EXPENSE TRACKER

Overview

The Expense Tracker is a comprehensive application designed to help users manage their personal finances. Built using Python and SQLite for data storage, the application allows users to categorize their expenses, set monthly budgets, track daily spending goals, and visualize their financial data. This project demonstrates my skills in database management, data analysis, and Python programming.

Project Structure

The project is organized into several functions, each handling specific tasks related to expense management. The key components of the project include:

1. **Database Initialization**:
   - `init_db()`: Creates the necessary tables for categories, expenses, and budgets if they do not exist.
2. **Category Management**:
   - `add_category(name)`: Adds a new expense category to the database.
3. **Expense Management**:
   - `add_expense(amount, category_name, date, description)`: Adds a new expense entry, linking it to the appropriate category.
4. **Budget Management**:
   - `set_budget(month, amount)`: Sets or updates the budget for a specific month.
   - `get_budget_status(month)`: Retrieves the budget status, including total expenses, budget amount, and remaining balance for the month.
5. **Expense Tracking**:
   - `get_monthly_expense(month)`: Calculates the total expenses for a given month.
   - `get_daily_spending_goal(month)`: Computes the daily spending goal based on the remaining budget and days left in the month.
6. **Visualization**:
   - `visualize_expenses()`: Generates visual representations of expenses by category and monthly expenses using pandas and matplotlib.

Key Features

- **User-Friendly Interface**: The application provides a text-based menu for easy navigation and input of expense data.
- **Database Integration**: Utilizes SQLite for persistent storage of expense data, ensuring data is saved between sessions.
- **Categorization**: Allows users to categorize their expenses for better organization and analysis.

- **Budget Tracking**: Enables users to set monthly budgets and track their spending against these budgets.
- **Daily Spending Goals**: Provides daily spending goals to help users manage their finances more effectively.
- **Data Visualization**: Includes pie charts and bar graphs to visualize spending patterns, helping users to understand their financial habits better.

Code Highlights

- **Database Operations**: Efficiently manages database connections and operations to ensure data integrity and performance.
- **Data Analysis**: Utilizes pandas for data manipulation and matplotlib for creating insightful visualizations.
- **User Input Handling**: Ensures robust handling of user inputs, including validation and error handling to provide a seamless user experience.

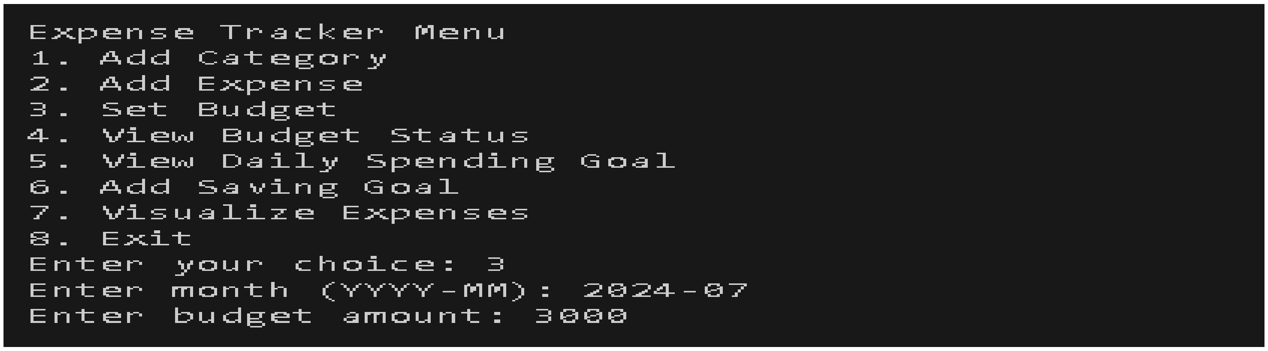
Learning Outcomes

- **Database Management**: Gained experience in designing and managing relational databases using SQLite.
- **Data Analysis and Visualization**: Improved my skills in data analysis using pandas and creating visualizations with matplotlib.
- **Python Programming**: Enhanced my understanding of Python programming, including working with functions, loops, conditionals, and user input handling.
- **Project Organization**: Learned to structure a project in a modular way, promoting code readability and maintainability.

