**DOCUMENTATION**

**Personal Finance Management Application**

This project is a simple command line application that helps users to manage their personal finances by tracking income, expenses, and generating financial reports.

**1.Prerequisites**

1a. Install Python: Ensure Python (version 3.7 or higher) is installed. You can download it from https://www.python.org/

1b. Check SQLite3: SQLite3 is included in Python by default, so no additional installation is needed for database functionality.

**2.Running the Code**

2a. Save the Script: Copy the code into a file named, for example, financemanager.py

2b. Run the Script: Open a terminal or command prompt then navigate to the directory containing the script now run the script using the command : **python financemanager.py**

2c. Initialize the Database: The init\_db() function is automatically called when the program starts. It creates the necessary database (finance\_manager.db) and tables if they don't already exist.

**3.Using the Application**

The application provides a menu-driven interface.

**MAIN MENU:**

1. Register

2. Login

3. Exit



**Option 1: Register**:

* Enter a unique username and password to create an account.
* If the username already exists, you'll be prompted to try a different one.

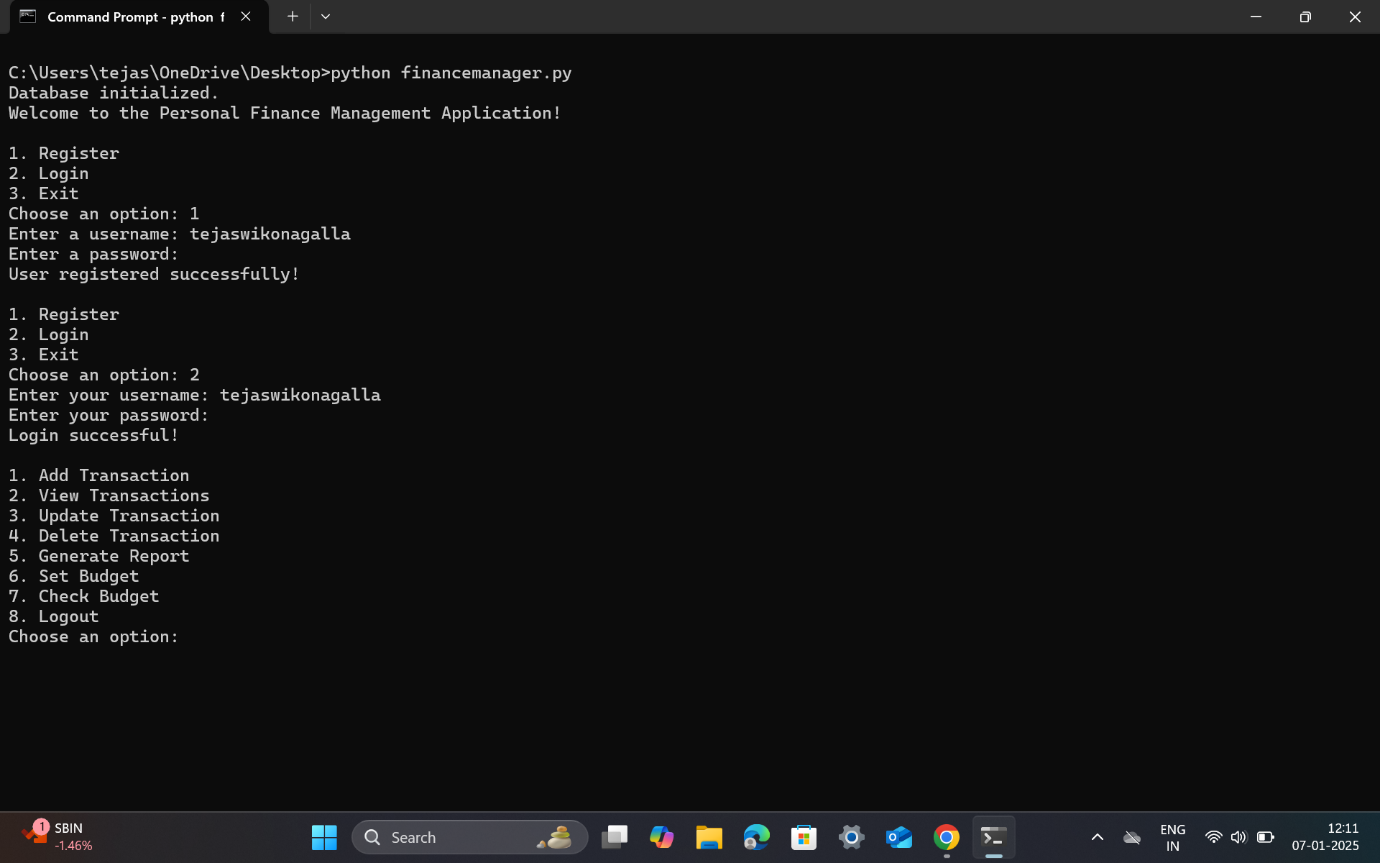
**Option 2: Login**:

* Log in using your registered username and password.
* Upon successful login, you'll access the user's menu.

**Option 3: Exit**:

* Exits the application.

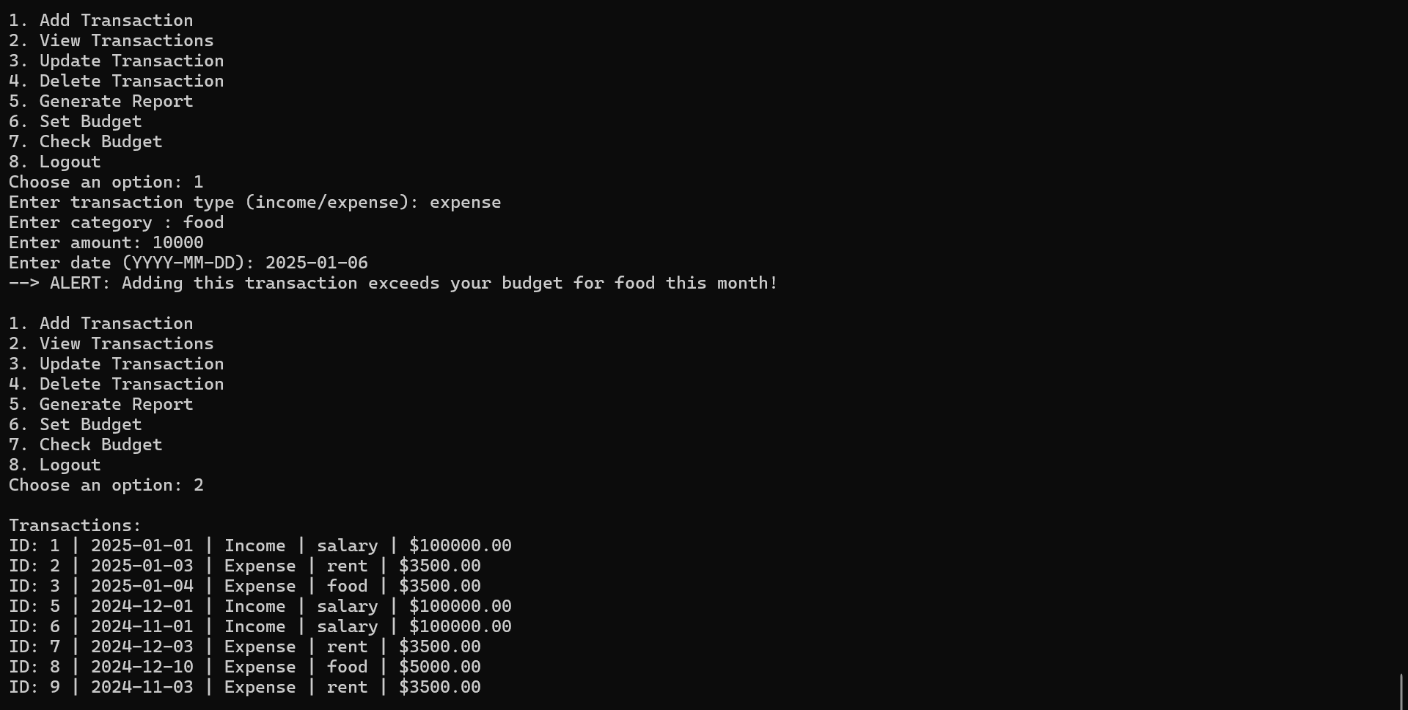
**User Menu**: Once logged in, users can perform the following tasks:



1. **Add Transaction**:

* Specify transaction type (income or expense), category (e.g., Food, Rent, Salary), amount, and date (leave blank to use today's date).
* If the transaction type is income, then transaction will be automatically saved into the database.
* If the transaction type is expense, then at first checking will be done to see if there is any budget that is set already set for that particular category in the month specified under date. If the expense exceeds the budget for that month under the category specified, then an alert will be generated and the transaction will not be added otherwise the transaction will be added to the database.





