EXPENSE SPLITTER

INTRODUCTION

As the world becomes more integrated than ever, monetary transactions and the management of financial costs become vital in daily life as well as in a professional setup. It can be the case of roommates who share rent and utility expenses, students sharing a dinner, or colleagues who are out on a business trip and have to share expenses. In all these cases, expense sharing and tracking who owes how much can become a tangled situation leading to conflicts. More often than not, individuals turn to handwritten notes or computer spreadsheets to tabulate how much each person is in the deficit, which is often very difficult, tiresome, and unproductive. This difficulty, however, increases, especially when there is a larger number of parties involved, or the shares of expenses are uneven.

The problem of dealing with and tracking shared expenses is persistent. The need to use spreadsheets to keep track of expenses owing to conflicts or misunderstandings has been done away with by the development of the Expense Splitter System, which is a mobile application. Not only does this application solve the issue of how to share the expenses, but it also makes sure that conflicts do not arise at all regarding the sharing of costs. This program saves individuals time as well, eliminating any manual labor aimed at computing shared expenses.

The Expense Splitter is fundamentally based on its practicality. The application enables the users to enter the amount of the expense, the number of persons involved, and, if any, the amounts of the contributions each one made. The software can determine each individual’s expense and provides an overview of the parties’ accounts, identifying who owes what or deserves to be paid back. The system can support a range of algorithms for partitioning coffee, for example, into equal parts; into parts proportional to dimensions; and into parts by several user input factors.

Besides the main purpose of the function—the splitting of expenditures—the tool is also aimed at the general public, including users who are not geared towards technology. The user interface (UI) will be simple and user-friendly, allowing input of data, computations, and reports on past expenses to be done with ease. Thus, with these attributes, the Expense Splitter does not require manual computations, which are tedious and susceptible to errors.

This section should introduce what the expense splitter tool does:

* It keeps track and helps manage group expenses effectively.
* The application automates calculation by who spent what and how much one owes or owes to others.
* It makes settlements simple by showing how money is divided in clear ways.
* Importance of Expense Management in Group Settings
* Shared accommodation or travelling with friends often requires dividing up the expenses. Poor management can bring in confusion, and arguments and even damage friendships. While at one time it was common practice to do this manually like paper and pen or spreadsheet, nowadays it is accepted that automation reduces errors and saves time.
* Discuss the following use cases with examples as they are quite common ones.
* Utility bills, groceries and other household expenses are split by roommates.
* Groups of friends on a tour, where each person will contribute to travel, food, and entertainment costs.
* Event planning or office collaboration, which requires sharing costs among different participants.
* Problems with Traditional Approaches
* Traditional approaches, such as paper-based ledgers or spreadsheet-based systems, are typically time-consuming and prone to errors due to constant updating and manual calculations. Examples of common problems include:
* Manual entry errors or forgotten expenses.
* Lack of transparency on who paid for what and how much each one is due.
* Handling complex scenarios, like expenses shared unevenly, can be challenging.
* The role of Python in solving the challenges is discussed next; simplicity, readability, and powerful libraries make it an excellent choice for building a robust user-friendly application.

OBJECTIVE

The main goal of the Expense Splitter is to produce software that makes it easier and clearer to split the expenses among different parties. The tool must facilitate the following:

* Easy input for users to add expenses
* Automate the computation for each person's share.
* Detailed reports of transactions and account balances
* Supports users in multiplicity and expands upwards for future development such as more functionalities or integration purposes.

Specific Objectives

* Ease of Expense Entry: The application should allow users to input expenses with minimal effort. Users should be able to state who paid for what, the amount, and how many people are sharing the cost.
* Automated Calculations: The system should calculate each user’s share and show whether they owe money or are owed money. For more advanced scenarios, it should allow for unequal splits (e.g., if one person paid more for a specific category).
* Transparency and Reporting: Detailed reports must be generated, breaking down each expense, showing who paid for what, and how the amount was split.
* Easy Payback: It should support fast payback to people and settle who pays whom or tell the user how to most efficiently pay the balances.
* Scalable: Though beginning simple, the system needs to be designed in a way so it can easily grow into something with features such as gateway support for payments, multi-currency support, or a mobile app.

