

# **Livrable Santé et Bien-être IA**

## **Sommaire :**

- 1-Mise en place des dépendances
- 2-Préparation de la base de donnée
- 3-Recherche API
- 4-Un peu de frontend

# Mise en place des dépendances:

Contexte :

Pour la préparation du projet, je devais au préalable préparer les fichiers/dossiers pour avoir une ligne directrice de mon projet et ainsi savoir où je devais aller (bien sûr, certains fichiers seront voués à changer en fonction de potentiel imprévu).

Pour cette partie-là donc, je déjà préparé le répo github (<https://github.com/Altrevis/goodlifeapp>) contenant les fichiers et dossiers de mon projet.

Par la suite, il me fallait installer les dépendances de mon projet, j'ai donc au préalable segmenter mes fichiers dans deux rubriques très simples :

-La rubrique backend qui comportera tous mes fichiers liés à la db, à mes connexions d'API ou bien encore à la structuration de mes LLM, tout fait en python flask pour m'en servir avec React.

backend	-le dossier API pour y rajouter tous mes fichiers python liés à l'API de santé que je rajouterais.
> api	
> db	
> llm	-La section DB pour copier les colonnes et les valeurs de ma base de données.
> venv	
app.py	-La partie LLM pour y rajouter mon IA et pouvoir ainsi l'utiliser.
requirements.txt	

-La partie venv est là pour installer un environnement python pour ainsi manipuler les commandes python et ainsi avoir un petit serveur en local qui tourne bien.

-Et enfin le fichier txt pour y rajouter toutes mes dépendances.

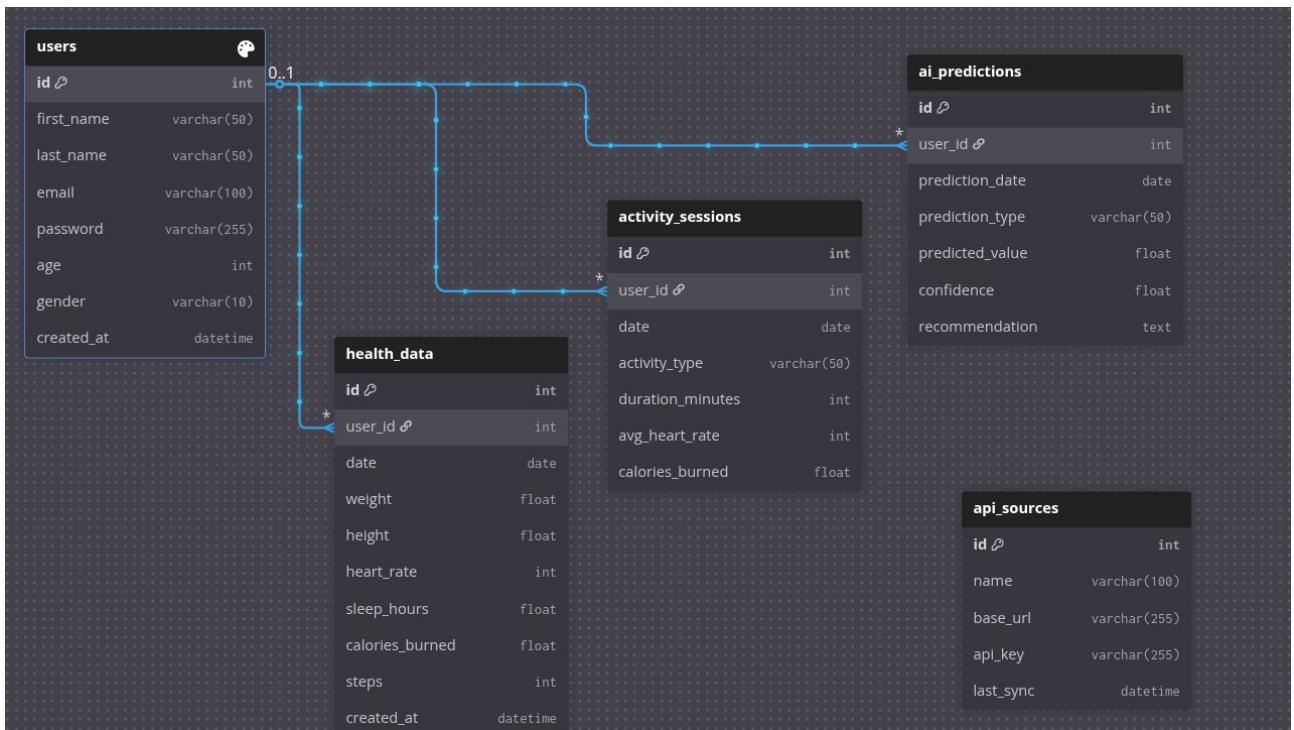
-La rubrique frontend qui lui, contient les fichiers liés à React typescript. Il présentera donc mes fichiers liés au frontend et donc, la partie visuelle de mon projet.

frontend	-le dossier node_modules servant de base pour que toutes les dépendances JS soient bien installées.
> node_modules	
> public	
> src	-le gitignore pour éviter de télécharger des fichiers / dossiers qui seraient soit trop lourds, soit pas compatibles d'un pc à un autre.
◆ .gitignore	
{ package-lock.json	-le README qui sera tenu continuellement à jour pour pouvoir mettre les commandes pour installer les dépendances et autres.
{ package.json	
① README.md	
tsconfig.json	

## Préparation de la base de donnée:

Pour la mise en place de la db, je me suis autant servie des valeurs que je pouvais trouver sur internet, que de mes connaissances mais aussi des IA pour trouver les meilleures informations à devoir traiter et ainsi ne pas me perdre sur les multitudes de valeurs possible à traiter.

