

Démarche choisie

I) Début du projet

En premier lieu, lorsque je me suis lancé dans ce projet, j'ai choisi d'appliquer les instructions à la lettre : organisation du travail. J'ai choisi Trello pour créer un tableau afin de séquencer mon projet, et ainsi appliquer une méthode agile pour en venir à bout. Voici le lien : <https://trello.com/b/edCZfAZl/projet-5>. Puis j'ai créé mon premier repository GitHub à l'adresse : <https://github.com/raoofrachidi/Projet-5>. Dans la foulée j'ai aussi écrit le fichier README associé, afin de me donner une direction.

J'ai alors rencontré mon mentor pour ce projet : Thierno Thiam. Avec qui nous avons eu une discussion sur les aspects techniques de ce projet. Ayant déjà réalisé les projets 6 et 9 du parcours de Développement d'Application Python avant celui-ci, je n'ai eu aucun problème à créer une base de donnée fonctionnelle ainsi que le Modèle Physique de Données. Je me suis servi de SQL Power Architect pour faire mon Modèle Physique de Données, quant à la base de données, j'ai utilisé MySQL Workbench.

II) Difficultés rencontrées et solutions choisies

J'ai réalisé une première version de mon programme mais j'ai rencontré quelques problèmes. Au départ je n'en ai pas vraiment eu, malheureusement mon ordinateur m'a lâché et donc, j'ai dû le remplacer. C'est là que les problèmes ont commencé à s'accumuler. Alors que tout fonctionnait parfaitement, j'ai commencé à avoir des erreurs qui n'étaient pas présentes avant. Par exemple, j'en avais une lorsque je voulais remplir ma base de données, il s'est avéré au final que ça n'était qu'un problème d'agencement. Avec l'aide de mon mentor, j'ai simplement modifié l'ordre des lignes et cela a suffi pour arranger cela. Ensuite j'ai eu un souci car malgré le fait que la base de données était remplie, je n'arrivais pas à sélectionner une catégorie.

C'est à ce moment que j'ai décidé de reprendre ce projet à zéro. En effet j'ai décidé de recommencer, mais cette fois, en prenant compte de tous les conseils que m'avait donné mon mentor, et avec tout l'expérience que j'avais pu acquérir en réalisant ce projet une première fois. J'ai bien séparé mes fichiers afin d'avoir un code lisible et facile à comprendre.

III) Conception du programme

Comme je l'ai dit précédemment, mon programme est bien séparé. J'ai commencé avec un fichier constants.py. J'y regroupe donc toutes les constantes : tout ce qui relève de la connexion à la base de données (le nom de la base de données, l'host, l'utilisateur et le mot de passe), les fonctionnalités d'affichage (le nombre de produit à afficher, le nombre de substituts à afficher, le nombre de produits que l'on doit charger de l'API) et enfin une liste des noms des catégories à partir duquel vous voulez chercher des produits. Ensuite j'ai mis dans le fichier database.py ma chaîne de connexion à la base de données ainsi que la fonction qui charge les données de l'API dans un fichier json et qui me le retourne.

Concernant les tables de ma base de données, je les ai divisées en 3 fichiers : un pour

chacunes d'entre elles (product, favorite et category). Ils ont été créés sur une même base : les 2 premières fonctions servent à la création de la table puis à l'insertion de données à l'intérieur. Il y a en plus, dans le fichier favorite, une fonction qui permet d'afficher la liste des favoris et une autre qui permet de supprimer les favoris enregistrés. Concernant le fichier product, 2 fonctions ont aussi été ajoutées. Une première permettant d'afficher les produits de la catégorie sélectionnée et une seconde afin d'afficher une liste de produits de substitution avec un meilleur nutriscore.

Dans mon fichier sql.py, je vérifie si la base de données demandée n'a pas déjà été créée. Si ce n'est pas le cas, je la crée automatiquement avec une autre fonction. Ensuite il lance les fonctions des fichiers précédents afin d'y créer les tables et insérer les données.

Le fichier functions.py contient différentes fonctions du programme notamment concernant l'affichage du menu. On peut aussi y trouver 3 autres fonctions. La première va permettre de vérifier que l'on entre bien un entier valide (c'est-à-dire qui est contenu dans l'intervalle donné). La seconde permet d'être sûr que l'utilisateur entre bien « oui » ou « non » et cela peu importe la manière dont c'est écrit (exemple : « nOn »). La dernière sert à gérer l'affichage des produits.

Pour le fichier app.py, on peut y trouver 4 fonctions. Elles servent à afficher : les favoris dans la base de données (ou à les supprimer), les produits et enfin les produits de meilleure qualité par rapport au produit choisi par l'utilisateur. La dernière fonction, quant à elle, enregistre les produits sélectionnés dans la base de données (ou non).

Le fichier choice.py permet de mettre à jour la base de données (il faut effectuer cela 1 fois par semaine). Il permet aussi de quitter le programme. Ainsi que d'afficher les favoris à travers la fonction du fichier functions.py et il en est de même pour les produits. Cela va permettre de gérer comment les résultats vont aussi être affichés.

Enfin le dernier fichier, le fichier main.py, va permettre de lancer le script.