



LITRevu

**Développez une application Web en
utilisant Django**

Développez une application web en utilisant Django
Projet 9 : Développez une application web en utilisant Django
Peggy Marillier

Introduction

A circular graphic with the text "DÉVELOPPEZ UNE APPLICATION WEB EN UTILISANT DJANGO" written around its perimeter in a light beige color. The word "Introduction" is centered over this graphic in a large, white serif font. A thin vertical line extends downwards from the bottom of the circular graphic.

Ce projet a pour objet le développement d'une interface utilisateur pour une application web en utilisant le Framework Django.

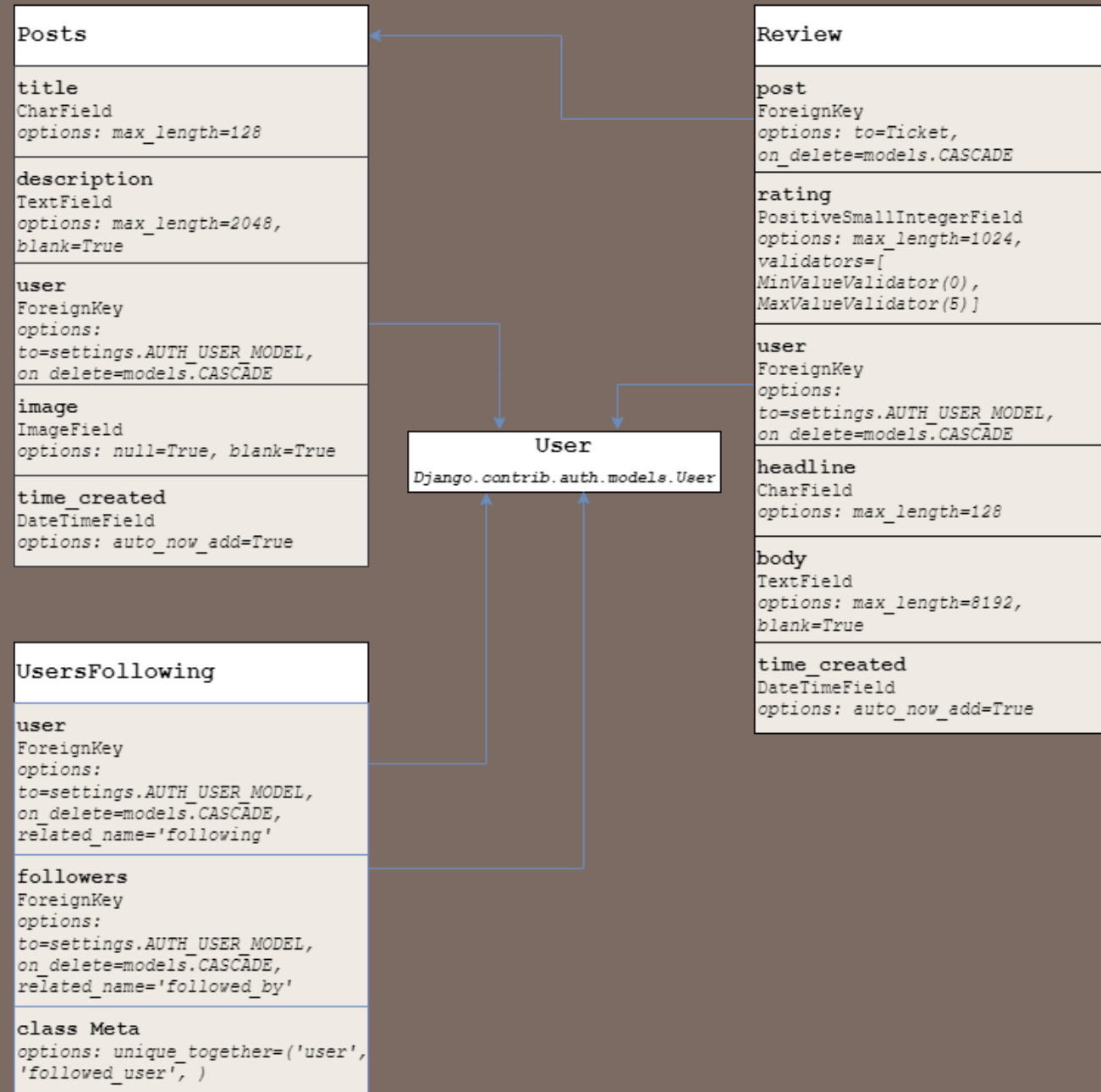




Sommaire

1. Diagramme UML du projet
2. Fonctionnalités
3. Détails du code
4. Bonnes pratiques pour l'accessibilité
5. Difficultés rencontrées
6. Améliorations envisagées

Diagramme UML de base de l'application LITRevu



Fonctionnalités



Connexion

L'utilisateur doit pouvoir se créer un compte, puis se connecter avec un compte existant.