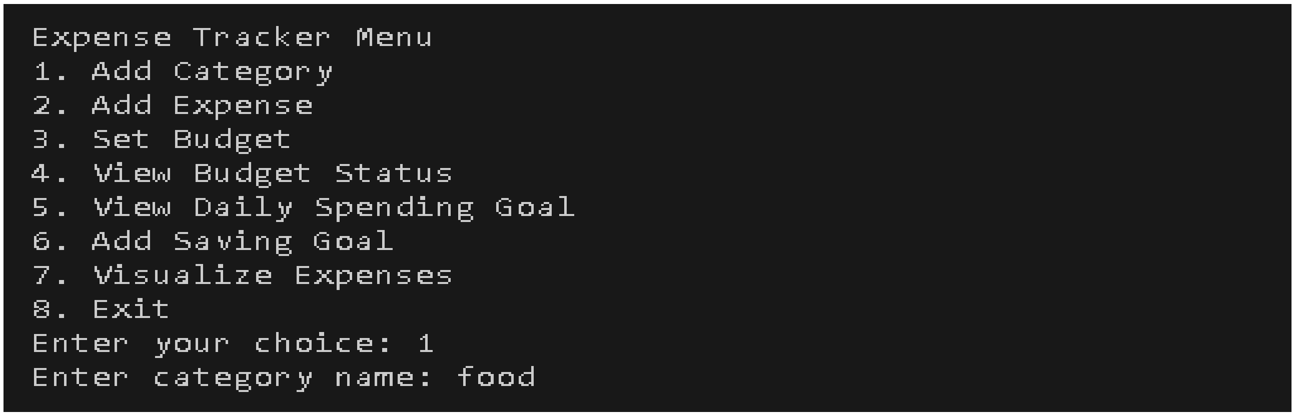
Conclusion

The Expense Tracker project showcases my ability to develop practical and user-friendly applications using Python and SQLite. By integrating data management, budget tracking, and data visualization, I have created a tool that can help users gain better control over their finances. This project highlights my technical skills and my commitment to creating useful and efficient software solutions.

