

PROJET 11

Améliorez un projet existant en Python

Lien GitHub : <https://github.com/Tony380/Projet11>

Cadre de départ :

L'objectif ici, est de choisir un projet effectué pendant le parcours et de l'améliorer en y ajoutant une ou plusieurs fonctionnalités.

Pour ma part, j'ai choisi de revenir sur le projet 8, qui avait pour but de faire une application web avec le framework Django.

Les améliorations :

J'ai opté pour plusieurs fonctionnalités. Ces dernières sont utilisées dans la majorité des projets liés au développement d'un site web.

Ces fonctionnalités sont les suivantes :

- Une photo de profil par défaut lorsqu'un utilisateur crée son compte
- La possibilité de changer cette photo par la suite
- La modification des données personnelles (adresse mail, nom d'utilisateur, etc...)
- La modification du mot de passe
- La suppression du compte

Toutes ces nouvelles fonctionnalités sont visibles et utilisables dans la page de profil de l'utilisateur.

Photo de profil par défaut :

Le premier pas a été de créer une table « Profile » dans les « models » de l'application « users », qui va contenir la photo de profil. Cette table est reliée à la table User par une relation « un à un ».

J'ai ensuite créé un fichier signals.py qui va se charger d'attribuer la photo de profil par défaut à l'utilisateur nouvellement inscrit.

Cette photo est stockée dans un dossier « media » à la racine du projet. J'ai donc dû rajouter dans les settings du projet, la route qui indique où trouver les photos avec MEDIA_URL et MEDIA_ROOT.

Changement de photo de profil :

Pour qu'un utilisateur puisse modifier sa photo de profil, il faut créer un formulaire et indiquer un champs pour la photo. Puis, dans la méthode servant a renvoyer la page de profil de l'utilisateur, faire une méthode « POST » à cet effet. Enfin, il faut créer le document HTML relatif à ce formulaire. Les photos ajoutées par les utilisateurs sont stockées dans un dossier « profile_pics » dans le dossier « media ».

Modification des données personnelles :

Cette étape est la même que celle pour la modification de la photo de profil. J'ai créé un formulaire avec les champs concernés (nom d'utilisateur, courriel, nom et prénom), puis dans la même méthode « POST » de la fonction renvoyant la page de profil, j'ai ajouté le formulaire de modification des données personnelles. J'ai aussi créé un document HTML à cet effet.

Modification du mot de passe :

Django possède une vue prête à l'emploi. Il suffit donc de rajouter dans les « urls » le nom de la vue en question (« PasswordChangeView »). J'ai juste surchargé la méthode pour lui ajouter un message si le mot de passe a été modifié avec succès et créé un document HTML pour cela.

A noter qu'il faut, dans « l'url », rajouter le paramètre « success_url » pour indiquer une redirection après avoir modifié le mot de passe.

Suppression du compte utilisateur :

Enfin, pour cette étape, il a suffi de créer une fonction permettant de déconnecter l'utilisateur puis de le supprimer de la base de données, ainsi que d'indiquer « l'url » dans urls.py. Cette fonction opère une redirection à la page d'accueil et indique à l'utilisateur, par un message, que son compte a été supprimé avec succès.

Résolution d'un bug :

Pour cette partie, j'ai simulé une faute de frappe dans la vue permettant de rechercher des produits. J'ai ensuite ajouté, au test relatif à cette vue, une vérification supplémentaire.

Enfin, j'ai rédigé les courriels pour le client comme si nous étions dans un cadre professionnel.

Conclusion :

Pas de difficultés particulières à signaler pour ce projet, si ce n'est la recherche de tutoriels sur internet ainsi que les heures passées à rechercher les informations nécessaires dans la documentation de Django.