

PROJET 8- CRÉER UNE PLATEFORME POUR AMATEUR DE NUTELLA

Lien trello : <https://trello.com/b/F7Sw7DbI/projet-8>

Lien GitHub: <https://github.com/yannhamdi/projet8>

Lien Heroku: <https://p8-yh.herokuapp.com/>

Dans le cadre de ce projet, nous avons comme instructions de créer une application qui permettra à l'utilisateur de faire une recherche afin d'obtenir un aliment plus sain selon l'aliment recherché.

L'application devra proposer une interface visuelle, permettre à l'utilisateur de s'inscrire, de se connecter à notre application, d'enregistrer la recherche s'il est connecté et enfin de consulter ses recherches.

Ce document présentera comment j'ai décidé de procéder afin d'offrir une solution adéquate qui satisfera le cahier des charges du projet 8.

Je vous détaillerai donc, les difficultés que j'ai pu rencontrer, les solutions trouvées puis nous survolerons les différentes applications de mon projet.

DIFFICULTES :

Indéniablement, la partie front-end est la partie qui était la plus difficile pour moi, du fait que je ne suis toujours pas à l'aise avec la partie visuelle d'une application, il faut dire que ce n'est que le deuxième projet que nous faisons avec la partie front-end. Mais le client nous a facilité la tâche en nous précisant qu'il voulait une application sur le modèle de « Creative Bootstrap ». En ce qui concerne le côté serveur, l'algorithme est très similaire au projet 5 du parcours. La seule vraie difficulté était donc de prendre en main le Framework « Django ».

SOLUTIONS :

La solution pour ce projet a été de découper le projet en plusieurs applications, chaque application ayant son propre rôle. La première étape que je trouvais nécessaire était de créer ma base de données puisque toute l'application tourne autour de notre base de données Open Food Facts. Ensuite à partir de cette base de données, il fallait créer les vues qui allaient graviter autour de cette base de données. Une fois la logique backend complétée, je me suis concentré sur la partie visuelle de notre projet. Voici les différentes parties de ma présentation.

- **Application « products »**
- **Application « users »**
- **Application « favoris »**
- **Partie « Creative Bootstrap »**
- **Partie « Heroku »**

DESCRIPTION DE L'APPLICATION PRODUCTS :

Comme chaque application créée sur un projet Django, on y retrouve quasiment les mêmes modules (models, views, urls...)

J'ai donc commencé par le module « models.py » qui va nous fournir la structure de notre base de données « OpenFoodFacts », j'ai consulté le cahier des charges ainsi que le site d'openfoodfacts afin de présélectionner les informations nécessaires pour le bon fonctionnement de notre projet.

Une fois, la structure de mon modèle en place, il a fallu à l'aide du module « importapi » récupérer les données via l'api d'Openfoodfacts.

Nous avons donc notre modèle et nos données, j'ai donc créé une commande personnalisée « dbopenfood » afin de remplir cette base de données.

Base de données étant fonctionnelle, j'ai conçu les vues pour mon application « products », on y retrouve une vue pour afficher les informations d'un produit sélectionné., une vue qui affichera le produit recherché et le produit de substitution (nous reviendrons sur cette vue lors de la description de l'application favoris), une vue qui va faire appel au module « managers.py » qui procédera à la requête afin d'obtenir un aliment avec un nutrition grade meilleur que le produit recherché.

Puis, nous avons le module « forms.py » qui proposera un formulaire à l'utilisateur afin qu'il effectue sa recherche.

PROJET 8- CRÉER UNE PLATEFORME POUR AMATEUR DE NUTELLA

Et enfin un module « urls.py » afin de créer les adresses qui permettront d'accéder aux vues.

L'application « products » comme son nom l'indique s'occupera de gérer la partie concernant les aliments.

DESCRIPTION DE L'APPLICATION USERS :

L'application « Users » permet de gérer la partie authentification de notre projet.

Nous avons la partie modèle qui va nous permettre de définir notre base de données à partir du modèle prédéfini de Django.

Puis nous avons la partie formulaire afin d'inviter l'utilisateur à soit entrer ses identifiants soit s'inscrire ainsi qu'une partie validation afin de renvoyer une erreur dans le cas où l'utilisateur fournit des identifiants erronés.

Et finalement, la partie « Views » qui va faire le lien entre l'interface graphique et le backend.

On aura une vue pour l'inscription, une autre pour se connecter, une vue également pour se déconnecter et enfin une vue pour afficher les informations du compte comme demandé dans le cahier des charges.

DESCRIPTION DE L'APPLICATION FAVORIS :

L'application « favoris » va gérer les enregistrements des recherches de l'utilisateur et l'affichage des enregistrements.

J'ai donc également créé un modèle avec une colonne qui va pointer vers l'utilisateur afin d'identifier quel utilisateur a enregistré la recherche, une colonne qui va pointer vers le produit recherché et une colonne vers le produit proposé, ensuite les vues de l'application géreront l'affichage.

Il y a également une contrainte pour cette application, les vues sont accessibles uniquement si l'utilisateur est connecté, j'ai utilisé pour ça le décorateur « @loginrequired() ».

DESCRIPTION DE LA PARTIE CREATIVE BOOTSTRAP :

Voilà, la partie Django backend étant terminé, nous passons à la partie Front-end.

Notre client nous fournit une maquette Bootstrap, en effet il souhaite que son site ait l'aspect visuelle relativement identique à celui de « creative bootstrap ».

Nous récupérons donc le git de creative puis l'adaptions à notre projet. Nous récupérons l'html du fichier index.html que nous modifions selon l'esquisse fournit par notre client.

A partir de là, nous travaillons sur les fichiers « scss » afin de modifier le CSS du projet en adaptant les couleurs, les photos et la police fournit par notre client. Puis avec « npm » nous modulons les modifications.

DEPLOYER SUR HEROKU :

Une fois, notre application fonctionnelle, il nous faut la déployer sur Heroku. Le déploiement est relativement facile, la seule vraie subtilité exporter nos bases de données vers le serveur Heroku. Une fois les bases de données exportées, nous déployons notre application sur heroku.

L'application est enfin en production.

Je souhaite partager mon ressenti pour ce projet, j'ai trouvé la partie backend relativement facile, Django est relativement facile à prendre en main et le tuto officielle est très bien expliqué et bien détaillé.

Avec ce projet, j'ai découvert que j'avais énormément de lacunes au niveau front-end, il y beaucoup d'élément à gérer positionnement, aspect responsive. Je me rends compte que je dois peaufiner mes compétences au niveau CSS et Bootstrap. Je me rassure en me disant que ce n'est que le deuxième projet que l'on prépare au niveau front-end alors que le backend, nous avons déjà eu un parcours bien complet.