🎯 Projet Data Engineer - Détection de Fraude

Phase 1 : Compréhension du problème et cadrage

- Définition des règles simples de détection :
    - même carte utilisée sur plusieurs IP très différentes ;
    - email jetable + carte prépayée + device inconnu.

Phase 2 : Ingestion & stockage des données

- Simuler ou utiliser un fichier de transactions (`transactions.csv`).
- Écrire un script d'ingestion Python (pandas) pour charger les données.
- Stocker dans SQLite, PostgreSQL, ou en fichiers plats (CSV, Parquet).
- Automatiser l'ingestion (via un script CLI, ou Airflow simple si tu es à l'aise).
- **Résultat attendu** : script d'ingestion + base de données ou fichier prêt pour traitement.

Phase 3 : Transformation & détection de fraude

- Nettoyage des données : formats de date, suppression de doublons, traitement des emails.
- Implémenter les règles définies dans la phase 1 sous forme de fonctions Python.
- Ajouter une colonne `fraud_score` ou `is_suspect` pour marquer les cas suspects.
- Possibilité d'ajouter des features supplémentaires (nb d'utilisations d'une IP, ratio email jetable, etc.)
- **Résultat attendu** : table/fichier enrichi avec détection appliquée.

Phase 4 : Visualisation et alerting

- Créer un notebook Jupyter avec des visualisations : histogrammes, heatmaps, top 10 des cartes/IP/email suspects.
- (Optionnel) Créer un mini dashboard avec Streamlit ou Dash.
- **Résultat attendu** : outil simple pour visualiser rapidement les cas de fraude potentielle.

Phase 5 : Packaging, déploiement local et documentation

- Organiser le projet proprement :
    - `data/`, `notebooks/`, `scripts/`, `output/`, `README.md`, `requirements.txt`
- Ajouter des explications claires dans le README :
    - comment exécuter chaque script
    - à quoi sert chaque étape
- (Optionnel) Dockerisation pour exécution locale simple.

  Résultat attendu : projet clean, versionné sur GitHub, prêt à être montré en entretien.
