Stellium

## InterviewExam

# Step 1 : Reésolution des bugs

Lorsqu’on lance l’application on a une erreur de configuration dû à la non-configuration de cosmo db. On renseigne le account , la key et on insère dans cosmo db emulator le content avec comme id stellium

Ensuite dans mon cas, j’ai du adapter ma version de node pour le package node-sass

Ensuite une fois npm install réussi, on voit que dans les dépendances qu’il manque axios. Il faut donc le rajouter.

Ensuite, il faut rendre cohérent les objets. Car item coté api contient id et description et pour le front, on a id et content. Dans mon cas, j’ai remplacé content par description. Escape pour l’encodage posait aussi un problème, je l’ai remplacé par « encodeURIComponent ».

Une fois l’item correctement chargé, il faut corriger l’interface.

1. On positionne l’un en dessous de l’autre le logo et le bouton de chargement des données
2. On centre le container au milieu de la page

# Step 2 Authentification :

Pour utiliser l’authentification d’azure avec, il faut utiliser azure/msal-react

Sur authentification créer un fournisseur d’identité.

Dans un fichier regourer la configuration, l’id de l’application, l’autorité pour l’identification Microsoft et une adresse de redirection l’application locale dans notre cas.

Pour sécuriser le frontend, on crée un composant pour le cas ou l’utilisateur n’est pas authentifié avec un bouton pour lancer l’authentification. Lorsque l’authentification est réussie, l’utilisateur est redirigé vers la page principale.

Au chargement de la page, on récupère les informations du compte, on fait un premier appel à l’Api pour enregistrer l’utilisateur. L’Api enregistre l’utilisateur et lui génère un token. Ensuite se token devra être transmis avec chaque requête de l’application.

Comme dans notre cas on passe le token à la fonction de chargement des données qui va rajouter le header.

Pour l’hébergement de l’application, on peut opter pour un hébergement classique Microsoft sur azure ou sous forme d’image docker afin d’utiliser Kubernete et avoir une application scalable.