

Guide d'étapes clés : Développez une application web en utilisant Django

Comment utiliser ce document ?

Ce guide vous propose un découpage du projet en étapes. Vous pouvez suivre ces étapes selon vos besoins. Dans chacune, vous trouverez :

- des recommandations pour compléter la mission ;
- les points de vigilance à garder en tête ;
- une estimation de votre avancement sur l'ensemble du projet (attention, celui-ci peut varier d'un apprenant à l'autre).

Suivre ce guide vous permettra ainsi :

- d'organiser votre temps ;
- de gagner en autonomie ;
- d'utiliser les cours et ressources de façon efficace ;
- de mobiliser une méthodologie professionnelle que vous pourrez réutiliser.

Gardez en tête que votre progression sur les étapes n'est qu'une estimation et qu'elle sera différente selon votre vitesse de progression.

Recommandations générales

Dans le cadre du projet, vous développez une interface utilisateur (« user interface » (UI), en anglais) pour une application web en utilisant le framework de template Django. Ce projet vous permet de maîtriser et de montrer votre créativité en tant que compétence transversale (« soft skill »). Les profils démontrant des capacités créatives sont énormément recherchés par les entreprises et les recruteurs.

- N'hésitez pas à demander des retours sur ce sujet à votre mentor au fur et à mesure du projet.
- Suivez le cours OpenClassrooms [Développez votre créativité](#).

En même temps, gardez à l'esprit que cette application web est un « Minimum Viable Product » (MVP). Par conséquent, lorsque vous exercez votre créativité, veillez à ne pas trop compliquer l'interface utilisateur.

- La créativité ne signifie pas « ajouter beaucoup de détails et de choix personnels ».
- Vous pouvez trouver des éléments de conception à choisir qui n'affectent pas la fonctionnalité.
- Veuillez toujours garder en tête l'utilisateur ! Une expérience utilisateur (« user experience » ou UX) avec une navigation facile est toujours mieux qu'une belle UX avec une navigation compliquée.

Au fur et à mesure du projet, vous devrez prendre en compte les contraintes des personnes en situation de handicap. Assurez-vous de respecter les bonnes pratiques d'accessibilité du référentiel WCAG (Web Content Accessibility Guidelines ; Règles pour l'accessibilité des contenus Web en français) lorsque vous créez des pages pour l'utilisateur.

Étape 1 : Initialisez le projet

10% de progression

Avant de démarrer cette étape, je dois :

- m'être familiarisé avec la programmation orientée objet, les environnements virtuels et Git.

Une fois cette étape terminée, je devrais avoir :

- la page d'accueil par défaut de Django qui s'affiche ;
- mon application Django qui fonctionne dans un environnement virtuel.

Recommandations :

- Suivez les étapes pour initialiser votre projet :
 - Créez un environnement virtuel ;
 - Installez Django via Pip dans l'environnement ;
 - Lancez l'application avec la commande `runserver` de `manage.py`.

Points de vigilance :

- N'oubliez pas d'installer Django dans l'environnement virtuel ;
- Ne faites pas les migrations pour éviter tout problème futur avec le modèle utilisateur ;

Ressources :

- Commencez à suivre le cours OpenClassrooms [Débutez avec le framework Django](#) :
 - Toute la partie 1 'Plan de cours' ;
 - Chapitre 1 de la partie 2 [Configurez un nouveau projet avec l'utilitaire de ligne de commande de Django](#).
- Vous pouvez également lire le cours [Allez plus loin avec le framework Django](#).

Étape 2 : Créez les pages de connexion et d'inscription

30 % de progression

Avant de démarrer cette étape, je dois avoir :

- une plateforme Django fonctionnelle.

Une fois cette étape terminée, je devrais avoir :

- les pages de connexion et d'inscription.

Recommandations :

- Créez un modèle utilisateur personnalisé ;
- Créez un super utilisateur ;
- Vérifiez le fonctionnement de l'admin de Django ;
- Effectuez les migrations ;
- Développez si besoin les vues de connexion, inscription et deconnexion ;
- Créez les templates des vues d'inscription et de connexion ;
- Créez les URL des pages d'inscription et de connexion.

Points de vigilance :

- Attention à ne pas faire de migration avant la création du modèle utilisateur personnalisé ;
- Prenez en compte les contraintes des personnes en situation de handicap. Assurez-vous de respecter les bonnes pratiques

d'accessibilité du référentiel WCAG lorsque vous créez des pages pour l'utilisateur. Par exemple :

- Fournissez des titres et des textes alternatifs pour les images ;
- Donnez un titre explicite pour chaque lien ;
- Évitez une différence de contraste trop faible entre le fond et le contenu.

Ressources :

- Le cours OpenClassrooms [Allez plus loin avec le framework Django](#), notamment la partie 2.
- Le cours OpenClassrooms [Concevez un contenu web accessible](#).
- [Règles pour l'accessibilité des contenus Web \(WCAG\) 2.1 \(le référentiel officielle WCAG 2.1 traduit en français\)](#)
- [Résumé des contraintes WCAG 2.1](#)
- [Guide de référence rapide WCAG](#)

Étape 3 : Créez les modèles

50 % de progression

Avant de démarrer cette étape, je dois avoir :

- une plateforme Django fonctionnelle avec le système d'authentification et d'inscription.

