

Ce projet est disponible sur Github à l'adresse:

<https://github.com/mehdi-monfort/-P5-OpenFoodFacts>

## **1. Ma démarche:**

### **1. Début du projet :**

J'ai d'abord rédigé le README.md avec l'explication des différents besoins de l'API avec les détails des différents menus et choix. Puis est fait le projet agile détaillant le parcours du projet et la manière dont j'allait m'y prendre jusqu'à sa concrétisation, le modèle physique de données et enfin je me suis mis à la programmation.

### **2. Le programme :**

L'idée était d'apprendre à faire les requêtes avec les bases de données de l'OpenFoodFact, de les utiliser dans un code python pour les afficher sur la console. J'ai commencé par cela.

Puis est créée la database en python (qui utilise un fichier SQL disposant des différentes tables), avec l'aide de la [documentation](#). J'ai, par la suite, rempli les tables à l'aide de la commande « .get », testé les différentes requêtes d'affichage nécessaire au programme.

J'ai terminé par la « main » où est créée le menu et d'où les différents appels sont faits pour l'affichage console.

## **1.1. Constitution de l'API :**

Le programme est constitué de :

Un README.md

Un fichier config.py avec la configuration de la base de données (modifiable par l'utilisateur)

Un fichier main.py contenant le menu de l'application et ses différents choix.

Un dossier Database contenant les fichiers :

create.py pour la suppression et création de la base de données

filling.py permettant de remplir la base de données

Un dossier SQL contenant les fichiers :

table.py avec le code SQL des différentes tables

request.py avec les différentes requêtes nécessaires à l'application

## **2. Difficultés :**

Ma difficulté principale est de mettre le code en orienté objet, il me restera la main à améliorer. Mon mentor m'a aidé à débloquer la situation grâce à ses explications ce qui m'a facilité la tâche par la suite et devrait se ressentir sur la suite de la formation.

J'ai donc dû refaire mon code pour l'orienté objet et également pour y insérer les requêtes directement en python sans passer par un fichier SQL comme c'est le cas pour mes tables. J'ai également dû travailler sur mes « cursor.execute » qui ne se trouvaient pas tous dans le même fichier et qui étaient faites, en partie, dans la main.

Désormais toutes mes requêtes se font dans le même fichier en python à l'aide de la commande cursor.execute(), le code n'en est que plus compréhensible.

## **2.1. Améliorations possibles :**

Modifier la main pour l'orienter objet avec une classe disposant d'une méthode pour chaque menu.

Créer un fichier avec les différents messages d'erreur (ex : les prints, erreurs...).

Utiliser un mot de passe chiffrer plutôt que de l'avoir en clair dans le programme.

Utilisation de Tkinter pour créer une interface graphique.

Utilisation de Flask pour en faire une application web, cela permettrait de se rapprocher d'une demande client.

## **3. PEP8 et vérification :**

Après utilisation de flake8 à la racine, la majorité des erreurs rapportées étaient dû à la longueur des lignes. Supérieur à 79 caractères.

Une erreur de syntaxe qui semble être une erreur de flake8, elle à été rapporté à de nombreuses reprises comme étant un faux-positif dans certaine condition.

J'avais également une erreurs dans la main dû à un try, except car je n'avais pas préciser d'erreur pour l'exception.