

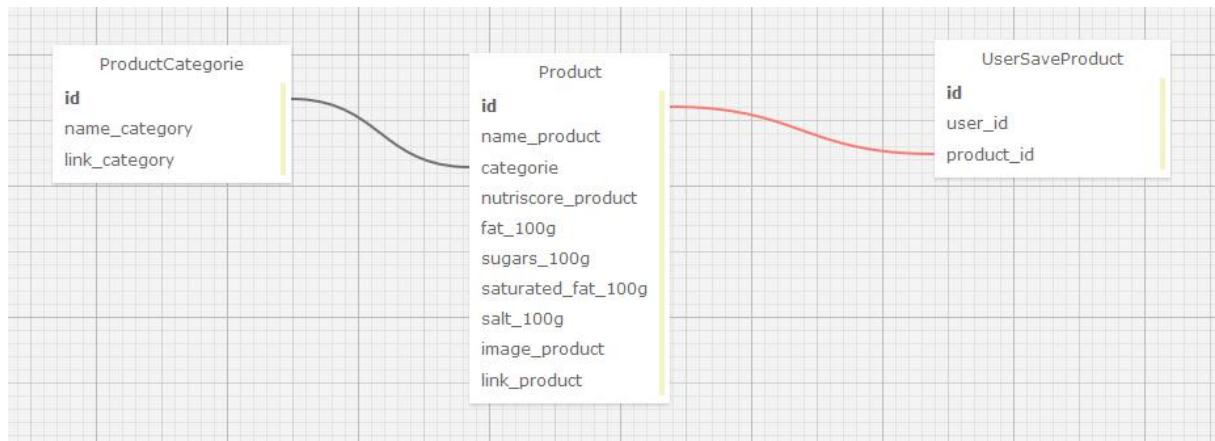
P8 Créer une plateforme pour amateur de Nutella

1. Développement :

J'ai commencé par initialiser un projet en installant le framework Django et Postgresql, puis initialisé le projet avec git. J'ai créé un backlogs sur Trello en fixant les objectifs.

Ensuite j'ai ajouté à mon projet tous les fichiers en suivant le model MVT.

J'ai repris la table du projet 5 et j'ai recréé mon fichier models.py en utilisant les champs adéquats pour obtenir le model suivant :



Une fois le fichier models terminé j'ai exécuté la commande `./manage.py migrate`

Une fois la migration terminée, j'ai créé une fonction qui permet de dumper les informations d'OFF pour les ajouter en base de données.

Après avoir peuplé la base j'ai codé la fonction de recherche des aliments et celle pour afficher le détail du produit sélectionné.

Lorsque ces 2 fonctions ont été fonctionnelles, j'ai ajouté les éléments nécessaires au template.

Ensuite je me suis occupé de la partie création et connexion au compte utilisateur pour pouvoir gérer la sauvegarde des aliments.

J'ai rencontré mon premier problème pour l'affichage des produits sauvegardés d'un utilisateur. La page de sauvegarde affichait bien la mise en forme avec le nombre de produits mais avec uniquement leurs noms.

Après avoir fait le point avec mon mentor, le problème a pu être résolu en ajoutant dans les paramètres « `product_id` » entre le nom de ma table et le nom du champ.

Une fois tout le projet fonctionnel, j'ai effectué les modifications pour pouvoir déployer le programme sur Heroku.

2. Déploiement Heroku :

Après avoir terminé la préparation du projet pour le déploiement, j'ai pushé sur Heroku le projet et j'ai été confronté à plusieurs problèmes.

Le 1^{er} problème était le fait qu'il manquait un fichier css ce qui empêchait Heroku de générer les fichiers statics sur le serveur.

Après avoir réglé le problème de css, j'ai importé le dump de ma base de données avec succès mais un autre problème généra une erreur 500.

Après avoir parcouru plusieurs heures le web, j'ai fini par trouver la solution sur Stackoverflow.

Le problème venait du fait que dans mon fichier settings.py, il manquait la variable `STATIC_URL = '/static/'` avant la condition `if os.environ.get('ENV') == 'PRODUCTION'`

J'ai passé pas mal de temps sur ce projet mais cela m'a permis de voir une infime partie du potentiel de Django !

Site en ligne: <https://pur-beurre-website.herokuapp.com>

Dépot Github: https://github.com/sebpy/P8_pur_beurre

Trello: <https://trello.com/b/eTOl4DSV/pur-beurre-site>