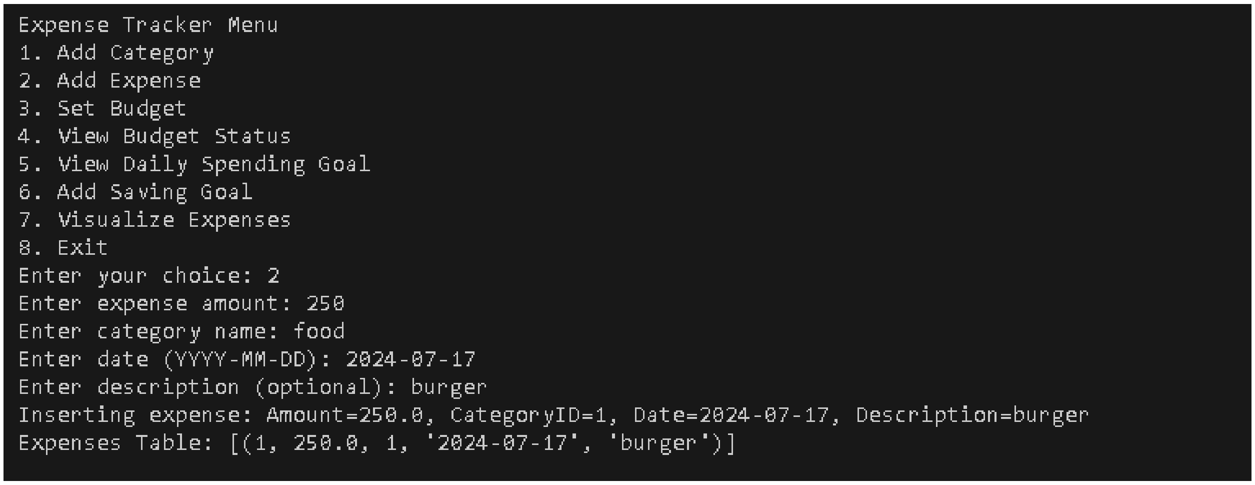
Implementation



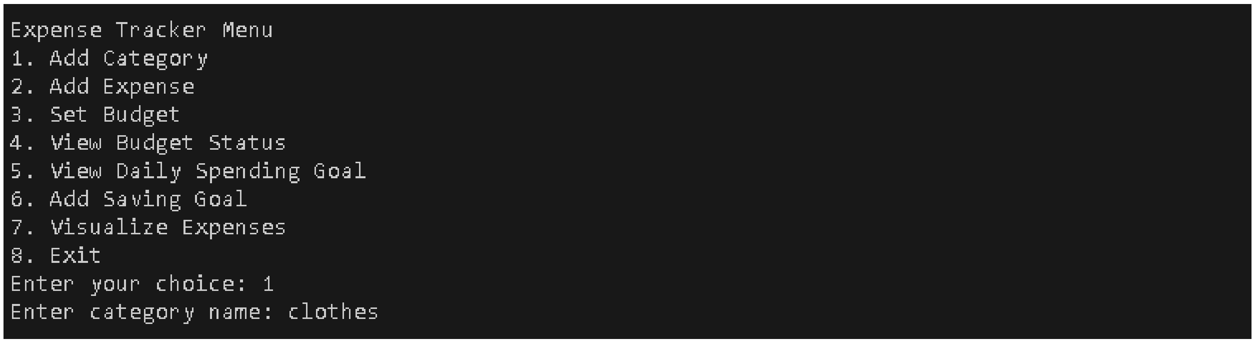
```
Expense Tracker Menu
1. Add Category
2. Add Expense
3. Set Budget
4. View Budget Status
5. View Daily Spending Goal
6. Add Saving Goal
7. Visualize Expenses
8. Exit
Enter your choice: 3
Enter month (YYYY-MM): 2024-07
Enter budget amount: 3000
```



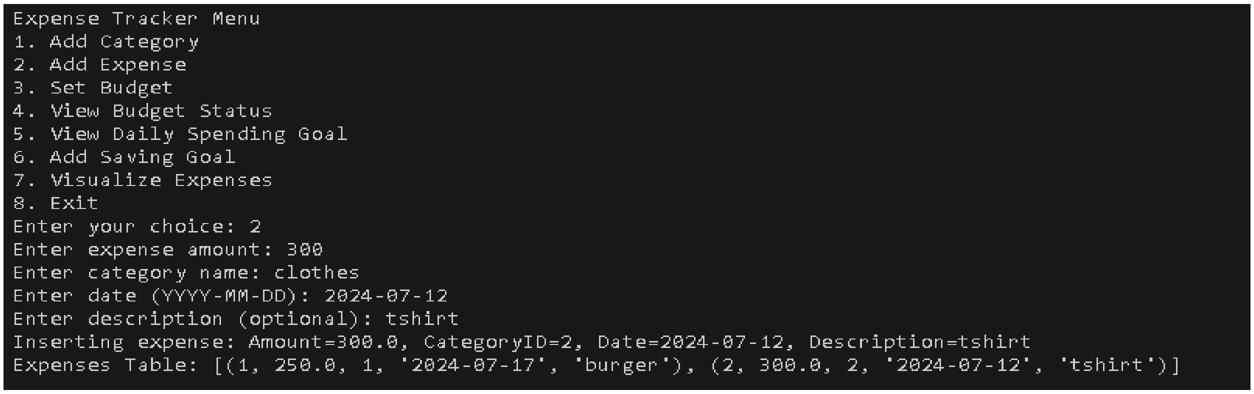
```
Expense Tracker Menu
1. Add Category
2. Add Expense
3. Set Budget
4. View Budget Status
5. View Daily Spending Goal
6. Add Saving Goal
7. Visualize Expenses
8. Exit
Enter your choice: 1
Enter category name: food
```