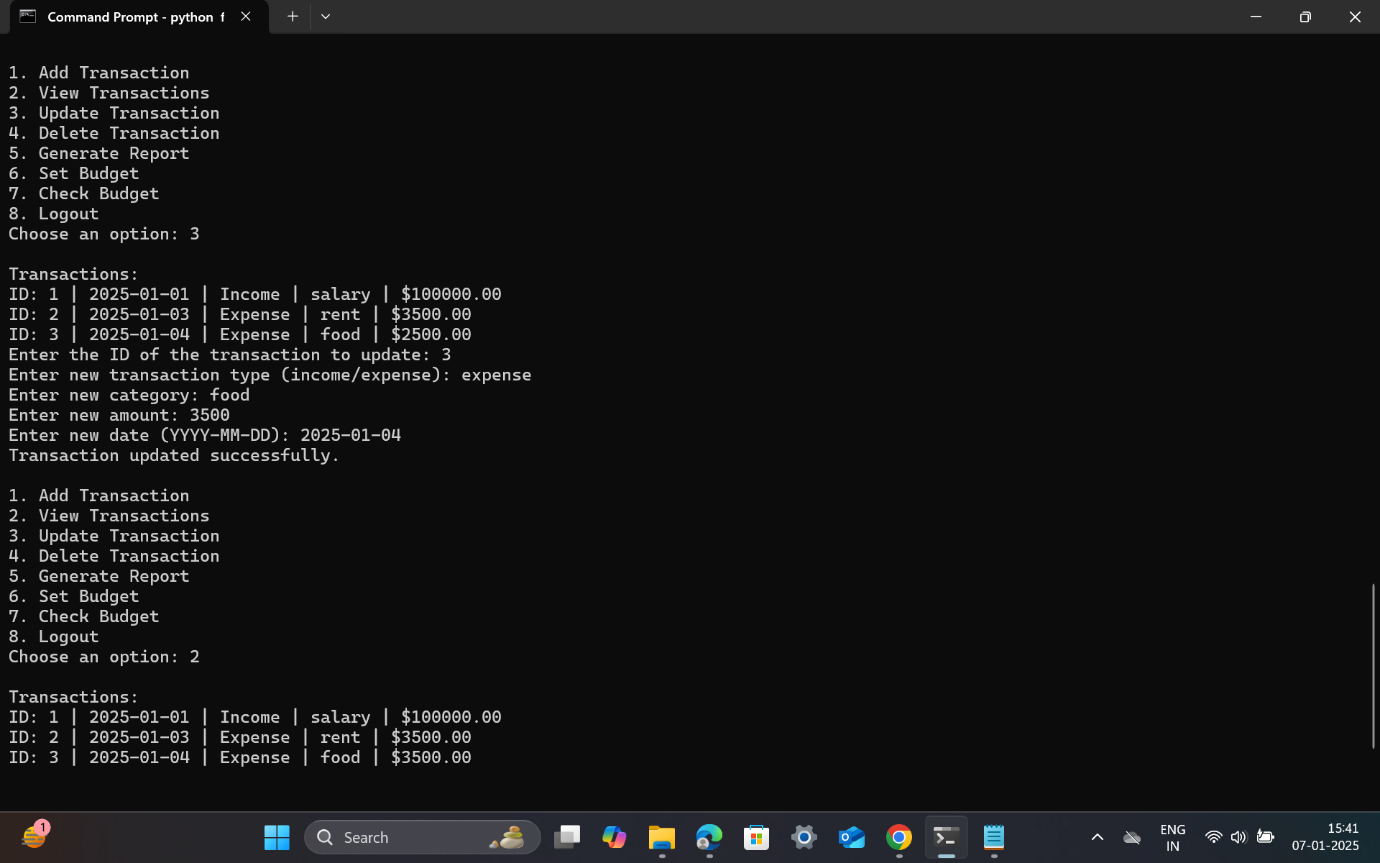
1. **View Transactions**:

