

Documentation du Projet Assortie

1. Introduction

Le projet de gestion d'événements sociaux est une application web développée avec Django. Son objectif est de permettre aux utilisateurs de créer, gérer et participer à des événements au sein de groupes d'amis. Les fonctionnalités incluent la gestion de profils, l'ajout d'amis, la création de groupes, la proposition et la gestion de sorties, ainsi que la possibilité de voter pour des dates proposées pour des sorties.

2. Installation et Configuration

Prérequis :

- Python 3.8 ou supérieur
- Django 4.2 ou supérieur
- Un serveur web (comme Gunicorn pour la production)
- Un serveur de base de données (SQLite pour le développement, PostgreSQL pour la production recommandée)

Étapes d'installation :

1. Cloner le dépôt du projet :

```
git clone <URL_DU_DEPOT>
cd sortie_project-main
```

2. Créer et activer un environnement virtuel :

```
python -m venv env
source env/bin/activate # (SousWindows,utilisez`env\Scripts\activate`)
```

3. Installer les dépendances :

```
pip install -r requirements.txt
```

4. Configurer les paramètres de l'application :

- Copier le fichier `settings.py.example` en `settings.py`
- Mettre à jour les configurations (base de données, email, etc.)

5. Appliquer les migrations :

```
python manage.py migrate
```

6. Créer un superutilisateur :

```
python manage.py createsuperuser
```

7. Lancer le serveur de développement :

```
python manage.py runserver
```

3. Utilisation

3.1. Gestion des Profils :

- **Création de compte** : Les utilisateurs peuvent créer un compte via la page d'inscription.
- **Édition de profil** : Les utilisateurs peuvent éditer leur profil pour ajouter une photo, des informations personnelles, etc.
- **Ajout d'amis** : Les utilisateurs peuvent rechercher et ajouter des amis.

3.2. Gestion des Groupes :

- **Création de groupes** : Les utilisateurs peuvent créer des groupes d'amis.
- **Ajout de membres** : Les créateurs de groupes peuvent ajouter des membres à leurs groupes.

3.3. Propositions de Sorties :

- **Proposition de sorties** : Les utilisateurs peuvent proposer des sorties avec des descriptions, des lieux et des dates.
- **Vote pour les dates** : Les membres du groupe peuvent voter pour les dates proposées pour les sorties.

3.4. Participation aux Sorties :

- **Participation** : Les utilisateurs peuvent indiquer s'ils participent ou non à une sortie.
 - **Annulation** : Les utilisateurs peuvent annuler leur participation.
-

4. Modèles de Données

4.1. Profile :

- **User (OneToOneField)** : Lien avec le modèle User de Django.

- Friends (ManyToManyField) : Liste des amis.
- Photo (ImageField) : Photo de profil.
- Autres champs supplémentaires.

4.2. GroupeAmis :

- nom (CharField) : Nom du groupe.
- createur (ForeignKey) : Créateur du groupe.
- membres (ManyToManyField) : Membres du groupe.
- administrateur (ForeignKey) : Administrateur du groupe.

4.3. SortieProposee :

- groupe (ForeignKey) : Groupe lié à la sortie.
- nom (CharField) : Nom de la sortie.
- description (TextField) : Description de la sortie.
- date (DateTimeField) : Date de la sortie.
- lieu (CharField) : Lieu de la sortie.
- createur (ForeignKey) : Créateur de la sortie.
- participants (ManyToManyField) : Participants à la sortie.

4.4. Participation :

- sortie (ForeignKey) : Sortie liée à la participation.
- membre (ForeignKey) : Membre participant.
- vient (BooleanField) : Statut de participation.

4.5. PropositionSortie :

- groupe (ForeignKey) : Groupe lié à la proposition.
- nom (CharField) : Nom de la proposition.
- description (TextField) : Description de la proposition.
- lieu (CharField) : Lieu de la proposition.
- createur (ForeignKey) : Créateur de la proposition.
- dates_proposees (ManyToManyField) : Dates proposées pour la sortie.

4.6. Vote :

- utilisateur (ForeignKey) : Utilisateur ayant voté.
- date_proposee (ForeignKey) : Date proposée pour la sortie.
- vote (BooleanField) : Vote (oui/non).

5. Tests

Les tests sont implémentés pour vérifier les fonctionnalités principales de l'application. Ils incluent des tests pour la création de profils, la gestion des groupes, les propositions et participations aux sorties, et les votes.

Commandes pour exécuter les tests :

```
python manage.py test
```

6. Conclusion

Le projet de gestion d'événements sociaux offre une plateforme robuste et extensible pour organiser des sorties et des événements avec des amis. Grâce à l'utilisation de Django, le projet bénéficie d'une architecture solide et de nombreuses fonctionnalités intégrées, ce qui facilite le développement et la maintenance de l'application.

Axes d'amélioration futurs :

- Intégration de notifications en temps réel pour les votes et les participations.
- Amélioration de l'interface utilisateur pour une meilleure expérience.
- Migration vers une base de données plus robuste pour gérer un volume plus élevé d'utilisateurs et de données.