**Projet 5 - Utiliser les données publiques d'Open Food Facts**

**Généralités :**

Programmes écrits en Python3. Dépôt Github <https://github.com/Rouizi/OC_project5>

**Librairies utilisées:**

• **requests** (pour utiliser des données de l’API d’Open Fodd Facts).

• **mysql-connector-python**: communication avec la base de données.

**Base de données** : Mysql.

**Description du programme** :

Lorsque le programme est lancé une base de données *alimentation* est créée avec les tables *Category*, *Product* et *Susbstitute* en faisant appel à la classe *database,* après on insert les données dans les tables en utilisant l’API d’Open Food Facts grâce à la classe *OpenFoodFact* (on prend seulement 10 catégorie et 50 produits par catégorie pour ne pas surcharger la base de données).

Une fois la BDD créée le programme affiche un menu principal avec 2 choix possibles :

1. Quel aliment souhaitez-vous remplacer ?
2. Retrouver mes aliments substitués.

L’utilisateur sélectionne 1. Le programme pose les questions suivantes à l’utilisateur et ce dernier sélectionne les réponses :

. Sélectionnez la catégorie. [Plusieurs propositions associées à un chiffre.

L’utilisateur entre le chiffre correspondant et appuie sur entrée]

. Sélectionnez un produit. [Plusieurs propositions associées à un chiffre.

L’utilisateur entre le chiffre correspondant à l’aliment et appuie sur entrée]

. Le programme propose un substitut, sa description, un magasin ou

l’acheter (le cas échéant) et un lien vers la page d’Open Food Facts

concernant cet aliment.

. L’utilisateur a alors la possibilité d’enregistrer le résultat dans la base de données.

Si l’utilisateur choisi de sélectionner les catégorie alors une requête « SELECT » est faite sur la table *Category* pour afficher les catégories avec un chiffre avant chaque catégories, l’utilisateur choisi une catégorie et ses produits sont affichés avec une requête « SELECT » sur la table *Product*. L’utilisateur choisi un produit, on lui affiche le nom du produit et son nutriscore . Pour faire ça on utilise l’url du code barre du produit, on l’a convertie au format json (donc on utilise l’API du site) et on extrait du contenu de l’url du produit son nutriscore et le nom des catégories auxquelles il appartient (un produit peux appartenir à plusieurs catégories).On prend le nom de la catégorie qui contient le moins de produit et on cherche sur celle-ci un substitut, sa description (marque, quantité, ingrédients, nutriscore), un magasin ou l’acheter (le cas échéant) et un lien vers le site (si le produit à un nitriscore de E on doit trouver un substitut avec un nutriscore de D, C, B ou A).

Si l’utilisateur choisi de sélectionner un produit alors on affiche une liste de 50 produits et on propose à l’utilisateur d’afficher la suite ou de sélectionner un produit dans la liste (on a 10 catégories et 50 produits par catégorie, donc 500 produits en tous), si il veut afficher la suite on affiche les 50 produits suivant jusqu’à afficher tous les produits sinon le programme refait la même chose que dans le choix d’une catégorie.

En fin, et dans les deux cas, on propose à l’utilisateur soit d’enregistrer le substitut et sa description dans la base de données, soit de sélectionner un autre produit, soit de retourner au menu secondaire, soit de retourner au menu principal. Si l’utilisateur choisi d’enregistrer le substitut on l’enregistre dans la table *Substitute* avec une requête *« INSERT ».*

L’utilisateur a aussi la possibilité de consulter tous les substituts des produits qu’il a choisi d’enregistrer, pour cela on fait un simple « SELECT » sur la table *Substitute*.