

PROJET 5 : Utilisez les données publiques de l'OpenFoodFacts

Lien GitHub: <https://github.com/Tony380/Projet5>

Lien Trello: <https://trello.com/b/GZ0Q6qKK/projet-5>

CADRE DE DÉPART:

Après avoir généré un repo Git et un environnement virtuel pour démarrer ce projet, j'ai créé un tableau *Trello* avec des user-stories pour définir les fonctionnalités du futur programme. J'ai ensuite, dans ce même tableau, créé des tâches, pour planifier les étapes de création du programme.

J'ai aussi créé un fichier *README.md*, dans lequel j'ai inclus les user stories et expliqué comment fonctionne le programme.

Pour ce projet, la méthodologie utilisée étant le "Readme Driven Development" ou "Doc Driven Development", ces deux fichiers sont donc nécessaires à la construction du programme.

BASE DE DONNÉES:

J'ai créé une base de données, nommée "*Purbeurre*", dans le fichier *database.sql*, afin de stocker les informations qui doivent être extraites de l'API d'OpenFoodFacts. Cette base de données contient trois tables (*Category*, *Product* et *Substitute*).

Cette démarche est accompagnée de la création d'un modèle physique de données avec *Draw.io*, qui décrit la structure de la base de données et des tables qui la composent.

Le fichier *database.sql* sera, plus tard, inclus dans le script Python.

DÉVELOPPEMENT:

Création du fichier *program.py* qui va contenir la classe "*Program*" ainsi que toutes ses méthodes, pour la création de la base de données, son remplissage, et le fonctionnement du programme.

La classe "*Program*" a en attributs les informations nécessaires pour se connecter à la base de données.

On peut citer quelques méthodes, comme par exemple, la méthode "*db_creation()*" qui va créer la base de données grâce au fichier *database.sql*, la méthode "*db_fill()*" qui va la remplir, ou encore la méthode "*display_substitute()*" qui va afficher l'alternative plus saine d'un produit.

Le fichier *program.py* aura besoin des imports suivant :

- *"requests"* : pour se connecter à l'API d'OpenFoodFacts.
- *"mysql.connector"* : pour se connecter à la base de données *"Purbeurre"*
- *"config"* : fichier que j'ai créé pour que l'utilisateur configure les données qui lui sont propres, comme son *"localhost"*, son nom d'utilisateur ou son mot de passe, afin qu'il puisse se connecter à la base de données *MySQL*.

Le besoin de l'installation de ces librairies est précisé dans le fichier *"requirements.txt"* créé à cet effet.

Par la suite, j'ai créé le fichier *main.py* qui va contenir notre programme principal.

Ce programme va avoir un affichage en console pour que l'utilisateur puisse choisir un produit parmi ceux présents dans notre base de données, et avoir son substitut.

Le fichier *main.py* nécessitera donc l'import du fichier *program.py*.

La méthode utilisée pour la construction du programme est l'utilisation de boucle *"while"* représentant des menus et des sous menus et l'affichage par des *"print()"* de textes indiquant les options que peut choisir l'utilisateur.

A cela, s'ajoute l'instanciation de la classe *"Program"* pour se connecter à la base de données et toutes ses méthodes pour afficher les choix de l'utilisateur, ainsi que l'utilisation de conditions tel que *if/elif/else*.

CONCLUSION :

Ce projet a nécessité l'apprentissage du fonctionnement d'une base de données et du langage *SQL*, ainsi que l'apprentissage du fonctionnement d'une API.

Les premières difficultés rencontrées furent la recherche et l'extraction de données via une API.

Ces difficultés ont été surmontées grâce à la documentation de l'API d'OpenFoodFacts qui explique très bien la recherche des données par critères de sélection, ainsi que la documentation de la librairie *"requests"* pour en extraire les informations voulues.

Une autre difficulté rencontrée fut la sélection d'un substitut au hasard, qui fut surmontées grâce à la méthode *"RAND()"* du langage *MySQL*.

Enfin, la sauvegarde d'un produit et de son substitut, ainsi que leur affichage simultanément, ont aussi été sources de difficultés. J'ai trouvé la solution en créant une troisième table (*Substitute*) contenant les *"id"* des produits et des substituts sauvegardés, avec des clés étrangères reliant cette table à celle des produits (*Product*). Il suffit donc d'aller chercher les produits et leurs substituts par leur *"id"* et de les afficher en console.

Pour conclure, le cahier des charges étant assez restreint, il serait possible d'améliorer ce programme en présentant, par exemple, plus de choix et de possibilités à l'utilisateur.