

Configurer le docker compose avec vos connexions si elles ne sont pas les même (ports exposés)

!!!!!! Il faut bien vérifier que tous les ports dans le front et dans le back correspondent

back : port 8084

Front : port 3000

penser à bien builder le dockerfile du front et celui du back (sur vscode j'ai juste cliqué droit et j'ai l'option, sinon le faire en ligne de commande (je ne sais pas comment le faire autrement, je vais tenter de retrouver ça))

une fois cela fait, bien regarder le docker compose, et recréer des bdd avec des noms cohérents.

j'ai demandé à copilot les commandes pour dupliquer et remplir les bases

## METHODE 1

déplacer ou dupliquer les fichiers sql et les fichiers de dataset clean dans le dossier initdb qui se place dans le dossier du reverseproxy (nginx)

Rajouter un 01 devant le nom du sql create table

Rajouter un 02 devant le nom du sql import data  
(ça va les gérer dans cet ordre)

modifier les chemins pour que le sql import data prenne bien les fichiers du datasetclean

## METHODE 2

après avoir dupliqué les bases, en fait il suffit d'en faire une sauvegarde (j'utilise pgadmin) avec les options : format plat, et dans options de données cocher predata, data, postdata).

ATTENTION : pour remplir les bases correctement, il faut un fichier différent pour chaque backend, initdb -> fr, initus->db\_us, initch->db\_ch

vous pouvez aussi tout simplement le récupérer sur le git dans la branch windows.  
si jamais vous installez vos bases en local et que ça ne marche pas, c'est parce que docker prends le pas sur le local, il faut que les bases soient installées dans nos conteneurs

commande pour le build final : dans le dossier du proxy

docker-compose down -v

docker-compose up --build

le -v est extrêmement important ! il détruit les volumes, si ce n'est pas fait au build, les bases ne seront pas traitées (à priori sans grande importance si vous avez déjà bien créé et rempli les bases dans un précédent build, mais je n'ai pas testé, je suis pour la théorie de la terre brûlée)

Ah, petite chose importante, peut être avez-vous aussi ce problème, mais le dossier env\_api était corrompu chez moi. J'ai donc du tout réinstaller et faire une upgrade de pip, sans quoi aucune mais vraiment aucune dépendance ne fonctionnait.

et bien sur ce sont les commandes qui fonctionnent sous windows, il est probable que certaines de ces commandes soient à modifier, j'ai utilisé l'extension bito de vscode, elle est vraiment très efficace pour trouver les détails qui font flancher les commandes

traduction

npm install vue-i18n

Npm install pinia

Npm install chart.js

A exécuter dans le front (je l'ai rajouté au docker file )

j'ai modifié les fonctions du front pour éviter d'avoir une page inutile, celle de prédiction graphique. Au lieu de ça en passant par le store, j'ai fait en sorte de récupérer le graph produit par la prédiction (la page3) et on pourra l'afficher dans la page 2.

le back et le front, message apparents et message de logs disposent d'une traduction.