Mission

Si vous avez commencé ce projet avant le 31/10/2023, vous avez commencé votre travail sur **ce projet archivé**. Vous pouvez continuer l'existant ou recommencer avec ce nouveau projet, dont les modifications s'alignent avec les retours étudiants.

En outre, si vous avez commencé ce projet avant le 29/06/2023, vous avez commencé votre travail sur **ce plus ancien projet archivé**. Vous pouvez continuer l'existant ou recommencer avec ce nouveau projet.

Après un entretien réussi, vous décrochez le poste de Lead Développeur Python pour la jeune start-up LITRevu. Son objectif est de commercialiser un produit permettant à une communauté d'utilisateurs de publier des critiques de livres ou d'articles et de consulter ou de solliciter une critique de livres à la demande.

Récemment, LITRevu a réussi à lever des fonds et a donc le feu vert pour développer sa nouvelle application web.

Vous recevez un mail d'Alix, le directeur technique de la société :

De : Alix
À : Moi
Objet : Nouvelle appli web

Bonjour et bienvenue dans l'équipe ! J'espère que ton intégration se passe bien.

1. la publication des critiques de livres ou d'articles ;
2. la demande des critiques sur un livre ou sur un article particulier ;
3. la recherche d'articles et de livres intéressants à lire, en se basant sur les critiques des autres.

Nous cherchons à mettre en place une application web pour notre MVP (*minimum viable product,* ou produit viable minimum). Je sais que tu as beaucoup d'expérience comme développeur Python, je pense que Django sera un framework idéal pour intégrer cette application.

Je te recommande d'abord de consulter le cahier des charges, que tu trouveras ci-joint. Il détaille les besoins fonctionnels et techniques du site.

Notre UX designer a élaboré des wireframes (en pièce jointe). La conception du site doit correspondre aux wireframes fournis, mais n'hésite pas à y ajouter quelques touches personnelles ; il s'agit surtout d'un guide pour la mise en page. Veille à suivre les bonnes pratiques d'accessibilité du référentiel WCAG (Web Content Accessibility Guidelines ; Règles pour l'accessibilité des contenus Web en français) afin que le site convienne bien aux utilisateurs avec un handicap.

J'ai également conçu un schéma de base de données, que j'ai joint à ce mail, couvrant toutes les fonctionnalités requises pour le site. Parce que tu viens de nous rejoindre, je t'ai aidé à démarrer en créant un fichier models.py et en remplissant pour le modèle de révision ('Review').

À ce stade, pas besoin de te préoccuper du déploiement du site. Lorsque tu auras terminé, envoie-moi un lien vers le **repository GitHub** contenant tous les éléments de façon à ce que je puisse exécuter l'application en local :

- un fichier README.md contenant les instructions d'installation ;
- le fichier SQLite (le système de base de données en Django) avec des données de test ainsi que les informations de connexion au site afin que je puisse l'utiliser et le tester ;
- le code de l'application.

Également, envoie-moi un export au format PDF de la veille.

Bon courage pour le projet !
Alix
CTO
LITRevu

Pièces jointes :

- Cahier des charges
- Wireframes : HTML / PDF
- Schéma de la base de données : HTML / PDF
- models.py

Alix vous a fourni tout ce dont vous avez besoin pour développer l'application web Django. Vous êtes donc prêt à vous lancer !

Étapes

Étape 0 : Recommandations générales - Lisez avant de commencer

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.

- 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

Prérequis

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

Résultat attendu

- 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

Prérequis

- une plateforme Django fonctionnelle.

Resultat attendu

- 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 déconnexion.

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

Prérequis

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

Résultat attendu

- 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

Prérequis

- les différents modèles de billets et de commentaires.

Résultat attendu

- 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

- 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

Prérequis

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

Résultat attendu

- 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

Prérequis

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

Résultat attendu

- 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

Prérequis

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

Résultat attendu

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

Suivant