### Next Door Buddy - Application d'entraide de quartier

#### Introduction (Objectif du projet):

- Next Door Buddy est une application communautaire qui permet aux habitants d'un quartier de s'entraider, échanger des services, et discuter via une messagerie sécurisée.
- L'application facilite :
  - Les échanges d'objets et de services entre voisins.
  - Le volontariat (par exemple : aider une personne âgée à faire ses courses).
  - La communication locale avec une messagerie et un forum de discussion.
  - Les suggestions intelligentes pour connecter les utilisateurs ayant des intérêts similaires.

#### 📌 Fonctionnalités principales :

Catégorie	Fonctionnalité
Gestion des utilisateurs	Inscription, connexion, modification du profil, authentification sécurisée.
Système d'annonces	Création, modification, suppression d'annonces pour échanger des objets ou services.
Volontariat	Un utilisateur peut s'inscrire en tant que bénévole et aider les autres.
Messagerie	Chat privé entre utilisateurs + discussion de groupe via des canaux.
Notifications	Alertes pour nouveaux messages, nouvelles annonces, suggestions personnalisées.
Système de recommandations	Algorithme basé sur les interactions pour suggérer des annonces ou bénévoles pertinents.
Web scraping (événements locaux)	récupérer automatiquement des événements locaux

### rien vers le github et le trello :

https://github.com/Aurore-MB/Next Door Buddy

https://trello.com/invite/b/67d6a582cbcf5ff00dfc5a8b/ATTle6b2fe2628becd6b98562655 3ccda99970396052/next-door-buddy

# 📌 Technologies utilisées :

Composant	Technologie
Frontend	React.js
Backend (API Microservices)	FastAPI (Python) + Node.js
Base de données	Neo4J (recommandations), MongoDB (utilisateurs, annonces, messages)
Messagerie en temps réel	WebSockets + Redis
Stockage de fichiers	MinIO (S3-like)
Tâches en arrière-plan	Celery + Redis
Conteneurisation & Déploiement	Docker
Scraping	JavaFX

## 📌 Idées de traitement pour chaque fonctionnalités :

Catégorie	Idée de traitement (Approche technique)
Gestion des utilisateurs	Authentification JWT + stockage MongoDB. Chiffrement des mots de passe avec bcrypt. API REST pour inscription, connexion, profil.
Système d'annonces	API REST en FastAPI pour gérer les annonces. Stockage en MongoDB. Recherche optimisée avec filtres avancés.
Volontariat	API REST pour inscription des bénévoles.
Messagerie	WebSockets pour chat en temps réel Stockage des messages en MongoDB Chiffrement des conversations.
Notifications	Envoi en temps réel via Redis + Celery. Notifications push avec Firebase (FCM).
Système de recommandations	Algorithme de Neo4j basé sur les interactions. Mise à jour des recommandations avec Celery + Redis.
Web scraping (événements locaux)	Extraction de données avec JavaFX et stockage dans la base de données