C'est d'ailleurs pour ça que j'ai commencé à faire la base de données sur un logiciel du nom de dbdiagram.io pour ainsi faire un schéma claire et simplifié de ma base de donnée :



Chaque table représente des zones de valeurs que je devrais soit stocker soit qui seront stockés par les utilisateurs eu même.

Tout d'abord la table users est là surtout pour renseigner les informations de l'utilisateur et ainsi pouvoir envoyer un premier jeu de donnée à la base de donnée pour que plus tard, avec mon code, ça puisse préparer ce qu'il faut pour l'utilisateur.

La table health\_data est là pour que l'utilisateur puisse renseigner les informations de son état de santé actuel mais aussi sa corpulence en général (poids, taille, nombre d'heure de sommeil etc.) Elle sera là pour renseigner d'autre jeu de donnée pour ma table de prédition.

Activity\_sessions sera là table des activités que l'utilisateur devra accomplir peut-être sous format de quête pour rajouter un côté amusant et ludique et ainsi ne pas faire passer ces activités pour des corvées longue et pénible.

Pour stocker les prédictions que l'IA fera pour préparer les planning et futur projet sportif de l'utilisateur, la table ai\_predictions sera faite pour ce détail là, ce qui fera que les valeurs que l'utilisateur renseignera seront elles aussi utilisées pour aider à la prédition de la table.

Enfin la table api\_sources est là pour renseigner les clé api ainsi que les autres valeurs similaire pour permettre de ce servir des données données que l'api pourra me fournir. Bien que cette table reste surtout optionnel, elle gardera tout de même pratique pour du stockage rapide.

## Recherche API:

Pour la partie API, je me suis avant tout renseigné sur tout ce que je pouvais trouver qui serait compatible avec mes LLM de LM studio. Pour le moment ça reste encore et état de recherche, ne sachant pas vraiment pour le moment sur quoi me porter, bien que Mistral 7B me semble être une bonne approche vu qu'il coche pas mal de case pour ce que je demande :

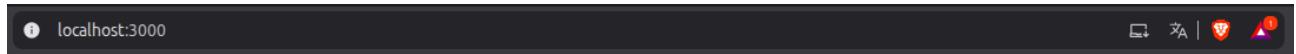
Fonction	LLM	API
Analyse sommeil / activité / calories	LLaMA 2 / Mistral 7B	Fitbit / Garmin / HealthKit
Prévisions performance / fatigue	Mistral 7B	Données historiques user + Santé Publique France
Suggestions nutrition / repos	LLaMA 2	Edamam API + données utilisateur
Enrichissement infos santé	Mistral 7B	OMS / European Health Information API

Pour les API, je regarderais mais il ce peux que je récupère plusieurs API pour plusieurs informations qui pourrait être différente en fonction des API choisis. C'est d'ailleurs une meilleure façon de pouvoir faire des moyens pour mon LLM et ainsi faire des données très proches du réel.

Catégorie	API	Notes	🔗
Wearables & fitness	Fitbit API	Récupère sommeil, rythme cardiaque, pas, calories, distance.	
Wearables & fitness	Garmin Health API	Similaire à Fitbit, très complet pour l'activité et la performance.	
Santé générale	Apple HealthKit API	Si tu cibles iOS, données santé complètes.	
Nutrition	Edamam API	Infos nutritionnelles et suivi alimentaire.	
Santé publique	Santé publique France / OMS / European Health API	Pour enrichir ton LLM avec des données statistiques et tendances générales.	

# Un peu de frontend:

Résultat obtenu après les premier ajout ( bienetre étant le nom de ma base de donnée ).



## Frontend React (TypeScript) connecté à Python 🚀

Connexion réussie à bienetre ✓

Une fois la base faite, j'ai pris la décision de prendre un peu d'avance sur la partie frontend du projet, pour le moment juste de quoi avoir trois page de faire le temps de préparer le reste et ainsi avoir un visuel de ce que je pourrais faire.

Page graphique :

-Pour le moment, seul une case est visible, mais à l'avenir, plusieurs graphique seront visible montrant donc les valeurs renseigné par l'utilisateur.

Page chat :

-La partie chat de santé est simple et restera probablement dans le même style étant donné que ça doit rester simple et facile à comprendre pour tout type de personne.