Formation OpenClassrooms
Développeur d'application – Python

Projet N° 9
Développez une application Web en utilisant Django

Powered By : EL-WALID EL-KHABOU

Guide des Étapes clés pour créer ce projet

Planification et conception:

- Analyser le cahier des charges pour comprendre les besoins et les exigences.
- Concevoir un schéma de base de données pour les modèles et les relations (fourni).
- Créer des wireframes pour les interfaces utilisateur (Fournis).
- Planifier l'architecture du projet, y compris la division en applications (par exemple, authentication, blog...).

Configuration initiale:

- Installer Django et initialiser un nouveau projet avec **django-admin startproject litreview**.
- Configurer les paramètres initiaux dans settings.py (bases de données, middleware, etc.).
- Initialiser un dépôt Git et effectuer le premier commit.

Application d'authentification:

Créer une nouvelle application avec **python manage.py startapp authentication**.

- Définir le modèle Profile pour étendre le modèle utilisateur par défaut.
- Créer des formulaires pour l'inscription et la connexion.
- Définir les vues pour l'inscription, la connexion et la déconnexion.
- Configurer les URL pour l'authentification.
- Créer des templates pour les vues d'authentification.
- Ajouter des styles CSS pour l'authentification.

Application de blog:

Créer une nouvelle application avec **python manage.py startapp blog**.

- Définir les modèles Ticket et Review.
- Créer des formulaires pour ajouter/modifier les billets et les critiques.
- Définir les vues pour afficher, créer, modifier et supprimer des billets et des critiques.
- Configurer les URL pour le blog.
- Créer des templates pour les vues du blog.
- Ajouter des styles CSS pour le blog.

Fonctionnalités avancées:

Implémenter la fonctionnalité pour suivre/désabonner d'autres utilisateurs.

- Ajouter une vue de flux pour afficher les billets et critiques des utilisateurs suivis.
- Implémenter des filtres personnalisés ou des tags de template si nécessaire (comme dans blog_extras.py).

Tests et sécurité (Non effectués) :

Écrire des tests unitaires pour les modèles, les vues et les formulaires.

- Configurer les paramètres de sécurité dans settings.py (par exemple, SECRET_KEY, DEBUG, ALLOWED_HOSTS).
- Tester les vulnérabilités courantes (injections SQL, scripts intersites, etc.).

Déploiement et documentation:

Préparer le projet pour le déploiement (configurer STATICFILES_STORAGE, MEDIA_URL, etc.).

- Rédiger un README.md détaillé avec des instructions pour la configuration, l'exécution et le déploiement.
- Créer un fichier requirements.txt avec toutes les dépendances nécessaires.
- Déployer l'application sur un serveur de production (non-effectué).
- Configurer un système de sauvegarde pour la base de données (non-effectué).

Améliorations et maintenance:

Surveiller les performances de l'application et optimiser si nécessaire.

- Mettre à jour régulièrement les dépendances pour garantir la sécurité.
- Recueillir les commentaires des utilisateurs et apporter des améliorations.

- Implémenter des fonctionnalités supplémentaires en fonction des besoins.
- Effectuer des tests d'accessibilité pour garantir la conformité WCAG.
- Documenter toutes les mises à jour et les changements dans le journal des modifications.
- Planifier des revues de code régulières pour maintenir la qualité du code.
- Assurer une communication régulière avec les parties prenantes pour s'assurer que l'application répond toujours aux besoins.

Ces étapes fournissent une vue d'ensemble de la manière dont j'ai élaboré ce projet depuis le début. Bien sûr, en fonction des besoins spécifiques et des circonstances, certaines étapes peuvent nécessiter plus de détails ou des ajustements.

Nous allons maintenant refaire ce planning mais en entrant plus dans les details :

1. Préparation et Planification :

- Analyser le cahier des charges pour déterminer les fonctionnalités nécessaires.
- Établir un plan d'architecture pour le projet.
- Concevoir un schéma de base de données pour les modèles et leurs relations.
- Créer des wireframes pour les interfaces utilisateur.

2. Configuration initiale :

- Installer Django et initialiser un nouveau projet avec **django-admin startproject litreview**.
- Configurer les paramètres initiaux dans litreview/settings.py (bases de données, middleware, etc.).
- Initialiser un dépôt Git, ajouter les fichiers initiaux et effectuer le premier commit.

