## 

## **PROJET 5 – Pure Beurre**

## 

**1.Présentation du projet**

Le but du projet est de pouvoir interagir avec la base de données Open Food Facts pour récupérer des produits et leurs substituants.

Pour ce projet, j’ai choisi de découper le programme en 5 fonctions principales ( voir la photo ci-dessus) en utilisant la méthode agile et avoir plus de contrôles via l’interface du programme :

* Une fonction qui lance la création de la base de données.
* Créer des comptes utilisateurs avec vérifications des doublons avant l’insertion.
* Accès aux sauvegardes par utilisateur.
* Recherche d’un produit en entrant le nom ou premières lettres de la catégorie.
* Une fonction qui supprime totalement la base de données

La connexion et les requêtes vers la base de données sont effectuées via la class *SqlRequest* du fichier “*database.py”*, Ont peut parametrer les identifiant et adresse de connexion au serveur via les 3 variables en tête du fichier “*start.py*”

La fonction de création de la base de données ”*create\_db*” va récupérer le fichier SQL “*database.sql*” et l’insérer via des requêtes qui aura au préalable séparé chaque commande du fichier via la fonction *split()*

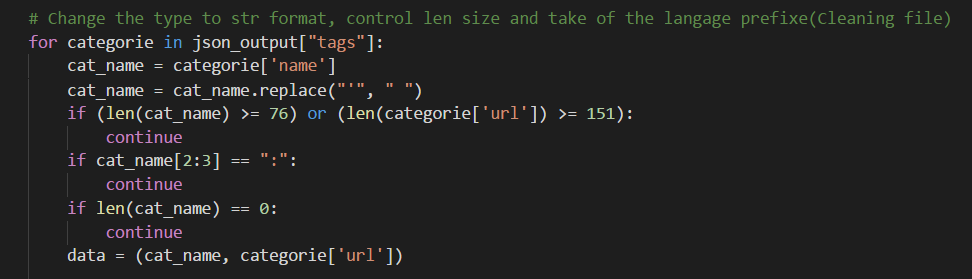
Les produits sont téléchargés au fur et à mesure de leurs sélections dans la recherche sur la table *“product”* et stocker pour pouvoir être enregistré via une clé étrangère sur la table *“saved”*

Concernant l’algorithme de substitution, je me suis basé sur le tags “nutrition\_score\_debug” et c’est 2 derniers caractères qui correspondent au score sur 100. Il est ensuite comparé aux autres scores de la même catégorie pour les afficher

Vous pouvez trouver le projet sur [GITHUB](https://github.com/pic0de/PurBeurreP5) via c’est lien.

**2.Difficultés rencontrées**

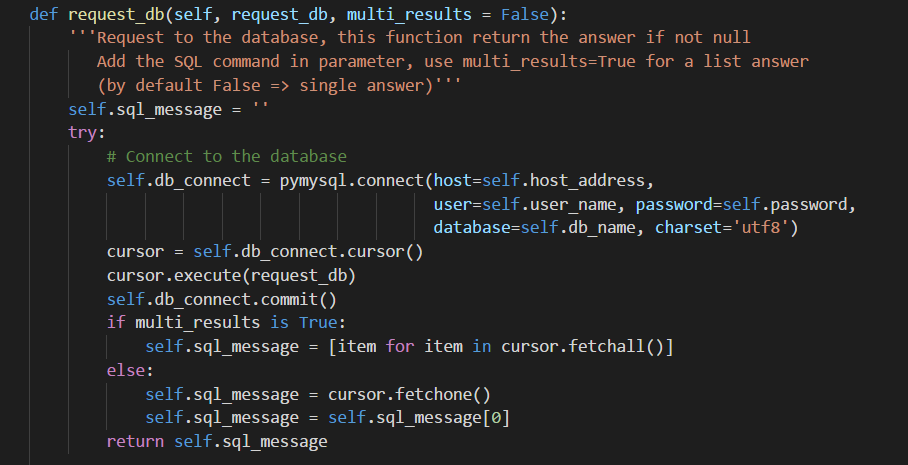
J’ai eu beaucoup de difficultés avec la base de données, car elle possède beaucoup de données erronées, au mauvais endroit, vide ou très long. j’ai dû développer un algorithme de tri qui me supprime environ 15% des données du “categories.json” FR.



Concernant les retours de requêtes SQL, j’ai eu du mal avec les résultats obtenus à l’unité ou multiple.

J’ai donc modifié la fonction de requête “request\_db”, qui permet de récupérer selon si on lui donne le paramètre “True” un résultat “fetchone()” ou “fetchall()”

J’ai dû aussi créer une boucle dans le résultat multiple pour renvoyer une liste.



Le dernier souci rencontré a été à propos de l’instance de la classe, je devais ré instancier à chaque utilisation d’une de ses fonctions. j’ai résolu l’erreur, c’était dû à 2 variables de la classe qui avait le même nom.