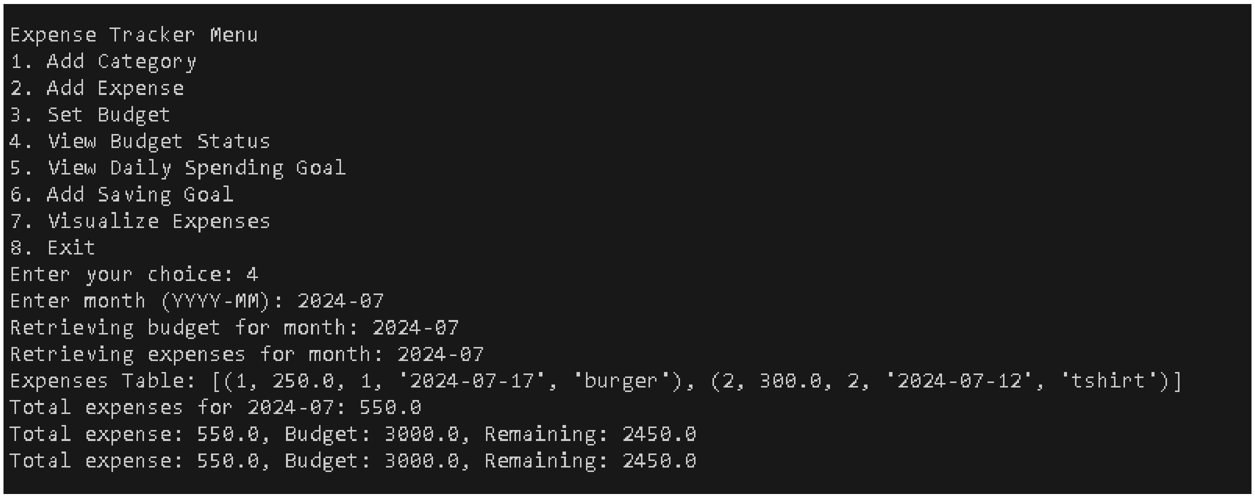
```
Expense Tracker Menu
1. Add Category
2. Add Expense
3. Set Budget
4. View Budget Status
5. View Daily Spending Goal
6. Add Saving Goal
7. Visualize Expenses
8. Exit
Enter your choice: 2
Enter expense amount: 250
Enter category name: food
Enter date (YYYY-MM-DD): 2024-07-17
Enter description (optional): burger
Inserting expense: Amount=250.0, CategoryID=1, Date=2024-07-17, Description=burger
Expenses Table: [(1, 250.0, 1, '2024-07-17', 'burger')]
```



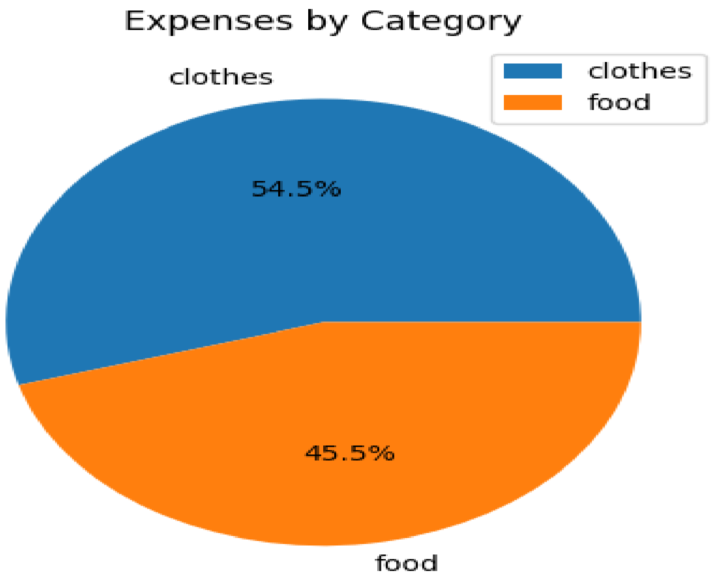
```
Expense Tracker Menu
1. Add Category
2. Add Expense
3. Set Budget
4. View Budget Status
5. View Daily Spending Goal
6. Add Saving Goal
7. Visualize Expenses
8. Exit
Enter your choice: 1
Enter category name: clothes
```

