Soutenance du projet 5 : Utilisez les données publiques de l'OpenFoodFacts !

Julien MILLOT

(Alias « julien millot » sur OC, alias « Jilvo » sur Github)

1 : Créer le cadre de départ :

https://github.com/Jilvo/OC_OpenFoodFacts

https://trello.com/b/UsfRiLHA/openclassrooms-projet-5

Après la création du repo Git, j'ai commencé à créer le Readme.md en listant les fonctionnalités nécessaires puis j'ai commencé par lire les cours sur les API pour savoir comment faire des « requests » via l'API d'OpenFoodFacts. J'ai d'abord utilisé PostMan pour faire une approche des requêtes et voir quels sont les données dont j'avais besoin et comment les récupérer. J'ai ensuite lu un cours de SQL pour pouvoir créer ma base de données avec Wamp et pouvoir utiliser MySQLConnector.

2 : Construire la base de données :

J'ai tout d'abord réalisé un modèle physique de données afin d'être sûr des données à utiliser, leurs tailles, leurs types, et de la façon de les classer dans la base de données par en créant les PRIMARY KEY et en rajoutant des contraintes via les FOREIGN KEY.

Une fois que mon mentor a validé ce modèle physique, j'ai commencé à créer la base de données puis d'y faire les tests afin d'y insérer les données récupérées de l'API OpenFoodFacts. Pour construire la base de données, j'ai utilisé PhpMyAdmin, puis j'ai exporté le fichier SQL.

La connexion avec la base de données se fait avec la méthode *open_data_base,* et le module MySqlConnector qui gère les champs du mot de passe, user, hôte... et fait ensuite un commit pour informer l'utilisateur si la connexion est un succès ou non.

3 : Construire le programme :

Le programme se situe dans une classe *Program*. J'ai construit le programme avec de la POO, c'est-à-dire que lorsque dans ma fonction principale je choisis de voir mes produits substitués, cela appelle une fonction qui va les afficher avec un formatage digeste et compréhensible pour l'utilisateur, ensuite pour choisir un produit cela appelle une fonction qui va m'afficher les catégories, m'en fais choisir une, appelle une fonction qui va afficher les produits correspondant à la catégorie et me fera choisir un produit, une fois le produit affiché, le programme va nous proposer un produit au hasard dans cette catégorie, produit initial compris et nous demandera si l'on veut qu'il substitue le produit original, si oui il l'ajoute à la table des *substitut_product*, et l'affiche, si non, il affiche la table puis ferme le programme.

4 : Interagir avec la base de données :

Le module *request_data*, lui possède 2 méthodes, une qui analyse le contenu de la base de données, si elle est vide appelle la fonction *fill_in_db*, qui va la remplir, si elle n'est pas vide, elle n'installe rien et on passe à la partie suivant du programme.

Dans le cas où la database est vide, la fonction *fill_in_db*, se constitue de deux boucles, la première est une ouverture d'un fichier .txt ,qui contient le nom des dix catégories nécessaires, avec une lecture de ce fichier qui ajoute à chaque ligne du fichier une catégorie sur la base de données, en même temps dans la deuxième boucle imbriqué ,fait une requête avec le nom de la catégorie et prends vingt produits où sont indiqués le grade nutritionnel et un magasin d'achat, si il manque un de ces 2 critères, on passe au produit suivant, une fois que l'on a les vingt produits on passe à la catégorie suivante jusqu'à avoir parcouru le fichier .txt soit dix catégories.

Problèmes rencontrés :

Lors du projet précédant j'avais quelques soucis sur notamment la structure du programme, l'algorithmie, la POO…, j'ai donc pu commencer sereinement de ce côté, j'ai toutefois eu du mal à bien versionner le projet.

J'ai eu des difficultés lors de la création du modèle physique de données, car je n'arrivais pas à comprendre l'intérêt d'un tel diagramme dans la réalisation du projet mais une fois réussi j'y ai entrevu l'intérêt. Et pour finir, quelques problèmes avec Trello pour lister tous les besoins et les objectifs.