BACKGROUND

As the project creates problems and solutions, this project targets the fast-developing tendency to search for digital means for addressing routine issues. Although there are many options when it comes to managing costs, the primary intention of developing the Expense Splitter was to keep it as simple as possible. There’s no need to over-engineer or add additional features to the process. Instead, because the app is built around the fundamental features of Python and employs uncomplicated data platforms, it may be appropriately used to settle common payments in shared circumstances without complex arrangements or setups.

The cost management tool is designed in such a manner that its users will not be required to download extra software that can make the experience time-wasting. Apart from the lack of complex pages, putting the application into practice can help people grasp how basic programming logic can help tackle real-life challenges. By allowing this, the application does not only offer a simple, expensive management platform but one that is effective in addressing conflicts while evenly distributing costs among the group members.

* It is challenging to track the exact amounts.
* Calculating shares might bring in errors.
* Lack of a system for the process to ensure fairness.
* The Emergence of Digital Expense Management Tools

As smartphones and computers became commonplace, digital solutions started to appear. Splitwise, Venmo, and several other apps help in sharing costs by automating the tracking and calculation of the amount owed. Such services offer a more practical solution to the problems posed by the need to share expenses among the parties concerned. Talk about automated payments, instant monitoring, graphical representations, and payment-integrated features on the applications, but note the following drawbacks as well: subscription, limited flexibility and personalization, privacy matters, etc.

Reason for Selection of Python

For the present project, Python was the most appropriate choice given its simplicity, library support, and follower community. Besides, the following aspects have an impact on the use of Python.

• Usability: The structure of sentences in English, while working with Python, is not complicated, making the process of application development fast and easy.

• Available Libraries and Tools: There are quite a few libraries that would be useful for this project. For instance, to plot the data we will use matplotlib, to store information we will use JSON, and to design the graphical user interface we can use PyQt or Kivy.

• Scalability: That is, from simple scripts to more complex applications, Python is great for future development of the tool.

HARDWARE AND SOFTWARE REQUIREMENTS

**Hardware Requirement**

* Processor: Intel Core i3 or higher
* RAM: 4 GB or more
* Storage: 500 MB free space
* Display: 1024x768 resolution
* Input Device: Keyboard and Mouse
* OS: Windows, macOS, or Linux

**Software Requirements**

* Language: Python 3.x
* IDE: PyCharm or Visual Studio Code
* Libraries: Built-in Python modules (`math`, `csv`, `json`)
* Storage: Local file handling
* OS: Cross-platform (Windows, macOS, Linux)

CODING

class ExpenseSplitter:

def \_\_init\_\_(self):

self.expenses = []

self.people = {}

def person(self, name):

if name not in self.people:

self.people[name] = 0.0

def expense(self, amount, contributors):

expense\_details = {}

total\_contributed = 0.0

for persons in contributors:

contribution = float(input(f"Enter the amount paid by {persons}: "))

expense\_details[persons] = contribution

total\_contributed += contribution

if total\_contributed == amount:

for persons in contributors:

self.person(persons)

self.people[persons] += expense\_details[persons]

self.expenses.append((amount, contributors, expense\_details))

def display\_expenses(self):

print("\nExpense Breakdown:")

for amount, contributors, details in self.expenses:

print(f"Expense Amount: {amount:.2f} | Contributors: {', '.join(contributors)}")

for persons, share in details.items():

print(f" {persons} contributed: {share:.2f}")

def display\_balances(self):

print("\nBalances:")

total\_expense = sum(amount for amount,\_,\_ in self.expenses)

num\_people = len(self.people)

average\_expense = total\_expense / num\_people if num\_people > 0 else 0

for persons, amount in self.people.items():

balance = average\_expense - amount

if balance > 0:

print(f"{persons} is owed: {balance:.2f}")

elif balance < 0:

print(f"{persons} owes: {-balance:.2f}")

else:

print(f"{persons} is settled.")

