

PARCOURS OPENCLASSROOMS DÉVELOPPEUR D'APPLICATIONS PYTHON

QUENTIN LATHIÈRE

Projet 8 : Plateforme Pur Beurre (Nutellove)

https://github.com/Synkied/OC_Projet-8

Lien vers l'app : <http://nutellove.herokuapp.com/>

Lien vers le kanban : <https://tree.taiga.io/project/synkied-nutellove/kanban>



Le présent document résume la réalisation du 8^e projet (Plateforme Pur Beurre) dans le cadre de mon parcours Développeur d'application - Python.



Table des matières

Définition du projet.....	3
Description.....	3
Fonctionnalités.....	3
Contraintes.....	3
Pré-projet.....	3
Environnement de développement	3
Anticipation des problèmes.....	3
Première difficulté rencontrée.....	3
Solution apportée.....	3
Mise en place du projet.....	4
Démarche choisie	4
Difficulté rencontrée (base de données)	4
Solution apportée.....	4
Difficulté rencontrée (base de données, Heroku).....	5
Solution apportée.....	5
Difficulté rencontrée (vues classes Django).....	5
Solution apportée.....	5
Post projet.....	5
Compétences acquises.....	5
Compétences renforcées	5



Définition du projet

Description

Ce projet consiste à créer une application permettant à ses utilisateurs de chercher un produit et leur proposer une liste de substituts plus sains qu'ils peuvent ensuite enregistrer dans leurs favoris et retrouver plus tard.

Fonctionnalités

- Utiliser Python 3 ;
- Construire l'application avec Django ;
- Pas de requêtes Ajax ;
- Interface responsive (adaptées à tous plusieurs types d'écrans) ;
- Authentification d'un utilisateur et gestion de favoris.

Contraintes

- Versionner son code en utilisant Git et le publier sur Github ;
- Développer dans un environnement virtuel en utilisant Python 3 ;
- Respecter les bonnes pratiques de la PEP 8 ;
- Code écrit en anglais : nom des variables, commentaires, fonctions...

Pré-projet

Environnement de développement

- Machine sous Windows 10. Environnement virtuel via virtualenv ;
- Sublime Text 3 avec plusieurs plugins (Principaux : Anaconda, GitGutter) ;

Anticipation des problèmes

- Construire la base de données ;
- Apprendre à utiliser Django ;
- Déployer sur Heroku.

Première difficulté rencontrée

- Appréhender Django et sa documentation conséquente.

Solution apportée

- Suivre le cours OpenClassrooms et différents autres cours, afin de comprendre la philosophie de Django.



Mise en place du projet

Démarche choisie

Comme demandé, j'ai d'abord écrit les démarches que je pensais effectuer pour ce projet, et l'ordre dans lesquels je pensais les réaliser. Pour me rappeler des tâches à effectuer, j'ai créé un Taiga et me suis fait des User Stories.

J'ai ensuite commencé à **coder directement en initialisant un projet Django** et en suivant le cours OpenClassrooms associé au projet.

Pour **construire la base de données**, j'ai repris mes modèles créés pour le projet 5 du parcours développeur d'application Python et les adaptés à l'ORM de Django.

Pour **alimenter la base de données**, j'ai adapté mon fichier *db_feeding.py* en utilisant la syntaxe de l'ORM de Django (j'avais utilisé Peewee pour le projet 5).

Ce script s'appuie sur un **fichier CSV** (*db_file.csv*) pour fonctionner et ne fait pas appel à l'API OpenFoodFacts, toujours expérimentale.

Pour **construire l'application Django** j'ai d'abord suivi le cours OpenClassrooms, qui m'a permis de bien débiter, et me suis ensuite **orienté vers divers autres cours et tutoriel**, notamment pour la partie authentification et utilisation de formulaires, ou des vues basées sur des classes.

Je me suis ensuite **concentré sur la partie graphique et Bootstrap**. J'ai **suivi de nombreux fils sur StackOverflow et adapté de nombreux snippets** pour arriver à créer un design satisfaisant le cahier des charges.

J'ai décidé d'**écrire les tests de mes fonctionnalités à la fin**, ce qui, avec le recul, n'est peut être pas le plus judicieux, mais qui m'a permis de me **concentrer d'abord sur mon apprentissage** de Django **et sur les divers problèmes rencontrés**, pour ensuite pouvoir écrire sereinement des tests.

Difficulté rencontrée (base de données)

- Construction de la base de données postgresql sous Windows. Un collationnement UTF-8 n'est apparemment pas possible sous Windows (French_france.1252) et le rapprochement des mots avec caractères accentués et non accentués (comme *Nestlé* et *Nestle*) ne se faisait donc pas.

Solution apportée

- Chercher sur Google (oui, encore) et me rendre compte que la seule solution était de développer sous Unix. Mon environnement de travail étant basé sur **Windows et bien rodé**, malgré une installation Linux Mint et la possession d'un MacBook Pro, j'ai préféré accepter le fait que le collationnement utf-8 n'était pas possible sous Windows et **faire avec**, tout en **retenant la leçon**. J'ai **prévu de mettre en place mon environnement de travail sous Linux** prochainement.



Difficulté rencontrée (base de données, Heroku)

- La version gratuite d'Heroku n'accepte pas plus de 10 000 lignes dans la base de données.

Solution apportée

- Réduire le jeu de données à un minimum.



Difficulté rencontrée (vues classes Django)

- Comprendre comment utiliser les **vues génériques fondées sur des classes** et les fonctions **get** et **post** associées.

Solution apportée

- Lire la documentation de Django et plusieurs posts sur StackOverFlow, mais aussi divers autres sources tel que : <https://simpleisbetterthancomplex.com>

Post projet

Compétences acquises

- Utiliser un nouveau framework python : Django ;
- Créer des vues génériques fondées sur des classes.

Compétences renforcées

- Déployer sur Heroku ;
- Utiliser plusieurs librairies tierces et lire des documentations ;
- Structurer un projet ;
- Écrire un Readme ;
- Écrire un makefile ;
- Écrire de la doc.