3. Création de l'application d'authentification :

- Initialiser l'application d'authentification avec **python manage.py startapp authentication**.
- Configurer l'application dans litreview/settings.py en ajoutant authentication à INSTALLED_APPS.

4. Définition des modèles pour l'authentification :

- Créer le modèle Profile dans authentication/models.py.
- Exécuter les migrations pour créer la table dans la base de données.

5. Création des formulaires pour l'authentification :

- Définir SignUpForm et LoginForm dans authentication/forms.py.

6. Définition des vues pour l'authentification :

- Créer des vues pour la connexion, l'inscription, la déconnexion, etc. dans authentication/views.py.

7. Configuration des URL pour l'authentification :

- Ajouter les URL pour chaque vue dans authentication/urls.py.

8. Création des templates pour l'authentification :

- Créer des modèles pour chaque vue, tels que login.html, signup.html, etc. dans le dossier authentication/templates/authentication/.

9. Stylisation pour l'authentification :

- Ajouter des styles CSS spécifiques à l'authentification dans : authentication/static/authentication/css/styles.css.

10. Tests pour l'authentification :

- Commencer les tests de manipulation d'authentification et de création d'utilisateurs apres avoir activer le serveur par la commande **python manage.py runserver**

11. Création de l'application de blog :

- Initialiser l'application de blog avec python manage.py startapp blog.
- Configurer l'application dans litreview/settings.py en ajoutant blog à INSTALLED_APPS.

12. Définition des modèles pour le blog :

- Créer les modèles Ticket et Review dans blog/models.py.
- Exécuter les migrations pour créer les tables correspondantes.

13. Création des formulaires pour le blog :

- Définir les formulaires nécessaires pour le blog dans blog/forms.py.

14. Définition des vues pour le blog :

- Créer des vues pour afficher, créer, modifier, et supprimer des billets et des critiques dans blog/views.py.

15. Configuration des URL pour le blog :

- Ajouter les URL pour chaque vue dans blog/urls.py.

16. Création des templates pour le blog :

- Créer des modèles pour chaque vue du blog dans le dossier blog/templates/blog/.

17. Stylisation pour le blog :

- Ajouter des styles CSS spécifiques au blog dans blog/static/blog/css/styles.css.

18. Tests pour le blog :

- Commencer les tests de manipulation de fonctionnalités du blog dans blog après avoir activer le serveur par la commande **python manage.py runserver.**

19. Intégration des ressources statiques :

- Intégrer les images, logos et autres ressources statiques nécessaires dans les dossiers static des applications.

20. Optimisation et Sécurité :

- Configurer les paramètres de sécurité dans litreview/settings.py.
- Tester les vulnérabilités courantes.

21. Documentation :

- Rédiger un README.md détaillé.
- Documenter le code source pour faciliter la maintenance et la collaboration.

22. Déploiement :

- Préparer le projet pour le déploiement.
- Configurer les paramètres de production dans litreview/settings.py.
- Déployer l'application sur un serveur de production.

23. Maintenance et Mises à jour :

- Surveiller les performances de l'application.
- Mettre à jour régulièrement les dépendances.
- Implémenter des fonctionnalités supplémentaires en fonction des retours des utilisateurs.

24. Tests d'accessibilité :

- Effectuer des tests d'accessibilité pour garantir la conformité WCAG.

25. Optimisation de la base de données :

- Surveiller les performances de la base de données.
- Optimiser les requêtes et les indices si nécessaire.

26. Intégration continue :

- Configurer un système d'intégration continue pour automatiser les tests et le déploiement.

27. Feedback et Améliorations :

- Recueillir les commentaires des utilisateurs.
- Apporter des améliorations en fonction des retours.

28. Formation et Support :

- Fournir une formation aux utilisateurs finaux.
- Mettre en place un système de support pour aider les utilisateurs en cas de problèmes.

29. Backup et Restauration :

- Configurer un système de sauvegarde pour la base de données.
- Tester régulièrement la restauration pour s'assurer que les sauvegardes fonctionnent correctement.

30. Surveillance et Alertes :

- Mettre en place des outils de surveillance pour suivre l'état de l'application.
- Configurer des alertes pour être informé en cas de problèmes.