

Projet - Plate-forme de jeu avec Django

Objectif

Créer une plate-forme web dédiée au jeu de Go, permettant aux membres de résoudre des problèmes, jouer contre d'autres joueurs et suivre leur progression. L'application doit être développée avec Django et utiliser une base de données relationnelle.

Contexte

Un club de jeu souhaite proposer une plate-forme en ligne pour le jeu de Go. L'objectif est de permettre aux membres de résoudre des problèmes (tsumego), de jouer contre d'autres joueurs et de consulter des parties de Go. La plate-forme comportera aussi un espace pour l'administration et un tableau de bord.

Fonctionnalités principales

1. **Inscription et profil utilisateur** : Les utilisateurs peuvent s'inscrire, créer un profil et suivre leur progression.
2. **Jouer une partie** : Les membres peuvent jouer contre un script automatique.
3. **Page d'accueil** : Informations sur le jeu de Go et sur la plate-forme.
4. **Tableau de bord** : Les administrateurs peuvent consulter des statistiques sur les parties.

Contraintes

1. L'application doit être développée avec Django.
2. Les données doivent être stockées dans une base de données relationnelle (autre que SQLite).
3. L'API doit suivre les normes OpenAPI.

Libertés

Vous pouvez utiliser toutes les ressources et bibliothèques tierces nécessaires pour réaliser le projet.

Objectifs

1. Développer l'application web avec les fonctionnalités mentionnées.
2. **Tests** : Créer des tests unitaires et fonctionnels pour assurer la qualité du code.
3. **Base de données** : Configurer et utiliser une base de données relationnelle.
4. **Accessibilité** : S'assurer que l'application respecte les normes d'accessibilité.
5. **Sécurisation** : Garantir la sécurité des données des utilisateurs.

Cahier des charges

Fonctionnalités essentielles

- **Inscription** : Permettre l'inscription d'un utilisateur avec gestion des erreurs.
- **Jouer une partie** : Les utilisateurs peuvent jouer contre un script.
- **Profil utilisateur** : Afficher le profil et la progression de l'utilisateur.
- **Tableau de bord admin** : Afficher les statistiques pour l'administration.

Exigences non fonctionnelles

- **Accessibilité** : Veiller à une bonne accessibilité de l'interface.
- **Sécurisation des données** : Protéger les informations personnelles des utilisateurs.

Formats

- **Problèmes** : Format JSON pour les problèmes de Go.
- **Parties** : Format SGF pour les parties de Go.

Rendu final

Une archive ZIP contenant :

- L'application Django fonctionnelle
- Les tests unitaires et fonctionnels
- La documentation technique
- La charte graphique de l'application

Backlog

1. Installation de Django et configuration de la base de données
2. Développement des modèles pour les utilisateurs, les problèmes et les parties
3. Création des vues pour résoudre des problèmes, jouer des parties et gérer le profil utilisateur
4. Mise en place de l'authentification et des permissions utilisateur
5. Création de l'interface d'administration
6. Tests et déploiement

Compétences à valider

- Conception d'une base de données relationnelle
- Développement d'applications web avec Django
- Gestion de l'authentification et des permissions
- Rédaction de tests unitaires et fonctionnels

ANNEXES - Ressources

- Interface de jeu de Go (JS) : <http://wgo.waltheri.net/documentation>
- Collections de parties de Go au format SGF :
<https://homepages.cwi.nl/~aeb/go/games/games/>
- Répertoire de problèmes de Go au format JSON :
<https://github.com/sanderland/tsumego/tree/master>