONUS : Ajouter un bouton permettant de filtrer la liste de manière à n'afficher que les appels relatifs à une prise de rendez-vous.

Journal des appels



Date	Appelant	Durée	Motif
Aujourd'hui à 14h00	01 98 76 54 32	7mn 35s	
01/08/24 à 9h25	01 XX XX XX XX	30s	
01/08/24 à 9h25	01 XX XX XX XX	8mn 21s	
01/08/24 à 9h25	01 XX XX XX XX	10mn 01s	
01/08/24 à 9h25	01 XX XX XX XX	30s	

Popup-fermée

Journal des appels



Date	<div> <div>Détails de l'appel</div> <div> </div> </div>		Motif
Aujourd'hui à 14h00			
01/08/24 à 9h25			
01/08/24 à 9h25			
01/08/24 à 9h25			
01/08/24 à 9h25			

Pop-up ouverte

3. Statistiques

On souhaite ajouter à l'interface utilisateur des statistiques sur l'ensemble des appels de l'agent :

- Durée moyenne de l'ensemble des appels
- Motif le plus fréquent d'appel
- Nombre d'appelants uniques
- Le temps **quotidien** moyen qu'a fait gagner l'agent au secrétariat, i.e. la moyenne du temps que l'agent a fait gagner chaque jour au secrétariat depuis l'appel le plus ancien - on considèrera qu'un appel concernant une prise de rendez-vous a fait gagner 5mn au secrétariat, et que tout autre appel fait gagner 3mn)

Pour anticiper un nombre important d'appels en base de donnée, tous les calculs associés à ces valeurs doivent être réalisés en back-end.

Le design des composants de l'UI est libre.

Contraintes

- Framework imposé : Next.js App Router
- Langage imposé : TypeScript

Il est autorisé d'exploiter autant de librairies que nécessaire pour réaliser le test. Il est également autorisé d'ajouter des fonctionnalités supplémentaires à l'application si vous le souhaitez. Il est recommandé d'ajouter les fonctionnalités dans l'ordre de l'énoncé.

Livrables

Le livrable devra être présenté sous la forme d'un repository Git privé sur votre compte GitHub.

À la fin du test, vous devrez inviter l'utilisateur GitHub `@vaflag` en tant que collaborateur sur le repository.

Évaluation

L'évaluation du test se fera sur les critères suivants :

- Respect des consignes
- Qualité du code
- Qualité de l'interface utilisateur