Sandrine SUIRE
sandrinesuire@me.com
11/06/2019

PROJET N°8 : Créez une plateforme pour amateurs de Nutella

github : https://github.com/sandrinesuire/project_8.git
trello : https://trello.com/b/jFdqMzJa
site : https://thebestsubstitute.herokuapp.com/

I. Organisation du travail

A. Initialisation du repository Github :

J'ai initialisé mon repository sur mon compte Github, et afin d'utiliser Trello pour créer les stories et tâches correspondant à l'application, j'ai créé un repository sur bitbucket en important mon repository Github. J'ai ajouter l'url de ce repository bitbucket, au config de git dans mon projet local, pour pousser mes modifications (à chaque git push), aussi bien sur mon repository Github que sur mon repository Bitbucket.

depuis le fichier config de git, dans la section [remote "origin"] j'ai ajouter l'url de mon dossier bitbucket url = https://bitbucket.org/sandrinesuire/project_8.git

B. Rédaction du readme.md

Le fichier README.md est écrit en Markdown. Il reprend les objectifs, description et fonctionnalités attendues de l'application. Il reprend également la description des étapes de travail et des livrables, afin de réaliser l'application. Pour finir, il contient les informations nécessaires à l'installation, au démarrage, ainsi qu'aux tests de l'application.

C. Tableau Trello des stories et tâches de l'application

J'ai transformer en stories les différentes étapes de travail nécessaires à la réalisation de l'application. Les étapes un peu complexe se sont décomposées en tâches, voir en sous-tâches lorsque nécessaire.

II. Créer le projet Django et paramétrer la Base de données

A. Création du projet Django :

J'ai configurer le virtuel environnement, installer django puis créer le projet nutella.

B. Paramétrage de la base de données

Une fois avoir installer psycopg2, j'ai créer la base de données nutella, puis paramétrer les settings afin d'authentifier l'accès à la base depuis le projet, puis migrer la base de données.

C. Ajout de librairies

Certaines librairies sont installées dans le but de faciliter la programmation :

- pip install pylint-django (j'ai ajouté les docstrings manquantes afin de démarrer avec la nouvelle application avec une note de 10/10 : pylint --load-plugins pylint_django substitue)
- pip install django-debug-toolbar

D. Création de l'application

- j'ai tappé dans une console à la racine de mon projet : django-admin startapp substitue, puis ajouter l'application dans les settings du projet nutella.

III. Codage de l'application

A. Création du template de base et de la page d'accueil search.html

- J'ai créer les dossiers pour les templates et static
- J'ai télécharger le thème demandé dans le cahier des charges
- J'ai créer le template de base comprenant les blocs qui se trouvent communs à toutes les pages de l'application (barre de navigation, footer...)
- J'ai créer la page search.html selon le modèle du thème mais en adaptant les chemin avec les statics de l'application
- J'ai placé tous les boutons et liens nécessaire, ainsi que les textes et photos.
- Pour chaque lien et bouton j'ai créer les cibles nécessaire et les templates (et urls) nécessaires à l'application

B. Adaptation des fonctions de remplissage de la base de données

- Le projet 5 comprenait déjà des fonctions d'appels à l'api et les fonctions de remplissage de la base de données. J'ai repris ces fonctions que j'ai adaptée au projet Django.

C. Fonction de recherche

- Le projet 5 comprenait déjà des fonctions de recherche d'articles et de substituts dans la base de données, ainsi que les fonctions de filtres. Je les ai reprises puis adapter à ce projet Django afin qu'elles fonctionnent.

D. Contrib.Auth

- Le projet 5 comprenait déjà le modèle User, je l'ai repris avec les fonctions qui lui étaient attachées, et converti en model Profile contenant un champ User de contrib auth, créant ainsi la page de compte utilisateur et les méthodes de connection, déconnexion, création de compte et enfin adapté la barre de menu pour qu'il s'adapte à la connexion de l'utilisateur.

E. Pages manquantes

- Par le même processus j'ai créer les pages manquantes, en m'appuyant sur les templates de bases. J'ai ajouté les vues correspondantes, et donc créé la page de détails, la page de mes substituts, la page de conditions légales, 404 et 550 pour les pages d'erreurs.

F. Tests :

Une fois la partie code terminée, j'ai créer les tests afin de contrôler chaque méthode pouvant être appelée.

G. Hébergement :

Une fois les tests fonctionnels, j'ai modifié mes paramètres dans les settings.py et dumper ma base de données, créer une nouvelle application sur Heroku, pousser mon code et remplit ma base de données.