def main():

splitter = ExpenseSplitter()

while True:

print("\n1. Add Expense")

print("2. Show Expense Breakdown")

print("3. Show Balances")

print("4. Exit")

choice = input("Choose an option: ")

if choice == '1':

amount = float(input("Enter the total expense amount: "))

contributors = input("Enter the names of contributors (comma separated): ").split(',')

contributors = [name.strip() for name in contributors]

splitter.expense(amount, contributors)

elif choice == '2':

splitter.display\_expenses()

elif choice == '3':

splitter.display\_balances()

elif choice == '4':

break

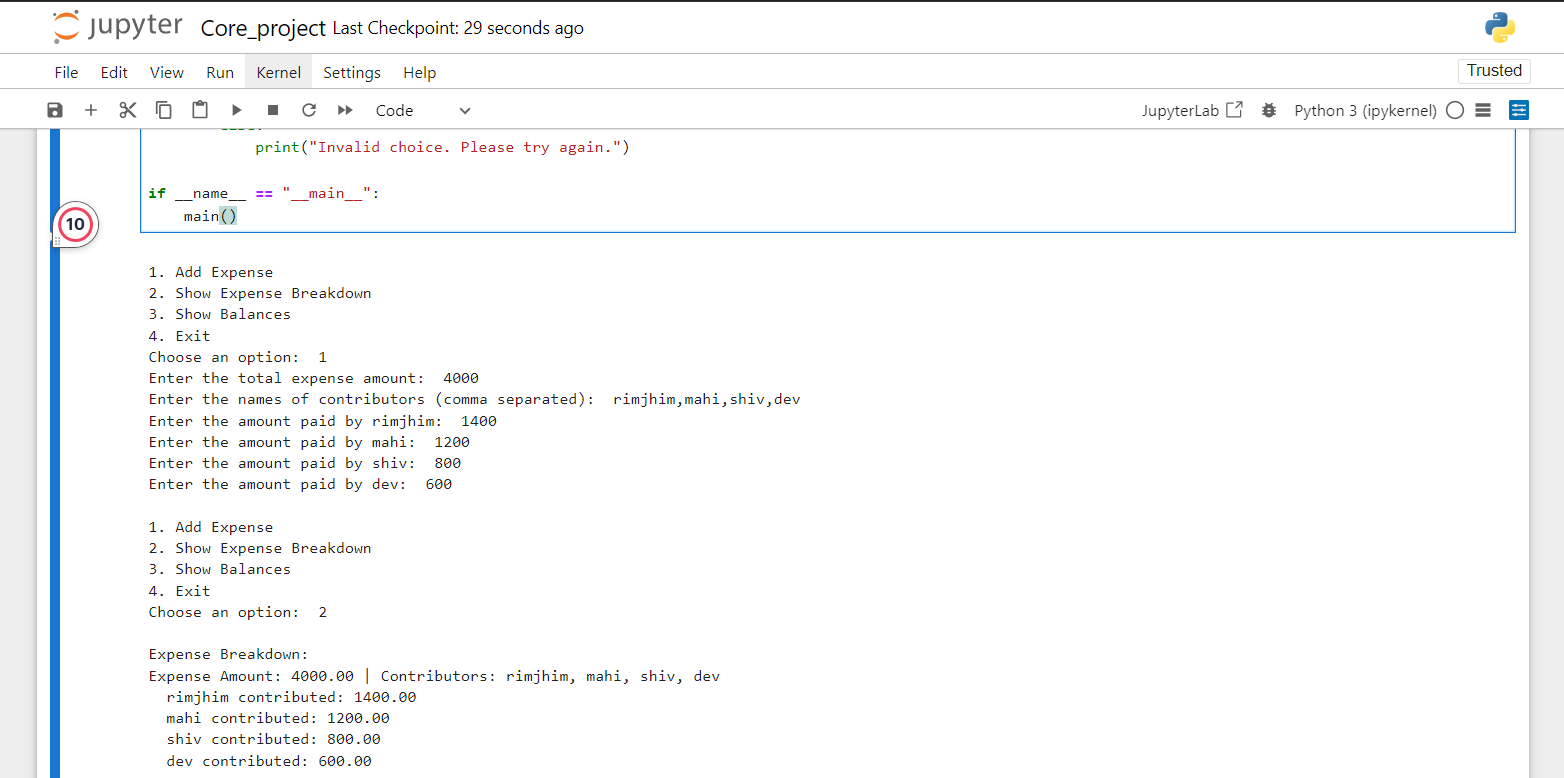
else:

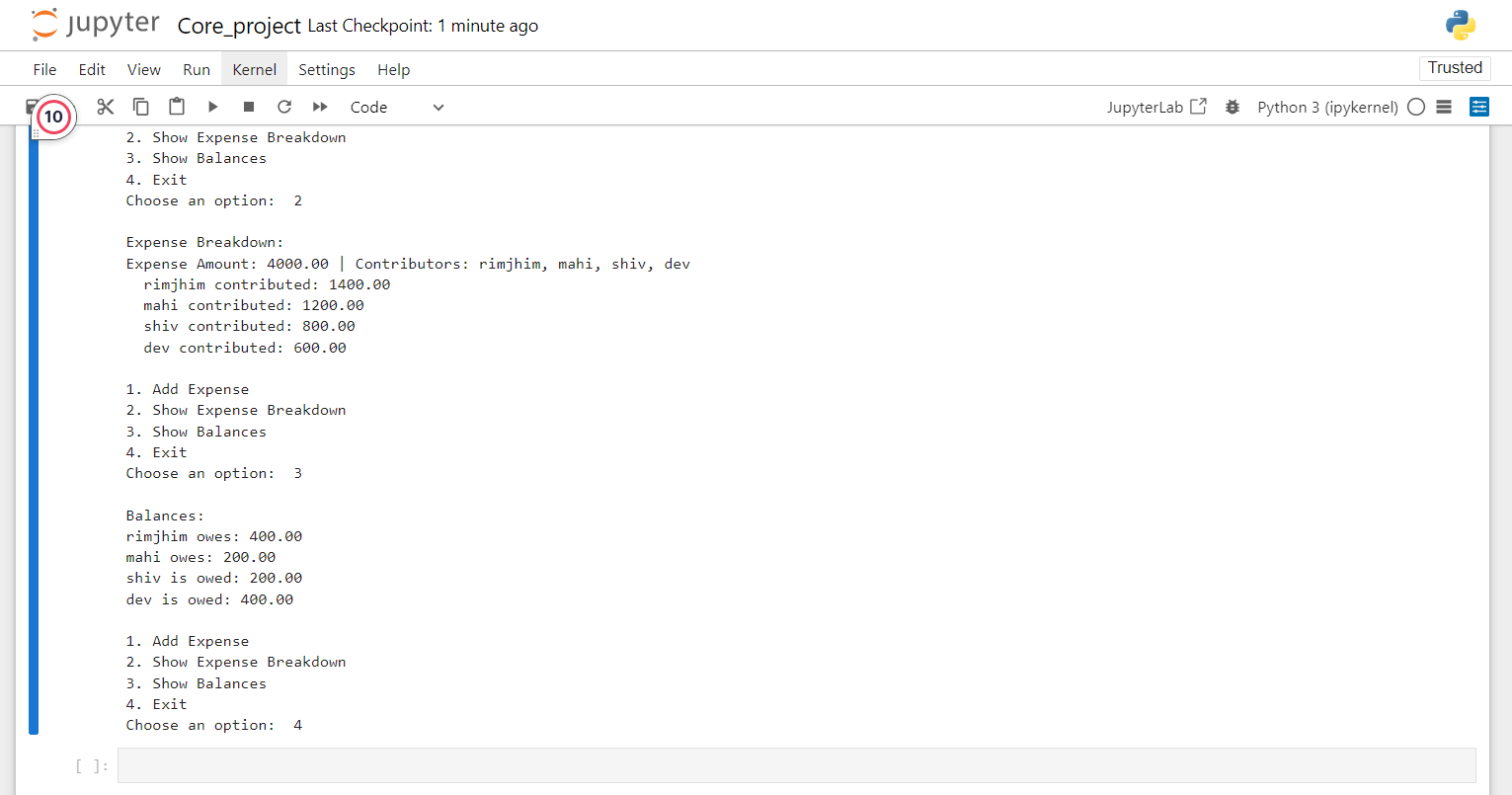
print("Invalid choice. Please try again.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

OUTPUT SCREENSHOT





FUTURE SCOPE

The Expense Splitter application is a good product that may be even better with improvements. Now it has a definite goal of optimization in the future, which includes improving usability, adding new features, and making it more versatile in the sense of usage. Here’s the summary of some focal points that need to be implemented or improved in future development:

