**Projet PiggyTrack :**

**Objectif :** Une application pour suivre ses dépenses, revenus et budgets mensuels.

**Fonctionnalités clés :**

* Ajout de transactions (dépenses/revenus) avec catégories (nourriture, loisirs, etc.).
* Génération de rapports (graphiques avec Matplotlib ou Seaborn en Python).
* Alertes pour dépassement de budget.
* Export des données en CSV/Excel. **Pourquoi c'est intéressant** :
* Utilisation de **structures de données**, **manipulation de fichiers** et **visualisation de données**.
* Projet utile et facile à présenter avec des résultats visuels.

**Répartition des Tâches pour 2 Personnes**

Voici une répartition équilibrée et logique pour travailler en parallèle :

**Personne 1 : Backend et Logique Métier**

* **Gestion des transactions** :
  + Créer les classes Transaction, Budget, et Utilisateur.
  + Implémenter les fonctions pour ajouter, supprimer et modifier des transactions.
* **Stockage des données** :
  + Sauvegarder les transactions dans un fichier JSON ou une base de données SQLite.
  + Charger les données au démarrage de l’application.
* **Logique des budgets** :
  + Calculer les dépenses par catégorie.
  + Vérifier les dépassements de budget et générer des alertes.

**Personne 2 : Frontend et Visualisation**

* **Interface utilisateur** :
  + Créer un menu en ligne de commande (avec cmd ou argparse) ou une interface graphique (Tkinter/PyQt).
  + Permettre à l’utilisateur d’interagir avec l’application (ajout, suppression, visualisation).
* **Visualisation des données** :
  + Utiliser matplotlib ou seaborn pour générer des graphiques (dépenses par catégorie, évolution mensuelle).
  + Afficher les alertes de dépassement de budget.
* **Export des données** :
  + Implémenter une fonction pour exporter les transactions en CSV/Excel.

**Tâches communes** :

* Définir ensemble la structure des classes et des fichiers.
* Tester et valider les fonctionnalités ensemble.
* Documenter le code (commentaires, README sur GitHub).

**Bibliothèques de Base :** pandas, matplotlib, seaborn, tkinter ou PyQt5, sqlite3 ou sqlalchemy, openpyxl, psycopg2.

**Base de données :** PostgreSQL