Une fois cette étape terminée, je devrais avoir :

- créé les modèles de billets, de commentaires et de suivi d'utilisateurs.

Recommandations :

- Listez les différents modèles et champs dont vous aurez besoin ;
- Implémentez les modèles ;
- Utilisez l'administration de Django pour tester vos modèles.

Ressources :

- Continuez à suivre le cours OpenClassrooms [Débutez avec le framework Django](#)
 - Partie 2, d'abord le deuxième chapitre [Servez du contenu à l'aide d'une vue](#)
 - Partie 3, d'abord le chapitre [Capturez des données avec des modèles et des champs](#).

Étape 4 : Créez les pages d'ajout, de modification et de suppression des billets et des commentaires

60 % de progression

Avant de démarrer cette étape, je dois avoir :

- les différents modèles de billets et de commentaires.

Une fois cette étape terminée, je devrais avoir :

- les pages d'ajout, de modification et de suppression des billets.

Recommandations :

- Créez la vue, le template et l'URL pour l'ajout de billet ;
- Une fois l'ajout fonctionnel, modifiez la vue pour gérer les modifications ;
- Créez les vues et l'URL pour la suppression d'un billet ;
- Reproduisez le même schéma pour les commentaires.

Points de vigilance :

- Les suppressions doivent se faire via une méthode POST ;
- Les suppressions et modifications ne sont possibles que sur les billets créés par l'utilisateur courant ;
- Comme avant, assurez-vous de respecter les bonnes pratiques d'accessibilité du référentiel WCAG pour les pages que l'utilisateur voit.

Ressources :

- Le reste du cours [Débutez avec le framework Django](#).

Étape 5 : Créez les pages d'abonnement aux utilisateurs

70 % de progression

Avant de démarrer cette étape, je dois avoir :

- une application permettant la création, la modification et la suppression de billets et de commentaires.

Une fois cette étape terminée, je devrais avoir :

- la possibilité de suivre d'autres utilisateurs en fonction de leur nom.

Recommandations :

- Créez la vue, le template et l'URL pour lister les utilisateurs suivis ;
- Créez la vue et le formulaire pour l'ajout d'utilisateur suivi ;
- Créez la vue pour la suppression d'utilisateur suivi.

Point de vigilance :

- Vous pourrez prévoir un message quand l'utilisateur connecté recherche un autre utilisateur qui n'existe pas.

Ressources :

- La [documentation officielle Django sur les liaisons plusieurs à plusieurs](#).

Étape 6 : Créez la page de flux

85 % de progression**Avant de démarrer cette étape, je dois avoir :**

- la possibilité d'ajouter des billets et des commentaires et de suivre des utilisateurs.

Une fois cette étape terminée, je devrais avoir :

- un flux affichant les billets et commentaires autorisés pour l'utilisateur courant.

Recommandations :

- Vérifiez quels billets et commentaires doivent être visibles pour l'utilisateur ;
- Mettez en place la requête et le tri des données ;
- Lisez encore l'appendice du cahier des charges sur les méthodes de tri de deux modèles différents.

Étape 7 : Nettoyez, documentez et publiez votre code

100 % de progression

Avant de démarrer cette étape, je dois avoir :

- une application qui effectue toutes les fonctionnalités attendues dans le cahier des charges.

Une fois cette étape terminée, je devrais avoir :

- le repository GitHub de l'application bien nettoyé, documenté et publié.

Recommandations :

- Relisez votre code, prenez le temps de vous assurer qu'il répond aux meilleures pratiques (docstrings, PEP8, etc.).
 - Vous devrez mettre en place des outils pour nettoyer et embellir votre code, par exemple ``Flake8`` ou ``Black``.
- Vérifiez qu'aucune donnée sensible n'est exposée publiquement (secret, clé privée, informations de connexion à la base de données, etc.)
- Assurez-vous que le repository est complet et permet le déploiement de l'application (fichier requirements, instructions de mise en place de la base de données, etc.).

Points de vigilance :

- Ne jamais livrer du code mal ou non documenté ;
- Assurez-vous que le contenu du fichier README est clair et suffisant pour faire fonctionner l'application en local ;
- N'oubliez pas de vérifier que votre code respecte les bonnes pratiques d'accessibilité du référentiel WCAG dans toutes les pages.

Ressources :

- En relisant votre code, vous pouvez vous référer encore au cours [Devenez un expert de Git et GitHub](#) OpenClassrooms. Ce cours sert de point de référence des bonnes pratiques de Git ;
- Pour vérifier que vous avez suivi les bonnes pratiques de développement PEP-8, référez-vous aux documents ci-dessous (tous rédigés en anglais) :

- [Python PEP-8](#) ;
- Outils permettant une vérification de la PEP-8 :
 - [Documentation de Flake 8](#),
 - [Documentation de Black](#).

Projet terminé !