

Projet 8 - Réalisation d'un site web « Pur Beurre »

Yohan SOLON

Github : https://github.com/milimi974/OC_Projet_8

Heroku : <https://p8solon.herokuapp.com/>

Analyse

Afin de mener à bien le projet, je l'ai décomposé en plusieurs étapes, me permettant d'avoir une ligne directrice pour sa conception.

La première étape consistait en l'analyse du besoin du client, pour ce faire j'ai parcouru l'ensemble des documents qui ont été fournis, pour avoir une idée générale du projet.

De cette première approche j'ai défini dans les grandes lignes, l'approche que j'allais adapter.

- Intégrer la maquette

- Mettre en place la base de données et les modèles

- Connecter les vues et les données de la base

- Tester le fonctionnement

- Mettre en production

Pour conserver mon analyse et surtout avoir un support de suivi j'ai mis en place un mind map, pour structurer les étapes de mon développement.

En faisant une corrélation entre les fonctionnalités voulues et la maquette j'ai pu avoir une idée précise de l'agencement du site et des étapes de transition entre chaque page. Cette étape est très importante avant le développement car elle permet de voir les incohérences ou de soulever des problématiques pour l'intégration de la maquette.

J'ai créé un projet test pour expérimenter des fonctionnalités répondant aux besoins que j'avais. De par mon expérience dans le développement de site web et de l'utilisation de Framework, j'avais déjà une idée précise concernant direction vers lequel m'orienter.

Pour pouvoir intégrer la maquette, il me fallait savoir comment afficher une page, comment mettre en place un système de layout et passer des variables à la vue. Une fois cette phase d'apprentissage réalisée sur Django, j'ai identifié les parties sur les différentes maquettes et procédé à leurs intégrations.

Dès lors que j'avais mis en place ma structure et testé son fonctionnement, j'y ai intégré le Template fourni par le client en créant les pages prévus dans la maquette et récupéré les ressources de la charte graphique, ajouté le css et js, en y apportant les modifications nécessaires.

Pour cette étape de développement j'ai analysé le type d'information qui allait être servi aux vues et créer une classe desservant des données fictives des champs que j'aurais besoin pour le fonctionnement de l'application, le plus souvent les données issues des ORM respect un format assez identique, c'est-à-dire qu'elle retourne une collection d'objets créer à partir des modèles associés à une table de la base de données, à laquelle on retrouve un ensemble de méthodes. L'objectif dans cette étape était plus de

transmettre des données à la vue pour faire la prise en main du moteur de template de django, pour la manipulation de variable et traiter les répétitions d'information en y intégrant des boucles. Cette étape m'a demandé un temps d'adaptation dans les conventions d'écriture, j'ai découvert des possibilités assez intéressantes pour le traitement, facilitant ainsi grandement mon travail.

Concernant l'intégration, je n'ai pas eu de difficulté particulière, d'un point de vue technique. L'approche adoptée par django restant assez similaire aux autres applications sur lesquels je travaille, sa prise en main en a été facilitée.

Pour la mise en place de la base de données par le biais des migrations, j'ai pris connaissance de la méthodologie à employer. J'ai fait beaucoup de recherches concernant l'utilisation des relations entre modèles sur django, car leur utilisation par django m'avais un peu dérouté. L'idée employée pour concevoir une table en base de données ou de créer son modèle est simple, c'est-à-dire que le système met en œuvre une méthodologie performante et accessible rapidement.

Lorsque j'avais estimé avoir les outils nécessaires pour continuer mon développement, j'ai réalisé un schéma rapide de ma base de données afin d'identifier les champs pour l'écriture de mes tables et définir les relations qui les liait. Pour ce travail je me suis basé sur les données attendues par les fonctionnalités, une connexion utilisateur pour la sauvegarde de ses favoris et une base contenant des aliments classés par catégories.

Mes modèles et mes tables construites je me suis attelé au développement de mes méthodes qui allait servir mes vues en données pour l'affichage et celles permettant de transmettre les données pour un enregistrement en base de données.

Le cas de l'enregistrement des produits étant quant à elle particulière, j'ai créé une commande `manage.py` pour réaliser cette action. Cette phase m'a demandé un peu de réflexion sur l'approche à employer pour une meilleure optimisation.

Une fois mes données sauvegardées en base et mes dernières connexions réalisées, telles que la recherche par mot-clef et la sauvegarde des produits de substitution, je suis passé à l'élaboration de mes tests unitaires.

J'ai mis en place différents tests pour le bon fonctionnement du site, tel que la gestion des vues ou de l'enregistrement de données. C'est lors de cette phase que l'on corrige les oublis ou les erreurs de programmation, permettant ainsi d'avoir une application robuste.

Cette étape m'a permis de pallier à diverses problématiques auxquelles je n'y avais pas songé et m'a permis d'approfondir l'approche des tests unitaire.

Le déploiement sur serveur heroku a été l'étape la plus problématique du projet, des erreurs de par mon code et à la difficulté pour les identifier. Ce qui m'a demandé un temps considérable de recherche et de décomposition du projet pour arriver à isoler le bug.

En conclusion la découverte de django, m'offre une nouvelle vision des possibilités que peut apporter python en termes de solution d'application web, sa simplicité de prise en main et son mécanisme de fonctionnement rendre agréable le développement.

Les erreurs que j'ai rencontrées m'ont poussé un peu plus dans mon apprentissage, me permettant ainsi d'acquérir de nouvelles compétences.