

## **Application Super Bowl Bets - Documentation Technique**

### **Réflexions initiales technologiques sur le sujet**

J'ai développé une application d'événements sportifs appelée Super Bowl Bets. Cette application permet aux utilisateurs de gérer leurs paris sur les matchs du Super Bowl. Les utilisateurs peuvent s'inscrire, se connecter, consulter les informations sur les équipes et les joueurs, accéder aux détails des matchs et placer des paris sur leurs équipes préférées.

### **Configuration de votre environnement de travail**

Pour configurer l'environnement de développement de Super Bowl Bets, suivez ces étapes :

1. Clonez le dépôt du projet depuis GitHub.
2. Installez Python et Django.
3. Créez un environnement virtuel et activez-le.
4. Installez les dépendances requises à l'aide de pip.
5. Configurez la base de données.
6. Exécutez les migrations de la base de données en utilisant le script ``manage.py`` de Django.

### **Modèle conceptuel de données (MCD)**

Le modèle de données de l'application comprend les entités suivantes :

1. ``Équipe`` : Représente une équipe sportive participant aux matchs du Super Bowl.
2. ``Joueur`` : Représente un joueur associé à une équipe spécifique.
3. ``Match`` : Représente un match du Super Bowl avec des détails tels que l'équipe à domicile, l'équipe à l'extérieur, le score, les commentaires, la météo et les côtes.
4. ``Pari`` : Représente le pari d'un utilisateur sur un match particulier, incluant l'équipe sur laquelle il parie et le montant du pari.

### **Explication de votre plan de test**

Afin de garantir la qualité et la fiabilité de l'application Super Bowl Bets, j'ai mis en place un plan de test complet qui comprend les stratégies suivantes :

1. **Tests unitaires:** Écrire des tests unitaires pour les fonctions, méthodes et modèles individuels en utilisant le framework de test intégré de Django.
2. **Tests d'intégration:** Tester l'intégration des différents composants, tels que l'application `users` et l'application `superapp`.
3. **Tests de bout en bout:** Effectuer des tests de bout en bout de l'ensemble du flux de travail de l'application, y compris l'inscription des utilisateurs, la connexion, la gestion des profils et la passation de paris.
4. **Tests automatisés:** Automatiser les tests pour garantir des résultats cohérents et faciliter la maintenance.
5. **Données de test:** Créer des données de test pour chaque scénario de test afin d'obtenir des résultats cohérents et reproductibles.
6. **Couverture des tests:** Viser une couverture des tests élevée pour s'assurer que la plupart des parties de l'application sont testées.
7. **Intégration continue:** Intégrer les tests dans le processus de développement en utilisant des outils d'intégration continue tels que Jenkins ou GitLab CI.

En suivant ce plan de test, nous pouvons garantir que l'application Super Bowl Bets fonctionne correctement et répond aux exigences attendues.