Personal Finance Tracker

Scenario: Develop an application for users to track their income and expenses, categorize transactions, and view summaries.

Evaluation Focus: OOP, DSA, Data Aggregation, Reporting, Data Validation.

Core Requirements:

1. User Registration & Authentication:
    a. Users can sign up, log in, and manage their profiles.
    b. Admins can manage access permissions.
2. Account Management:
    a. Users can create and manage multiple financial accounts (e.g., Savings, Credit, Wallet) with an initial balance. Each account maintains a dynamic balance updated in real-time.
3. Transaction Management:
    a. Users can add income and expense transactions with meta data (Amount, Date, Description, Category e.g., Salary, Gift, Food, Transport, Rent).
    b. Transactions are automatically reflected in the associated account.
4. Categorization:
    a. Users can define and nest custom categories for better organization.
    b. Categories can be reused across accounts and transactions.
5. Reports:
    a. Users can view all transactions or apply filters by account, category, or date range.
    b. Summaries and trends are generated based on filtered data.

Data Structures & Algorithms:

- Fast lookup of accounts or categories by name or ID
- Efficient insertion/removal of transactions in chronological order
- Dynamic list of nested unique categories
- Algorithms for calculating sums, averages, and generating reports.

Advanced Features (Choose 2+):

- Balance Summary: Display current balance for all accounts and a total net balance.
- Monthly/Annual Reports: Generate summaries of income vs. expenses for specific periods.
- Budgeting: Users can set monthly budgets for specific expense categories. Track actual spending against budget.

- Recurring Transactions: Mark transactions as recurring (e.g., monthly rent) and automatically generate them.
- Persistent Storage: Save all user data and transactions to a file (CSV/JSON).

SQL Problem:

You are given a simplified schema for a personal finance tracking system. The system stores transactions and their associated user details. Write a query to calculate the total amount spent by each user in each category, excluding income transactions.

Users

- user_id (INT, Primary Key)
- username (VARCHAR)
- email (VARCHAR)

Transactions

- transaction_id (INT, Primary Key)
- user_id (INT)
- amount (DECIMAL)
- category (VARCHAR)
- transaction_date (DATE)
- is_income (BOOLEAN)