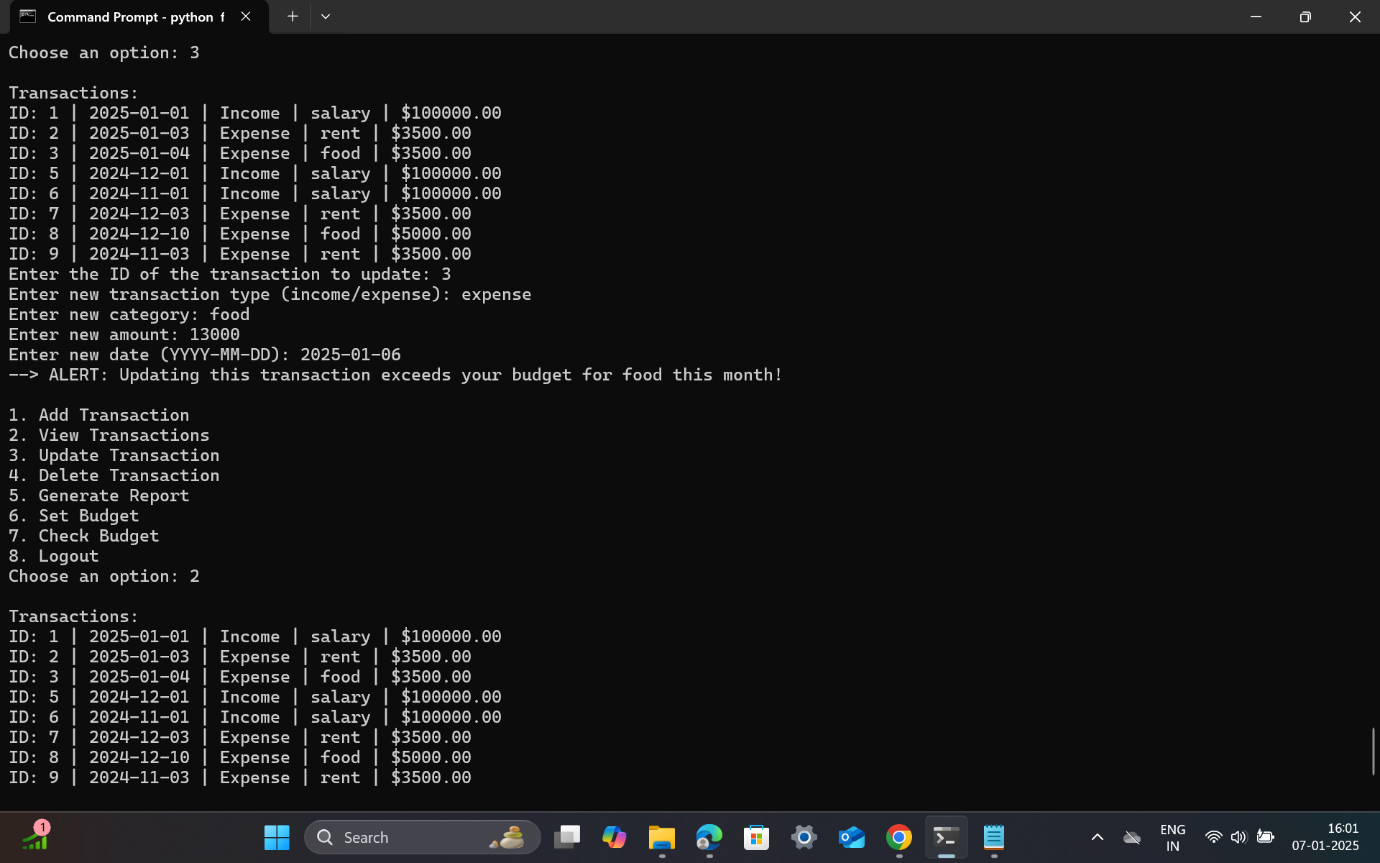
* Displays all transactions for the logged-in user, categorized by type and date.



