

P8 Pur Beurre

Le projet est disponible à l'adresse suivante : <https://pur-beurre-cc.herokuapp.com>

1 : Planification du projet

Après avoir pris connaissance du [cahier des charges](#), pour planifier le projet, j'ai suivi les étapes décrites dans la [description du projet](#). Les différentes étapes et sous étapes sont listées dans [un tableau Trello](#). J'ai ensuite initialisé un [repo GitHub](#).

2 - Création du projet Django

J'ai tout d'abord suivi les cours sur l'utilisation de Django et le modèle MVT afin de me familiariser avec le framework. J'ai créé le projet, avec pour commencer une application simple, dans laquelle j'avais prévu de tout d'abord mettre uniquement la page d'accueil statique.

3 - La page d'accueil

La page d'accueil est le premier élément du site que j'ai réalisé. En suivant le cahier des charges j'ai commencé par inclure le template Bootstrap proposé. J'ai modifié le CSS inclus dans le template afin de me conformer aux exigences de la charte graphique. L'ensemble des assets concernant l'aspect du site sont dans le dossier *static* de mon application.

4 - Espace utilisateur

Pour la création d'un espace utilisateur et des différentes vues de connexion et création de compte, j'ai fait le choix de créer une nouvelle application dans mon projet. J'ai fait ce choix dans un but de clarté afin de séparer les fichiers concernant les utilisateurs et ceux concernant les aliments. Cette architecture a également l'avantage de permettre la réutilisation de l'application concernant les utilisateurs dans le cas de création d'un autre site. Pour la connexion et la déconnexion de l'utilisateur, j'utilise les vues par défaut de Django, pour la création de compte j'utilise une vue qui fait appel à un formulaire par défaut de Django.

5 - Recherche

Suite au [projet n°5 du parcours](#), j'avais déjà une bonne vision du fonctionnement de l'API Open Food Facts. Après discussion avec mon mentor à propos des limitations de la plateforme Heroku sur la taille des bases de données PostgreSQL (limitation à 10000 éléments stockés en base), j'ai choisi de ne stocker dans ma base de données que les produits enregistrés par les utilisateurs. Ma recherche de produit s'effectue donc via des appels à l'API Open Food Fact et des retours en JSON mis en forme dans les templates. Le seul inconvénient de cette solution par rapport à un stockage de l'ensemble des produits en base de données locale est l'impact sur les performances. Les appels *requests* prennent plus de temps qu'une requête aux modèles. Lorsqu'on recherche un terme, la vue de recherche passe la chaîne de caractères à l'API de recherche Open Food Facts. Le retour des résultats est alors proposé à l'utilisateur qui va pouvoir choisir le produit pour lequel il souhaite avoir des substituts.

6 - Proposition de substituts avec meilleur nutriscore

Lorsque l'utilisateur choisit un produit, mon algorithme va déterminer la catégorie la plus pertinente pour les substitut. Pour cela on utilise la hiérarchie des catégories afin d'avoir la catégorie la plus fine pour un produit donné. Ensuite on recherche les produits dans cette catégorie ayant le meilleur nutriscore, et on affiche à nouveau les résultats obtenus en JSON dans le template. L'affichage de la page de détails d'un produit donné est également faite à l'aide d'un appel à l'API et d'un retour d'informations en JSON.

La fonctionnalité de sauvegarde des produits a également nécessité un choix de solution. Ma première démarche était d'avoir uniquement un modèle Produit pour stocker les informations des produits sauvegardés, dans lequel j'avais ajouté un champs "sauvegardé par" créant une relation 1-N avec le modèle des utilisateurs. Cette solution s'est avéré limitante lorsqu'il m'a fallu afficher les produits sauvegardés par un utilisateur donné, en les triant par date chronologique. Je n'avais pas l'information "date de sauvegarde". J'ai donc créé un nouveau modèle concernant uniquement les sauvegardes, liant un produit à un utilisateur avec une date de sauvegarde.

7 : Mise en ligne sur Heroku

Une fois toutes ces étapes terminées, lorsque le projet était complet en environnement de développement, je l'ai mis en ligne sur Heroku en suivant la documentation pour l'installation, la création du fichier Procfile, et le push du repo en production.

Améliorations possibles

Le projet pourrait être amélioré en ajoutant une recherche locale dans les produits sauvegardés par l'utilisateur. Il serait également intéressant d'indiquer les magasins dans lesquels on peut acheter les produits.