Expense Reporting System – Project Overview

1. Project Objective

Design and implement an Expense Reporting System using SQLite and Python with a Command-Line Interface (CLI). The system should allow users to log, manage, and analyze expenses while enforcing relational database principles and SQL queries.

You will create a Python program that runs and prompts the user for input on the command line (Console window / Terminal). All the commands should run in your program directly. Meaning all commands to be implemented in a single program.

2. Core Features

a. User Management & Role-Based Access Control (RBAC)

- Multi-user authentication system.

- Users are assigned roles with different permissions:

  - **Admin**: Can manage users, categories, and view all expenses.

  - **Regular User**: Can only manage their own expenses.

- Users can manage their own expenses.

Only login with Admin role is allowed to create users and categories. You can have one 'admin' user account created in the DB for admin use. You can then add more Admin roles.

b. Expense Management

- Users can add, update, delete, and view expenses for themselves.
- Each expense should have the following fields:
  - Expense ID (Primary Key)
  - User ID (Foreign Key)
  - Category (e.g., Food, Travel, Utilities, etc.)
  - Amount
  - Date
  - Description (Optional)
  - Tag
  - Payment Method (e.g., Cash, Credit Card, Bank Transfer)

3. Advanced Features

3.1 CSV Import Functionality

Users should be able to bulk import expenses from a CSV file.

- The format of the CSV file is same as adding single expense using a command
- The system should validate data before inserting it into the database.

- Handle duplicate entries and incorrect formats.

3.2 CSV Export Functionality

- Users should be able to export expenses to a CSV file.

5. Implementation

a. CLI Commands

Help and Information:

help - should print the list of commands available

User Authentication & Role Management:

- login <username> <password> – Authenticate user.
- logout – End user session.
- list_users – Show all users and their roles (Admin only).

User, Category & Payment Method Management:

- add_user <username> <password> <role> – Register a new user with a role. (role = Admin or User)
- add_category <category_name> – Add a new expense category. (Admin only)
- list_categories – View all available categories.
- add_payment_method <method_name> – Add a new payment method.
- list_payment_methods – View available payment methods.

Expense Management:

- add_expense <amount> <category> <payment_method> <date> <description> <tag> – Add a new expense.
- update_expense <expense_id> <field> <new_value> – Update an existing expense.
- delete_expense <expense_id> – Delete an expense.
- list_expenses [filters] – View expenses with optional filters (category, date, amount range, payment method, etc.). – Implementation of filter is up to you.

Import & Export:

- import_expenses <file_path> – Import expenses from a CSV file.
- export_csv <file_path>, sort-on <field_name> – Export expenses to a CSV file, sorted on given field_name

Error handling:

Implement robust error handling so that your program should not crash on bad input

b. Reports

- report top_expenses <N> <date range> – Show top N highest expenses for a given date range
- report category_spending <category> – Show total spending for a specific category.
- report above_average_expenses – Show expenses greater than the category average.
- report monthly_category_spending – Show total spending per category for each month.
- report highest_spender_per_month – Show the user with the highest spending for each month.
- report frequent_category – Show the most frequently used expense category.
- report payment_method_usage – Show spending breakdown by payment method.
- report tag_expenses – Show number of expenses for each tag

6. Submission, Evaluation and grading, presentations and Further Notes:

1. Only one student from each team will submit (upload to moodle) **a single ZIP file** that includes all your deliverables. Others submit a text file with name of student who submitted the assignment.
    a. E-R Diagram, Relational Schema design (detailed)
    b. sqllite Db file for the schema
    c. Fully working python program (any number of .py files. Ideally in a separate folder) Ensure that ZIP is extractable and runs on Windows. I have had trouble with submissions from *mac*. Please confirm and test working on Windows.
    d. Readme / Help file with all the command implemented
2. You will submit your assignment ***before* 12pm on April 4th**. I will not consider late submissions.
3. If anything is not clear in the project, you are expected to get your doubts cleared *before* our next class. Validate your understanding with me
4. You are free to add more commands or more reports as you see fit.
5. **Evaluation and Grading:**
    a. The project is for 20 marks.
    b. I will roughly divide the grading as follows: E-R and Database schema: 5 marks, All other deliverables including a fully working program – 10 marks. 5 marks for presentation, group-work, overall understanding, completeness of design, good schema implementation etc.
6. **Group Presentations:**
    Group presentations are scheduled on **Friday, April 4th and Saturday April 5th** .
    You will present only what you have submitted on moodle. You will have to download your submission at the time of presentation and then demonstrate.
    Local copy on any student laptop is not allowed!

All the best!