

Projet Python Django - LiveCampus Juillet 2025

Formateur : Alexandre RASPAUD

Projet 1 Python !

Une boutique en ligne !

Développez une boutique en ligne dans laquelle vos clients pourront:

- Se connecter à leur compte
- Ajouter ou supprimer des articles depuis leur panier
- Payer (pas de paiement à faire ... juste un clic sur “payer” vide le panier et change le statut de la commande)

Votre boutique proposera à minima les vues suivantes pour les clients:

- Page d'accueil avec un listing des articles disponible (il doit être possible de filtrer par catégorie)
- Page d'un article
- Page de connexion pour les clients
- Page panier pour les clients
- Page compte client

Votre boutique proposera à minima les vues suivantes pour les super_users:

- Dashboard (Manager et QuerySet) avec : la liste des commandes qui ont été validées et le CA total
- Recevoir une notification à la validation d'une commande (Signals)

Les articles et catégories d'article seront gérés via Django-Admin pour des “super_users”. Ces deux modèles de données doivent avoir (à minima) une image.

Mais encore ... Créer un Git et y mettre tout votre projet (bien entendu, je dois pouvoir voir vos commits et pas juste un upload en fin de projet). Aussi, développer des classes au lieu de simples méthodes. Pensez également à commenter votre code, pour vous, vos collègues et moi.



Projet 2 Python !

Plateforme de dépôt de candidature

Développer une plateforme sur laquelle des utilisateurs pourront postuler à des annonces pour des emplois. Les employeurs seront des 'super_user' ils pourront créer des annonce via des formulaires uniquement accessible aux super_users et les utilisateurs seront les candidats, ils devront pouvoir:

- Se connecter
- Accéder à leur profil (avec photo) et le modifier
- Avoir accès aux postes non pourvu
- Déposer une candidature
- Créer une entreprise (afin de pouvoir créer des annonces pour cette dernière) - Plusieurs photos doivent pouvoir être uploadées pour présenter la société aux candidats

Votre plateforme devra à minima avoir les vues suivantes pour les candidats:

- Page de connexion
- Dashboard avec candidatures déposées
- Page avec l'ensemble des offres d'emplois non pourvu
- Page du profil (avec photo) candidat de l'utilisateur
- Page contenant le détail d'une annonce, d'où ils pourront postuler

Votre plateforme devra à minima avoir les vues suivantes pour les super_users:

- Dashboard (Manager et QuerySet) avec candidats ayant postulé sur leurs annonces
- Page contenant le détail d'une annonce, d'où ils pourront valider ou invalider les candidatures
- Page du profil (avec photo)
- Notification lorsqu'un candidat postule à une de mes annonces (Signals)

Mais encore ... Créer un Git et y mettre tout votre projet (bien entendu, je dois pouvoir voir vos commits et pas juste un upload en fin de projet). Aussi, développer des classes au lieu de simples méthodes. Pensez également à commenter votre code, pour vous, vos collègues et moi.



Consignes

Votre projet devra comporter les éléments suivants:

- Un [README.md](#) qui expliquera comment lancer et initialiser votre projet localement

- Une API avec un seul endpoint (GET) qui retournera toutes les commandes ou candidatures qui ont été réalisées dans la journée sous format CSV

- Ajoute un bouton dans Django-Admin pour télécharger le CSV disponible par API

- Ajoute un bouton dans Django-Admin pour importer des commandes ou candidatures depuis un fichier CSV

Optionnel (aucun impact sur la note) :

- Mettre en place un cache avec un Docker ValKey (ou Redis)

Projet Python - Notation

Projet fonctionnel : 4

Structure des templates HTML : 1,5

Utilisation de CSS (ou librairie) : 1,5

Logique d'implémentation des modèles : 2

Respect de la structure Django : 3

Qualité du code : 2

Utilisation des classes : 3

Utilisation des QuerySet & Manager Django : 1,5

Utilisation des Signals Django : 1,5

Bonus : Création et utilisation d'un Docker-Compose avec Postgres : 1

