

Parcours développeur d'application - Python

Projet 8 : Créez une plateforme pour amateurs de Nutella

1 Description du projet

La startup Pur Beurre souhaite que nous développions pour elle une plateforme web à destination de ses clients. Le but de cette plateforme est de proposer un substitut sain à un produit choisi par l'utilisateur. L'objectif de ce projet est donc de développer une application web disposant des fonctionnalités suivantes :

- L'utilisateur doit pouvoir chercher un aliment pour lequel l'application lui trouvera un substitut.
- L'utilisateur doit pouvoir s'inscrire et s'authentifier sur l'application afin de sauvegarder ses aliments préférés.
- Le projet doit être réalisé à l'aide du framework Django et doit être responsive.

Afin de répondre aux mieux aux attentes de l'équipe Pur Beurre, un cahier des charges est fourni, ainsi qu'un thème pour le site web.

2 Réalisation du projet :

2.1 Etude du projet

Avant de me lancer dans la réalisation du projet, et conformément aux consignes de celui-ci, j'ai commencé par lister les différentes étapes à réaliser au sein d'un tableau créé grâce à l'outil Trello. Le tableau est disponible à l'adresse suivante : <https://trello.com/b/YZiYD4Kv/p8cr%C3%A9ez-une-plateforme-pour-amateur-de-nutella>.

Ce document n'abordera pas toutes les étapes apparaissant sur le tableau afin de se concentrer sur les points qui me semblaient les plus intéressants à détailler.

Les autres aspects du développement de l'application seront abordés lors de la soutenance orale.

2.2 Algorithme de recherche de substitut

Le point central de l'application est la proposition par la plateforme d'un substitut pour un aliment. Je disposais d'une très grande masse d'informations grâce à la base de données OpenFoodFacts. Cependant, il a fallu faire des choix afin d'obtenir les résultats les plus cohérents possibles.

N'ayant pas de réelles connaissances en diététique, les critères de recherche de substituts restent relativement basiques et mériteraient une étude plus poussée afin d'obtenir des résultats plus en accord avec les attentes des consommateurs.

Le premier obstacle dans cette étape a été de faire le tri dans les produits. L'application dispose de sa propre base de données PostgreSQL. Cette BDD est remplie grâce à un script python qui récupère les produits de OpenFoodFacts via son API. Cependant, OpenFoodFacts étant une base de données collaborative, la qualité des informations varie beaucoup d'un produit à l'autre. J'ai donc choisi d'effectuer un premier tri lors du remplissage de la base de données de l'application. En effet, seuls les produits contenant les informations suivantes ont été retenus : Nom, image, catégories, score, code, nutriments.

L'algorithme de recherche d'un substitut pour un produit donné se base principalement sur 2 critères : les catégories dont fait partie le produit et la note de celui (ou le nutri-score).

Dans un premier temps l'algorithme va chercher s'il existe un ou des produits ayant en commun toutes les catégories du produit recherché, ainsi qu'une meilleure note que ce dernier (ou une note de A si le produit recherché est lui-même noté A). S'il y a un résultat, le site affiche le ou les produits trouvés (avec un maximum de 6 substituts). Sinon, il relance la recherche en enlevant la dernière catégorie, jusqu'à trouver au moins un substitut. Les catégories étant le plus souvent listées de la plus générale à la plus restrictive, cette méthode

permet de trouver des substituts relativement cohérents. Cependant, cette méthode a des limites dans le cas où le produit proposé par l'utilisateur dispose de peu de catégories renseignées et que celles-ci sont très générales (ex : aliment à base de végétaux).

De plus, plus la base de données comprend de produits, plus il y a de chance d'avoir un substitut convaincant proposé. La limite de lignes dans une BDD pouvant être déployée sur la version gratuite de Heroku étant de 10 000, les résultats sont malheureusement moins cohérent sur la version en ligne de l'application que sur la version développée en locale.

2.3 Développement Front-End

Bien que le thème du site soit fourni, ce projet nécessitait de se pencher sur la partie « Front » de l'application. En effet, des ajustements étaient à faire sur le thème pour pouvoir introduire les fonctionnalités souhaitées par l'équipe de Pur Beurre.

Il a fallu, par exemple, créer une fonction javascript permettant de sauvegarder les aliments dans les favoris de l'utilisateur sans recharger la page de résultat, ou encore modifier des propriétés CSS pour que l'affichage soit toujours responsive après avoir modifier le template.

La partie la plus délicate de cette étape était s'approprier le code CSS du thème. J'ai dû effectuer de nombreux essais avant de comprendre le rôle de certaines lignes de code et savoir lesquelles modifier pour arriver au résultat souhaité.

2.4 Tests

Afin de tester l'application, j'ai utilisé le module 'test' de Django basé sur le module 'unittest' de la bibliothèque standard de Python.

J'ai donc réalisé des tests unitaires en les séparant en plusieurs catégories. Afin de distinguer les différents types de tests, j'ai choisi de créer deux fichiers :

- test_utils.py : Ce fichier teste les fonctions du fichier « utils.py ». Le fichier « utils.py » contient la fonction de recherche des produits dans la base de données ainsi que celle de recherche d'un substitut.

- test_views.py : Ce fichier contient les tests réalisés sur les vues. Il est lui même divisés en deux classes permettant, pour l'une de vérifier la bonne réception d'un code HTTP 200 lors de l'appel d'une vue avec les arguments attendus, et pour l'autre de vérifier que les informations renvoyées par les vues en fonction des arguments passés sont bien celles que l'on attend.

Les tests réalisés pour ce projet ne couvrent pas toute l'étendue des différents tests pouvant être fait pour ce type d'application. Cependant ceux décrit ci-dessus m'ont semblé être les éléments essentiels à tester.

3. Pistes d'améliorations

Le programme réalisé répond aux attentes de nos clients mais il reste encore perfectible. En effet, nous pourrions envisager d'améliorer celui-ci en développant, par exemple, les fonctionnalités suivantes :

- Les critères de recherche d'un substitut sont assez basique. Il pourrait être intéressant de faire une étude plus poussée sur les règles de nutrition afin d'améliorer l'algorithme d'association des produits.
- Les favoris de l'utilisateur sont présentés comme une simple liste de produit. Il serait envisageable de permettre à l'utilisateur de classer les produits au seins de catégories qu'il a lui même créé (ex : Petit-déjeuner, alternative végétarienne...) voir même de pouvoir partager ses favoris ou consulter ceux d'autres utilisateurs.

L'application web est disponible à l'adresse suivante : <https://purbeurreopc.herokuapp.com/>

Le code de l'application est hébergé sur git-hub : https://github.com/NanroYahel/P8_Creez-une-plateforme-pour-amateurs-de-Nutella