1. Tic Tac Bet

J'ai choisi de construire un Tic-Tac-Toe ancré dans l'univers du pari. Le principe : chaque partie peut être assortie d'une mise en monnaie virtuelle. Les joueurs peuvent soit créer une partie et attendre un adversaire, soit rejoindre directement depuis un lobby les parties proposées par d'autres utilisateurs — un peu comme des salons de paris ouverts.

Pour démarrer, chaque utilisateur reçoit 1 000 coins à l'issue de l'onboarding, prêts à être misés. En dehors du mode “online”, un mode entraînement gratuit permet de jouer contre une IA (trois niveaux de difficulté) ou en local à deux sur le même appareil.

Résumé rapide de ce que contient l’application :

- Onboarding explicatif, contenant simulation de partie guidée et récompense de 1000 coins à l'issue du tutoriel
- Mode entraînement : parties contre une IA (Minimax, 3 niveaux de difficulté) ou en local à deux sur le même appareil
- Salon pour participer ou créer un match avec liaison à une mise (en mode mocké)
- Historique des parties et statistiques
- Page de paramètres pour gestion du thème dark/light, niveau de difficulté, choix de la langue
- Design inspiré de l'écosystème betclic pour essayer de mettre l’utilisateur dans une application “entertainment”
- Test unitaires

2. Choix technique

- ***Clean Architecture feature-first*** : chaque feature est autonome avec ses couches domain / data / application / présentation.Le domaine ne contient aucun import Flutter
- ***Riverpod*** (via *riverpod_generator*) : pour le state management. Pouvoir être indépendant du contexte pour gérer les providers, mise à jour de l’état automatique, les providers sont bien isolés pour être testable etc..
- ***Freezed*** : Permet d’avoir des objets immutables, l’accès au copyWith, sealed class
- ***Hive*** : Permet d’avoir une persistance des données locales légères
- ***mocktail*** : pour les tests unitaires, simple et null-safe sans codegen

3. Ce que j’aurais aimé faire en plus

- Lier le mode “Compétition” à un serveur type supabase ou autre
- Authentification de l’utilisateur
- Mise en place de streak de victoire pour ajouter un multiplicateur de gains de coins
- Mettre en place des animations plus abouties lorsque l’utilisateur gagne ou perd une partie
- Mieux travailler le rendu *Theme Light* de l’application