🤖 CHECKLIST – ÉQUIPE LLM (Client + Serveur)

👨‍💼 **Team Leader : Ayman Zouini**

🎯 Objectif général

Intégrer une IA (LLM) dans l'application mobile existante pour créer un chatbot culinaire capable de :

- Comprendre les ingrédients ou le nom d'une recette
- Proposer des recettes correspondantes
- Donner les instructions et ingrédients à partir d'une base JSON

👩‍💻 ÉQUIPE LLM – CÔTÉ CLIENT (3 personnes)

Objectif : *faire fonctionner un petit modèle IA en local (mobile / offline)*

🔷 Client LLM 1 – Modèle IA local

- [ ] Choisir et installer le modèle léger (ex. Mistral 7B quantized ou TensorFlow Lite)
- [ ] Tester la génération locale (proposer des recettes à partir d'ingrédients)
- [ ] Optimiser vitesse et mémoire (≤ 300 Mo, inférence <100 ms)
- [ ] Vérifier le fonctionnement hors ligne

🔷 Client LLM 2 – Base locale et intégration JSON

- [ ] Connecter le modèle IA à la base JSON (recettes, ingrédients, instructions)
- [ ] Créer un petit moteur de recherche local pour filtrer les recettes
- [ ] Stocker les données dans SQLite
- [ ] Synchroniser les données locales avec le backend si connecté

🔷 Client LLM 3 – Communication & cohérence

- [ ] Gérer la communication entre le LLM local et le serveur IA
- [ ] Définir quand utiliser le modèle local ou distant
- [ ] Vérifier la cohérence des réponses (même format, même style)
- [ ] Tester le flux complet : saisie → suggestion → recette complète

💻 ÉQUIPE LLM – CÔTÉ SERVEUR (3 personnes)

Objectif : *gérer les tâches complexes et les réponses détaillées*

🔷 Serveur LLM 1 – Déploiement du modèle

- [ ] Installer le modèle LLaMA 3 8B ou OrcaMini-3B sur le serveur
- [ ] Configurer l'environnement Python (`transformers` , `torch` , `accelerate`)
- [ ] Vérifier les performances sur GPU / CPU

- [ ] Mettre en place une API simple pour interagir avec le modèle

🔷 Serveur LLM 2 – Connexion à la base JSON

- [ ] Intégrer la base JSON côté serveur (PostgreSQL ou fichier)
- [ ] Permettre au LLM d'utiliser les données de recettes
- [ ] Ajouter un retrieval simple (embeddings / recherche de similarité)
- [ ] Tester les réponses du modèle avec les données réelles

🔷 Serveur LLM 3 – Sécurité & Monitoring

- [ ] Protéger la communication (Tor ou HTTPS)
- [ ] Ajouter un cache (Redis ou dictionnaire Python) pour les réponses fréquentes
- [ ] Suivre les performances (temps de réponse, erreurs)
- [ ] Aider à la synchronisation client ↔ serveur