FlexiSaf Internship Project

Budget Tracker Application

Overview

This project is a simple, user-friendly **Budget Tracker** application that allows users to manage their finances. The application is built in Java and provides functionalities to record, edit, and analyze income and expenses, helping users maintain a clear financial summary.

Features

1. **Add Transactions**
    - Users can record transactions by providing details like date, amount, description, category, type (Income or Expense), and whether the transaction is recurring.
    - Validation ensures that no invalid data (e.g., negative amounts or empty descriptions) is entered.
2. **Edit Transactions**
    - Transactions can be modified by updating any of the fields.
    - The program ensures accurate updates to income, expenses, and recurring costs.
3. **View Financial Summary**
    - Displays total income, total expenses, recurring expenses, and overall savings.
    - Alerts users if they are in a negative budget.
4. **List All Transactions**
    - Transactions are displayed in a tabular format with details such as date, amount, description, category, transaction type, and recurrence status.
5. **Persistence**
    - While this prototype does not implement file/database storage, the architecture is designed to extend for saving data.

6. **Error Handling**

   - Ensures robustness with proper handling of invalid inputs and exceptions like incorrect date formats.

Tools and Technologies

1. **Programming Language**:

   - **Java**: Used for its object-oriented features, robustness, and simplicity.
2. **Libraries/Packages**:

   - `java.util`: For collections like `ArrayList` to store transactions.
   - `java.time`: For handling dates and times.
   - `java.util.Scanner`: For user input.
3. **Design Principles**:

   - **Encapsulation**: Used to restrict direct access to fields.
   - **Separation of Concerns**: Organized into `Transaction` and `Tracker` classes for modularity.

Technical Details

Class: Transaction

- Represents a financial record with attributes such as date, amount, description, category, type, and recurrence status.
- Includes:
  - Getters and setters for encapsulation.
  - A `toString()` method for easy formatting of transaction details.

Class: Tracker

- Manages all transactions and provides operations to add, edit, display summaries, and list transactions.
- Key attributes:
  - `ArrayList<Transaction>`: Stores all transactions.
  - **Income and Expenses**: Tracks total amounts.
  - **Recurring Cost**: Summarizes recurring expenses.

Sample Outputs

1. **Add Transaction**

```
Enter date (yyyy-MM-dd): 2024-12-01
Enter amount: 500
Enter description: Grocery shopping
Enter transaction type (Income/Expenses): Expenses
Enter category: Grocery
Is this a recurring transaction? (true/false): true
Transaction added successfully!
```

2. **View Summary**:

```
Your total income: 1000.0
Your total expenses: 700.0
Your recurring expenses: 500.0
Total savings: 300.0
```

3. **List Transactions**:

```
S/N  Date          Amount    Description       Category  Type       Recurring
1    2024-12-01    500.00    Grocery shopping Grocery   Expenses   Yes
2    2024-12-02    1000.00   Salary            Salary    Income     No
```

How to Run

1. Install a Java Development Kit (JDK).
2. Save the code as `BudgetTracker.java`.
3. Compile and run the program using: java BudgetTracker.java

---