Reviews

L'utilisateur doit pouvoir publier des reviews/avis littéraires.

Celle-ci peut être indépendante ou être une réponse à un Post



Post

L'utilisateur doit pouvoir publier une demande de review ou article



Relation de suivi

L'utilisateur doit pouvoir suivre d'autres utilisateurs et leurs publications. Il doit pouvoir également mettre fin à ce suivi ou bloquer.

Modèle d'architecture

Le Framework Django est conçu autour du modèle d'architecture MVT.

- **M** pour Model : représente la structure des données et la logique métier. Il gère la base de données, en définissant les tables et les relations entre elles.
- **V** pour View : contient la logique qui traite les requêtes utilisateur et interagit avec le modèle. Elle récupère les données nécessaires et les renvoie sous forme de réponse.
- **T** pour Template : définit la présentation des données. C'est la partie qui génère le HTML final à partir des données transmises par la vue.

J'ai donc respectée cette architecture pour créer cette application.

Détails du code

The Django logo, featuring the word "django" in a stylized, lowercase font with a green and yellow color scheme.

Mon code :

Les modèles :

En plus de User (indispensable pour la création d'utilisateurs et l'authentification), j'ai créé les modèles Review et Post pour les diverses publications possibles que pourra faire l'utilisateur.

Un modèle UsersFollowing a également été créé pour gérer la relation de suivi entre utilisateurs.

Les vues :

J'ai basé les vues sur des classes, ce qui garantit une meilleure organisation du code, un héritage des classes natives de Django (comme par exemple LoginRequiredMixin). Le code est ainsi plus facile à comprendre et à maintenir.

Les templates :

Plusieurs fichiers html permettront d'afficher sur la navigateur de l'utilisateur les différents éléments et pages que contient le site.

Les formulaires :

Différents formulaires sont présents pour inviter l'utilisateur à la saisie d'informations.

Accessibilité

Points de vérification de la conformité WCAG :

- . Contraste des couleurs
- . Navigation cohérente et structure logique
- . Compatible avec les technologies d'assistance

Grâce aux outils en ligne (voir image ci-contre) et à l'outil développeur intégré à Chrome, j'ai pu vérifier que les couleurs choisies étaient bien conformes et permettaient une lisibilité suffisante pour les personnes mal voyantes.

Chaque image présente sur le site contient un texte alternatif interprétable en audio par les technologies d'assistance.

Concernant la navigation, j'ai essayé de la rendre la plus intuitive possible, avec un menu toujours visible sur le côté gauche de la page.

Les URLs sont nommées d'après l'utilité de la page html qu'elles concernent (ex : create_review.html est la page qui sert à créer une nouvelle review, etc...) ce qui permet au site d'être structuré de façon logique.

Contrast Checker

[Home](#) > [Resources](#) > Contrast Checker

Foreground
Hex Value
#F4ECE4
Color Picker
Alpha 1
Lightness

Background
Hex Value
#4D3D33
Color Picker
Lightness

Contrast Ratio

8.84:1

[permalink](#)

Normal Text

WCAG AA: **Pass**

WCAG AAA: **Pass**

The five boxing wizards jump quickly.

Large Text

WCAG AA: **Pass**

WCAG AAA: **Pass**

The five boxing wizards jump quickly.

Graphical Objects and User Interface Components

WCAG AA: **Pass**



Text Input



Difficultés rencontrées

Assimilation du modèle MVT et utilisation de Django :

La première difficulté a été de bien comprendre le fonctionnement du modèle MVT, les interactions entre chaque partie.

Pour m'aider j'ai, en plus des cours Openclassrooms, suivi les tutoriels du site officiel Django.

Mise en place du lien de suivi entre utilisateurs :

Ce point particulier m'a donné du fil à retordre. L'objectif était de permettre à un utilisateur de suivre et d'interagir avec d'autres membres de la plateforme, tout en gérant les relations complexes comme le blocage ou le désabonnement.

Ces difficultés m'ont permis d'améliorer mes compétences en manipulation de relations de base de données complexes et en gestion de la logique métier au sein d'une application Django.

Voici les améliorations que j'ai envisagées pour cette application Web :

- Ajouter une page de profil des utilisateurs suivis affichant leurs publications (reviews, posts, etc.), permettrait d'enrichir l'expérience utilisateur. Les visiteurs pourraient non seulement suivre les activités d'un utilisateur, mais aussi accéder directement à son contenu dans une page dédiée.
- Ajouter une barre de recherche avancée qui permettrait de trouver des utilisateurs, des reviews ou des posts par mot-clé, tags ou catégories pourrait faciliter la navigation et la découverte de nouveaux contenus.

Améliorations envisagées