* Development of Mobile Application: Including the possibility of using the Expense Splitter with mobile devices (iOS and Android) will make it possible for users to conveniently record expenses when they are away from home or even when on holiday.
* Cloud Integration: Allowing external storage of user data will make it possible to synchronize data on personal computers and laptops. Users would never be out of touch with their records and history of expenses even when they change their system.
* User Account Management: The introduction of user verification, including logging in and passwords, will enable users to save their expenses and access them at different times or on different devices.
* Recurring Expenses Management: The ability to handle recurring expenses (e.g., monthly rent, utilities) and automate the splitting process each month could save time and effort for users in long-term living or shared expense scenarios.
* Voice and AI Assistance: Integrating voice assistants (e.g., Alexa, Google Assistant) to input data and interact with the application could enhance user experience, especially for hands-free use. Additionally, AI algorithms could predict future expenses or suggest optimal splits.
* Support for Multiple Currencies: Adding support for different currencies and exchange rates will make the application usable for international groups or travellers, ensuring accurate cost splitting across borders.
* Integration with Payment Systems: Future versions could allow direct payments or payment tracking by linking the app to popular payment systems (e.g., PayPal, Venmo), making the entire process more seamless.

CONCLUSION

The idea of the developing Expense Splitter app comes from transforming the common yet often overlooked dilemma of the common people: dealing with the splitting of the costs. Whether it be a group of friends out for a meal, coworkers on the same business trip, or tenants sharing rent and utility bills, shared expenses become complicated and invite a lot of mess. One of the main weaknesses of manual methods, such as spreadsheets and calculating by hand, is that it is quite tedious and prone to errors. The Expense Splitter minimizes this process by giving a simple yet effective automated alternative that ensures that expense management is efficient, fair, and transparent.

App powered by Python and offers an elegant and easy-to-use expense divider. The Expense Splitter tries its best to be practical and not aim for more than it should to avoid complicated setups or unnecessary dependence on the integration of third-party applications. The only two features of the application are expense division and expense calculation. It guarantees that when a person enters the total cost, number of participants, and their contribution, there is only that much necessary information to provide an accurate amount of how much he owes or is owed. The application includes several ways of splitting, such as equal splits, splits in percentages, as well as custom ones that suit different scenarios.

In the domain of user experience, the application is friendly, self-explanatory, and uncomplicated. The overall architecture of the application is such that users with little or no technical attributes can operate it with ease. The expense splitter, whether used for splitting day-to-day expenses or even different portions of a single bill, has cut down on the time and chances of making errors.

Apart from that, the application is very adaptable and may be enhanced even more in the future by adding more functionalities. The tool, however, can be improved in additional areas to give it the needed market reach, including cloud hosting, mobile support, strong reporting tools, and management of recurring invoices, among others. All these improvements would make Expense Splitter a more complex tool, which would enable the handling of all the different split expenses accounting, ranging from simple and singular ones to costly shared projects with complex financial management structures.

Expense Splitter is a good application with many uses, especially for people who share expenses. It reduces the complexity and the chances of making mistakes while sharing expenses. This helps people keep their transactions accurate and simple. Such an application helps handle finances better and reduces misunderstandings about the management of shared expenses. With the growth prospects of this application, the application has the potential to be very useful for the management of personal and group finances, addressing the increasing demand for financial solutions in a global environment.

An essential first step in automating group financial management is the Expense Splitter. The demand for this kind of tool will only increase as more people live together and collaborate in groups.

REFERENCES AND BIBLIOGRAPHY

* Expense Management Systems: Features and Architectures: This research paper examines the architecture of expense tracker applications, focusing on modules like data visualization, reporting, and backend integration for secure financial data storage and processing. (IJNRD)
* Advanced Features in Expense Splitter Applications: This study from the International Research Journal of Engineering and Technology (IRJET) highlights functionalities like multi-user access, role-based permissions, and advanced sorting mechanisms for transaction data. (IRJET)
* Expense Splitter Algorithms and Data Analytics: A publication exploring algorithms for equitable expense sharing and leveraging data mining techniques to identify user spending patterns for financial analysis. (IJCESR).
* Integration of Data Visualization in Financial Management Apps: This article focuses on the importance of visual tools like charts and graphs in helping users manage their expenses and derive insights into spending habits. (IJNRD)