

Projet 5 : Utilisez les données publiques de l'OpenFoodFacts

La startup *Pur Beurre* travaille connaît bien les habitudes alimentaires françaises. Leur restaurant, Ratatouille, remporte un succès croissant et attire toujours plus de visiteurs sur la butte de Montmartre.

L'équipe a remarqué que leurs utilisateurs voulaient bien changer leur alimentation mais ne savaient pas bien par quoi commencer. Remplacer le Nutella par une pâte aux noisettes, oui, mais laquelle ? Et dans quel magasin l'acheter ? Leur idée est donc de créer un programme qui interagirait avec la base Open Food Facts pour en récupérer les aliments, les comparer et proposer à l'utilisateur un substitut plus sain à l'aliment qui lui fait envie.

Cahier des charges

Description du parcours utilisateur

L'utilisateur est sur le terminal. Ce dernier lui affiche les choix suivants :

- 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 l'aliment. [Plusieurs propositions associées à un chiffre. L'utilisateur entre le chiffre correspondant à l'aliment choisi 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.

Fonctionnalités

- Recherche d'aliments dans la base Open Food Facts.
- L'utilisateur interagit avec le programme dans le terminal, mais si vous souhaitez développer une interface graphique vous pouvez,
- Si l'utilisateur entre un caractère qui n'est pas un chiffre, le programme doit lui répéter la question,
- La recherche doit s'effectuer sur une base MySql.

Etapes

1 - Organiser son travail

Découpez votre programme en user stories puis en tâches et sous-tâches. Créez un tableau agile et affectez des deadlines. Avant de coder, initialisez un repo Github et faites votre premier push.

Puis commencez à écrire la documentation. Oui, en premier ! Je vous propose une méthodologie de travail assez reconnue dans le monde du développement : le "Doc Driven Development" ou "Readme Driven Development". Créez simplement un fichier texte appelé Readme.txt.

Lorsque vous commencerez une nouvelle fonctionnalité, écrivez en premier la documentation. Que souhaitez-vous que votre programme fasse ? Comment le développeur comprendra le code ? Puis codez le nécessaire pour que votre programme "valide" le Readme.

2 - Construire la base de données

Avant de vous atteler aux différentes fonctionnalités de votre Readme, commencez par vous questionner sur les informations dont vous avez besoin et dessinez le schéma de la base de données. Quelles informations allez-vous enregistrer ? Quelles données allez-vous manipuler ?

Puis intéressez-vous aux données externes. La base Open Food Facts a une API (expérimentale pour le moment) qui vous permet de récupérer les données voulues au format JSON. Vous pouvez consulter la [documentation de cette API](#).

Créez la base de données : tables et clés étrangères.

Puis écrivez un script Python qui insèrera les données récoltées de l'API dans votre base.

3 - Construire le programme

Listez les fonctionnalités de votre programme pour vous interroger sur les responsabilités de chaque classe. Puis construisez l'architecture voulue.

4 - Interagir avec la base de données

Vous avez la base de données et vous avez les classes. Bravo ! À présent, permettez à votre utilisateur d'interagir avec la base de données.

Commencez par travailler sur le système de question réponse (input, validation des champs). Puis concentrez-vous sur la recherche : quelles requêtes SQL ? Dans quelle(s) table(s) ?

Enfin, cherchez comment enregistrer les données générées par le programme pour que l'utilisateur les retrouve.

Livrables

- **Modèle physique de données** (ou modèle relationnel) et utilisant l'outil informatique de votre choix (pas de dessin à main levée !).
- **Script de création** de votre base de données
- **Code source** publié sur Github

- **Tableau Trello**, Taiga ou Pivotal Tracker.
- Document texte expliquant la **démarche choisie**, les difficultés rencontrées et les solutions trouvées et incluant le lien vers votre code source sur Github. Développez notamment le choix de l'algorithme et la méthodologie de projet choisie. Expliquez également les difficultés rencontrées et les solutions trouvées. Le document doit être en format pdf et ne pas excéder 2 pages A4. Il peut être rédigé en anglais ou en français, au choix, mais prenez bien en considération que les fautes d'orthographe et de grammaire seront évaluées !

Compétences évaluées

- Sélectionner les langages de programmation adaptés pour le développement de l'application
- Travailler en respectant une méthodologie de projet Agile
- Implémenter le schéma de données dans la base
- Respecter les bonnes pratiques de développement en vigueur