21/Nov/2024

line of credit calculator

by- Vandana salian padmanabha

1. **Overview:**

The **Credit Calculator Application** is a desktop software program built using Python, designed to manage and calculate credit-related details for various accounts. This application provides functionality to store, update, view, and graphically display data related to credit usage and limits for individual accounts. The system interacts with an SQLite database and CSV files for persistent storage and data export. The tool is useful for users who need to track and analyse credit usage, available credit, and balances, and calculate interest based on used and unused credits.

1. **Key Features:**
2. **Account Management**: Users can input and update credit account details, including the account number, name, credit limit, and the amount used. This data is stored in a SQLite database and is also recorded in a CSV file for easy access.
3. **Interest Calculation**: The application calculates daily interest on both the used and unused credit amounts, based on predefined annual interest rates. These rates can be modified to reflect different policies.
4. **Graphical Representation**: The system generates pie charts to visually represent the distribution of used versus unused credit for a specific account.
5. **Data Persistence**: The application saves account details in both a SQLite database and a CSV file, ensuring that data is not lost between sessions.
6. **Account View**: Users can retrieve and view detailed information about a specific account, including calculated daily interest on both used and unused credit.
7. **CSV Export**: The data for all accounts can be exported into a CSV file, which users can view in a tabular format.
8. **Technologies Used:**

* **Python**: The application is built using Python, leveraging its powerful libraries for database management, GUI development, and data visualization.
* **SQLite**: SQLite is used for the database to store account information persistently.
* **Tkinter**: Tkinter is employed to create the graphical user interface (GUI) that allows users to interact with the system.
* **Matplotlib**: The Matplotlib library is used to generate pie charts for credit usage visualization.
* **CSV**: CSV files are used for exporting and viewing account details in a tabular format.

1. **Detailed Description:**
2. **1. Database Management (DatabaseManager class):**

The DatabaseManager class is responsible for managing the SQLite database that stores the credit account data. It supports the following functionalities:

* **Table Creation**: It creates a table credit\_accounts to store account details, including account\_number, name, credit\_limit, amount\_used, unused\_credit, balance, and last\_updated.
* **Insert/Update Data**: The insert\_or\_update method inserts new accounts or updates existing ones with the latest data.
* **Fetching Account Data**: The fetch\_account method retrieves account information from the database based on the account number.
* **CSV Integration**: It writes updated account data into a CSV file, and it can read all existing account details from the CSV for review.

1. **2. Interest Calculation (InterestCalculator class):**

The InterestCalculator class handles the calculation of daily interest for credit accounts:

* **Daily Interest Calculation**: It calculates daily interest for used and unused credit. The annual rates for used and unused credit are provided as inputs, and the class calculates the daily rate by dividing the annual rate by 365.

1. **3. Graph Plotting (GraphPlotter class):**

The GraphPlotter class generates a pie chart to visualize the distribution of used and unused credit:

* **Pie Chart**: It uses the matplotlib library to plot a pie chart showing the proportion of used versus unused credit for an account.

1. **4. Credit Account Representation (CreditAccount class):**

The CreditAccount class represents a credit account with attributes such as account\_number, name, credit\_limit, amount\_used, unused\_credit, balance, and last\_updated. It also includes a method for updating the account balance when the amount used changes.

1. **5. Graphical User Interface (CreditCalculatorApp class):**

The CreditCalculatorApp class implements the GUI using Tkinter. Key features include:

* **Input Fields**: The user can input the account number, name, credit limit, and amount used into the provided fields.
* **Buttons**:
  + Calculate: This button triggers the calculation of interest and updates the account information in the database.
  + Show Graph: This button generates a pie chart showing the distribution of used and unused credit for the selected account.
  + View Account: This button displays detailed account information, including interest calculations for used and unused credit.
  + View All Accounts: This button allows users to view all accounts stored in the CSV file.
* **Display**: Account details, including daily interest calculations and overall balance, are displayed in a label within the GUI.

1. **Workflow:**
2. **User Interaction**:
   * The user inputs the account number, name, credit limit, and amount used.
   * Upon clicking "Calculate", the application updates or inserts the data into the SQLite database and stores the updated account information in the CSV file.
   * The user can view the account details, calculate daily interest, or view a graph of credit usage.
3. **Database & CSV Integration**:
   * The data is stored and retrieved from the SQLite database for efficiency and reliability.
   * Simultaneously, the account details are saved in a CSV file, allowing the user to access the data in a more readable tabular form.
4. **Interest Calculation**:
   * The system computes daily interest for both used and unused credit amounts based on annual rates. The interest calculations are then displayed in the user interface.
5. **Graph Visualization**:
   * A pie chart visualizes the breakdown between used and unused credit, making it easier for users to understand their credit distribution.
6. **Possible Extensions:**
7. **Interest Rate Customization**: Allow users to modify the annual interest rates for used and unused credit.
8. **Report Generation**: Add functionality to generate comprehensive reports of all accounts over a certain period.
9. **Account Deletion**: Provide an option to delete accounts from the database and CSV file.
10. **Security**: Implement password protection or encryption for account data.
11. **Additional Charts**: Add more complex graphical representations (e.g., bar charts showing monthly credit usage).
12. **Conclusion:**

The **Credit Calculator Application** is a robust tool for managing and tracking credit account data. It allows users to input, update, and analyse their credit usage, interest, and balances while providing a user-friendly interface. This tool can be used by individuals or businesses to efficiently manage multiple credit accounts and make informed decisions about credit usage.