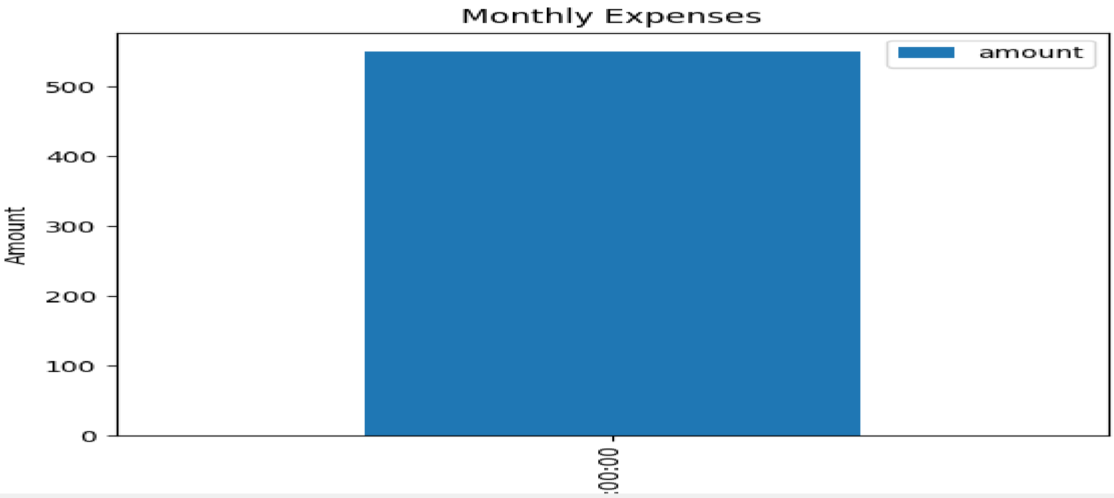


```
Expense Tracker Menu
1. Add Category
2. Add Expense
3. Set Budget
4. View Budget Status
5. View Daily Spending Goal
6. Add Saving Goal
7. Visualize Expenses
8. Exit
Enter your choice: 2
Enter expense amount: 300
Enter category name: clothes
Enter date (YYYY-MM-DD): 2024-07-12
Enter description (optional): tshirt
Inserting expense: Amount=300.0, CategoryID=2, Date=2024-07-12, Description=tshirt
Expenses Table: [(1, 250.0, 1, '2024-07-17', 'burger'), (2, 300.0, 2, '2024-07-12', 'tshirt')]
```



```
Expense Tracker Menu
1. Add Category
2. Add Expense
3. Set Budget
4. View Budget Status
5. View Daily Spending Goal
6. Add Saving Goal
7. Visualize Expenses
8. Exit
Enter your choice: 4
Enter month (YYYY-MM): 2024-07
Retrieving budget for month: 2024-07
Retrieving expenses for month: 2024-07
Expenses Table: [(1, 250.0, 1, '2024-07-17', 'burger'), (2, 300.0, 2, '2024-07-12', 'tshirt')]
Total expenses for 2024-07: 550.0
Total expense: 550.0, Budget: 3000.0, Remaining: 2450.0
Total expense: 550.0, Budget: 3000.0, Remaining: 2450.0
```




Monthly Expenses
amount
500
400
300
200
100
0
Amount
:00:00