1. **Update Transaction**:

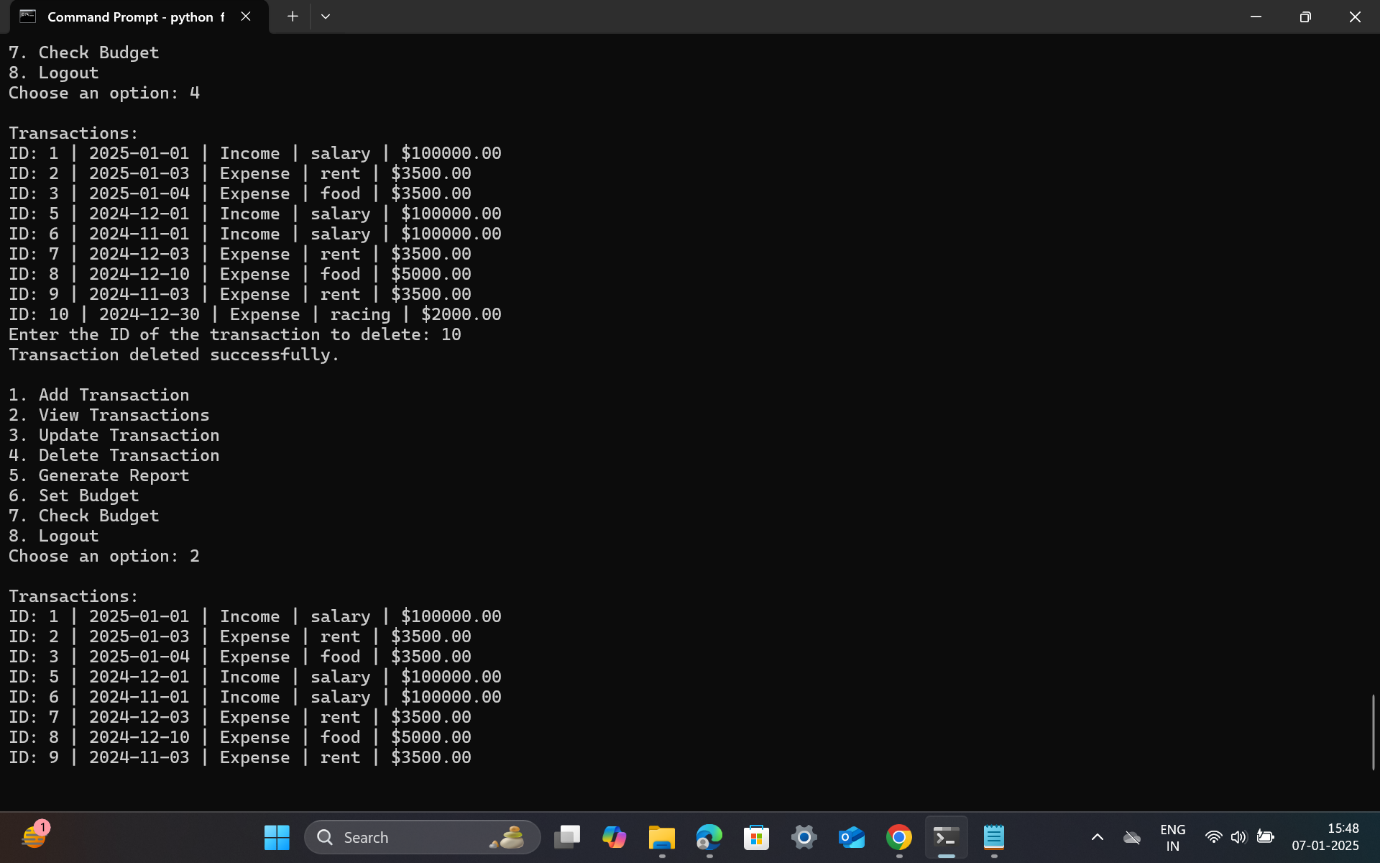
* Unless there is a restriction of budget under expense category, any transaction can be updated by specifying it's ID, new transaction type/new category/new amount/new date.

****

****

1. **Delete Transaction:**

* Any transaction can be deleted by specifying the ID.



1. **Generate Report**:

* It summarizes total income, total expenses, and calculates savings (income - expenses).
* Report can be generated for month / year upon requirement.





1. **Set Budget**:

* It helps to set a monthly budget for specific categories so that users can be notified when their expenses exceed the budget. For example, a $500 budget for Food.



1. **Check Budget**:

* The budget amount, category, month and year for which budget has been set can be checked by choosing this feature.



1. **Logout**:

* Returns you to the main menu.



Testing with invalid credentials:

