MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
A Constituent Institution of Manipal University

Expense Tracker:

A Python-based Expense management Application

By:

T Prajwal Prabhu :	**210953002**
Joshita Bolisetty :	**210953070**
Shubhanshu Verma :	**210953072**

Index

Abstract

In cutting-edge dynamic economic landscape, powerful cost management is critical for reaching non-public monetary stability and securing a brighter destiny. Traditional expense tracking strategies, such as manual records or spreadsheets, frequently show to be inefficient and vulnerable to mistakes, hindering individuals from gaining treasured insights into their spending patterns. To deal with this task, the Expense Tracker mission proposes the development of a human-friendly software using Python's abilities to streamline expense management for people. This modern tool will empower customers to seamlessly report their daily costs, categorize them effectively, and generate comprehensive reports, presenting a clear information of their spending habits and permitting informed monetary decisions.

Introduction

Financial well-being is an important thing of private success, and powerful expense tracking serves as a cornerstone of achieving monetary balance. Conventional expense tracking strategies, such as handwritten notes or spreadsheets, can be cumbersome and time-ingesting, regularly leading to faulty information and a lack of readability into spending styles. This lack of knowledge can result in overspending, overlooked financial savings opportunities, and ultimately, economic strain.

Background information

The advent of the electronics era and the big adoption of PCs and smartphones have opened up a world of opportunities for developing innovative solutions to deal with personal financial demanding situations. Expense monitoring applications have emerged as a popular device for individuals looking for to gain control over their budget. These applications offer a convenient and user-friendly platform to record, categorize, and examine prices, offering treasured insights into spending behaviour and permitting informed monetary selections.

By leveraging the strength of Python, the Expense Tracker mission aims to create a user-centric software that empowers individuals to take charge of their budget. This application will offer a seamless interface for recording costs, a established system for categorization, and complete reporting abilities, remodelling expense management into a simple and insightful experience.

Python concepts used

The Python concepts used are:

- **Tkinter** for application’s Graphical User Interface (GUI)
  The frontend of the Expense Tracker application is built using the tkinter library, which is the standard GUI toolkit for Python. In the project, tkinter is employed to create a user-friendly interface with various widgets such as labels, buttons, entry fields, and option menus. The GUI allows users to interact with the application by entering expense details, selecting categories, and navigating through different functionalities seamlessly. The structure of the GUI is modular, providing a clear layout for users to add expenses and view reports.

- **SQLite** for database management and storage of expense data
  The application utilizes the SQLite database to manage and store expense-related data. Two tables are defined: User for storing user credentials, and Expenses for recording expense details. The sqlite3 module in Python is employed to establish a connection to the database, create tables, and execute SQL queries. This ensures the persistent storage of user accounts and expense records.

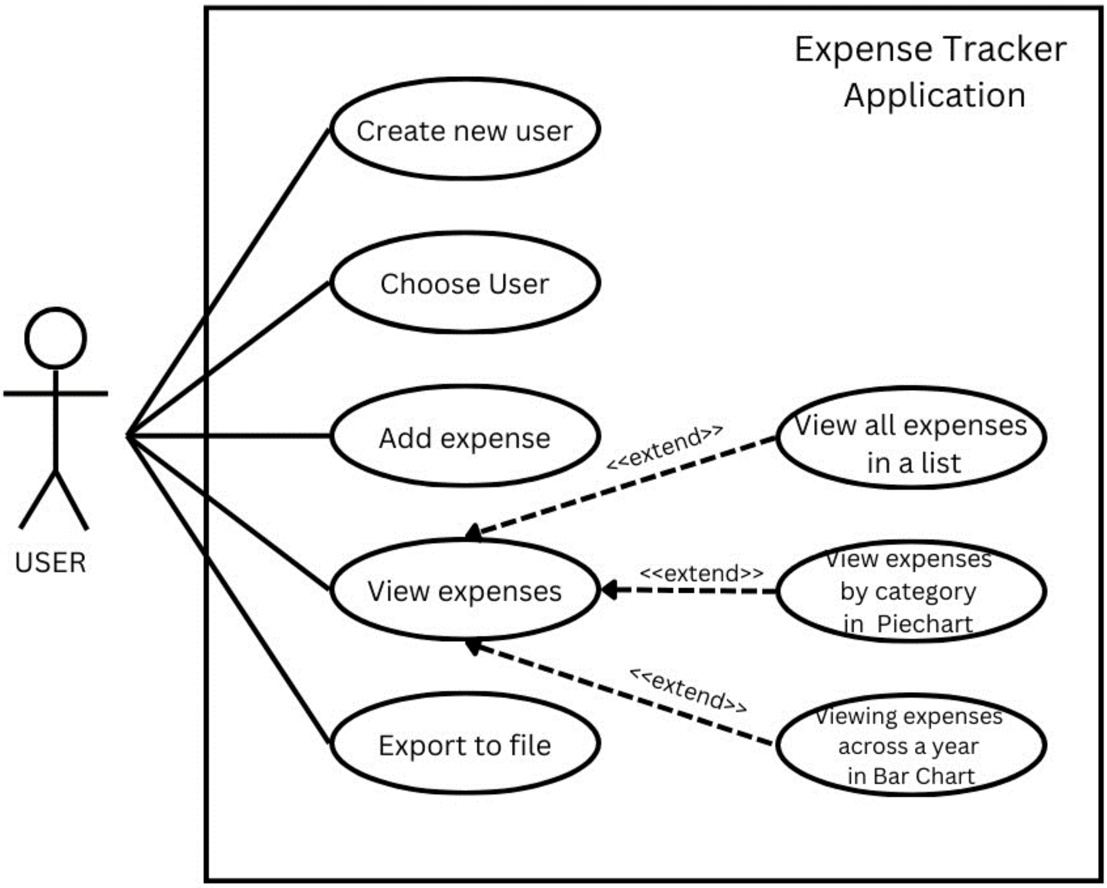
- **Matplotlib** for representing expenses in a visual manner
  To provide users with a visual representation of their expenses, the Matplotlib library is integrated into the project. Matplotlib allows the generation of various charts and graphs. In the project, it is used to create bar charts and pie charts that visually represent monthly and categorical expenses. These visualizations offer users a quick and informative overview of their spending patterns.

- **File handling** for reading and writing data to external files
  File handling is implemented to export expense data to external files in CSV format. The csv module is used for efficient reading and writing of CSV files. This functionality allows users to save their expense records locally and analyze them using external tools.

- **Error handling** to enhance robustness of the application
  Error handling is implemented throughout the application to enhance its robustness. Custom exceptions, such as FutureDateException, are defined and raised when specific conditions are not met. This ensures that the application gracefully handles errors and provides informative error messages to users, enhancing the overall user experience.

Methodology

Use case diagram


Expense Tracker Application
USER
Create new user
Choose User
Add expense
View expenses
Export to file
<<extend>>
View all expenses in a list
<<extend>>
View expenses by category in Piechart
<<extend>>
Viewing expenses across a year in Bar Chart

As it can be seen from the above use case diagram for the expense tracker application, the user is provided with the features like creating a new user. choosing the user, adding an expense, viewing the expense, and exporting the entire expense list to a csv file. Further, while viewing expenses, the user has the option to view them as a list, as a pie chart by category, or as a bar chart of the monthly expenses across the selected year.

The Expense class handles database interactions, while the ExpenseTracker class manages the Tkinter-based graphical user interface. Exception handling is implemented, including a custom exception (FutureDateException) to prevent the selection of future dates.

The structures of the tables used for storing the data in SQLite are:

1. User table

   It includes the following attributes, their data types and constraint/s if any:

Attribute Name	Data Type	Constraint
userid	integer	primary key (auto increment)
username	text	not null
password	text	not null

2. Expenses table

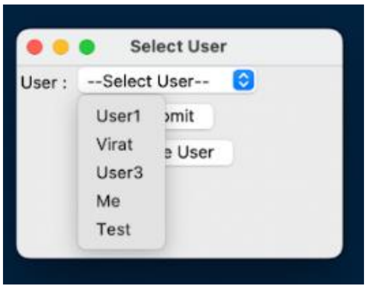
   Following are the attributes, their data types and special feature if any:

Attribute Name	Data Type	Constraint
user	text	not null
id	integer	primary key (auto increment)
category	text	not null
name	text	not null
amount	real	not null
date	date	not null

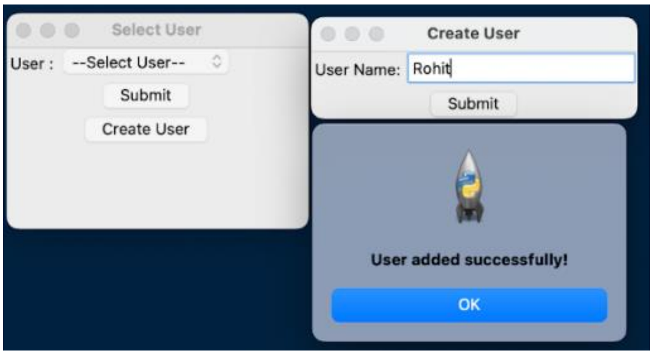
Implementation

This section of the report includes a comprehensive collection of screenshots that showcase the application's features and functionalities.

- Viewing users list

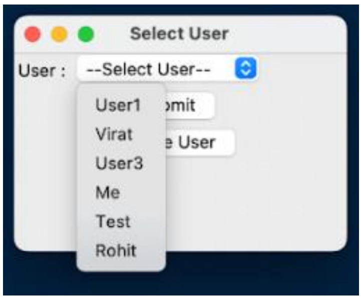


- Creating a new user

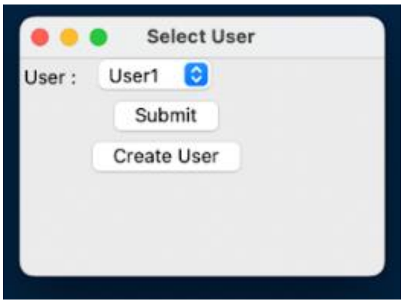


User Name: Rohit

User added successfully!

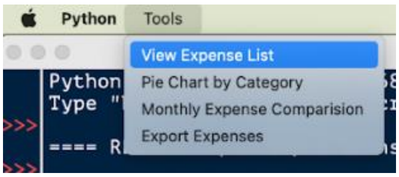


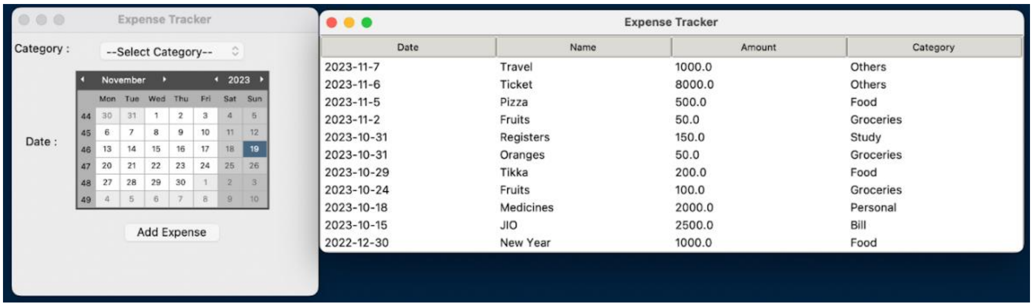
- Logging into the system by selecting a user



User : User1

- View list of all expenses



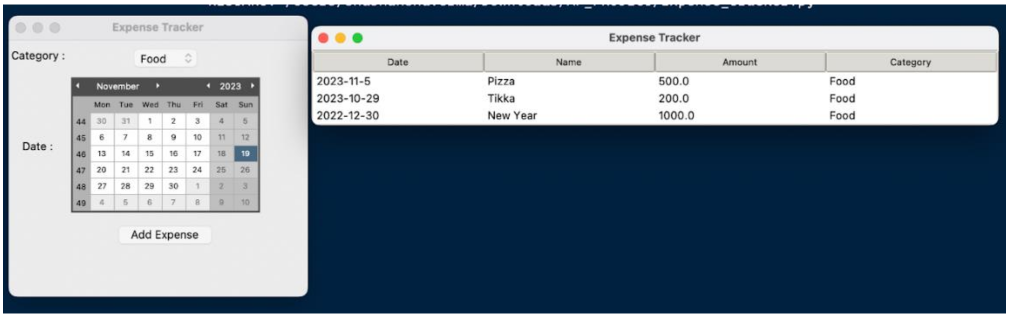


Category : --Select Category--

Date :

Date	Name	Amount	Category
2023-11-7	Travel	1000.0	Others
2023-11-6	Ticket	8000.0	Others
2023-11-5	Pizza	500.0	Food
2023-11-2	Fruits	50.0	Groceries
2023-10-31	Registers	150.0	Study
2023-10-31	Oranges	50.0	Groceries
2023-10-29	Tikka	200.0	Food
2023-10-24	Fruits	100.0	Groceries
2023-10-18	Medicines	2000.0	Personal
2023-10-15	JIO	2500.0	Bill
2022-12-30	New Year	1000.0	Food

- View list of expenses of a particular category

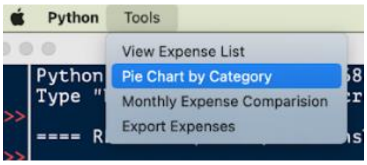


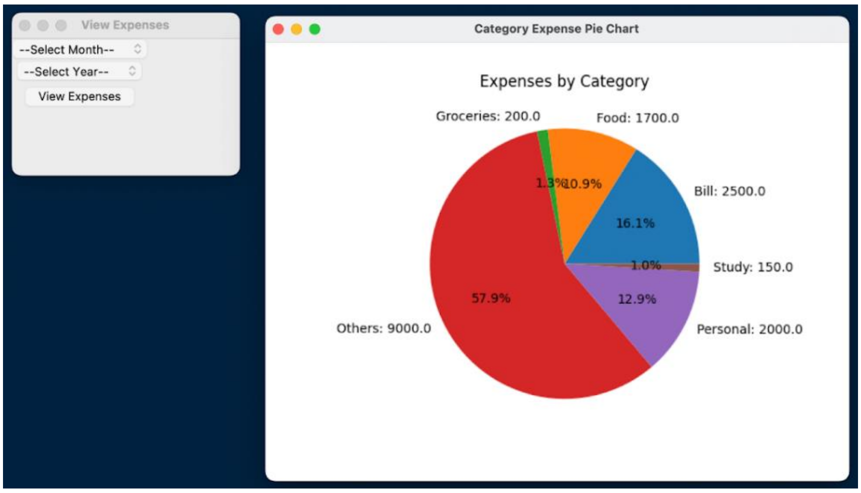
Category : Food

Date :

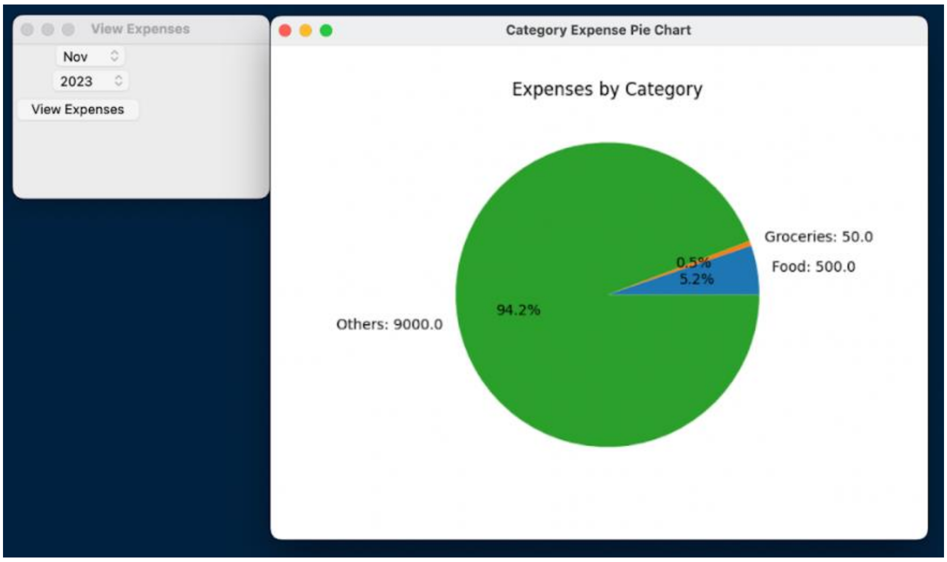
Date	Name	Amount	Category
2023-11-5	Pizza	500.0	Food
2023-10-29	Tikka	200.0	Food
2022-12-30	New Year	1000.0	Food

- View pie chart of all expenses

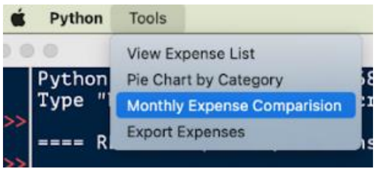


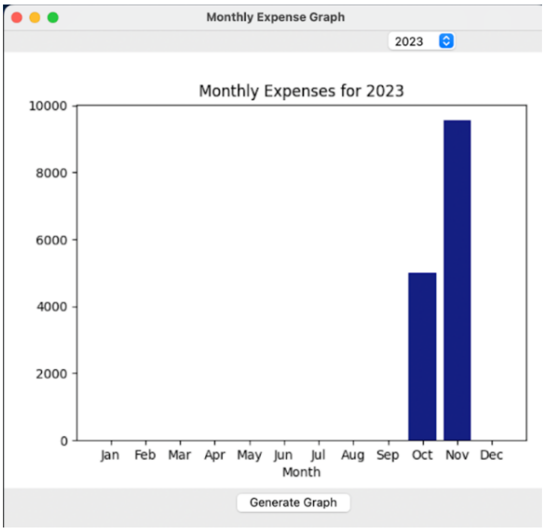


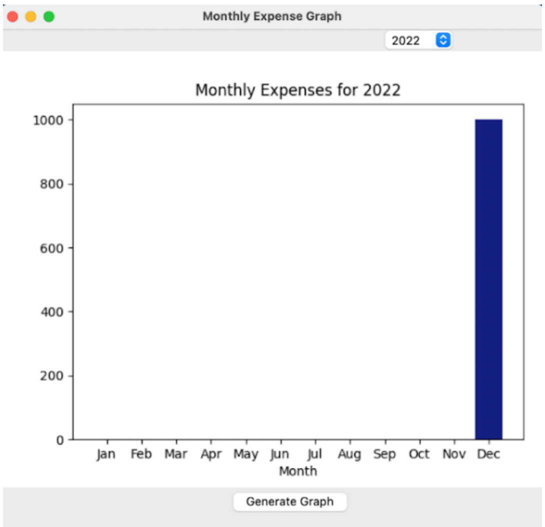
- View pie chart of expenses of in a particular month and year



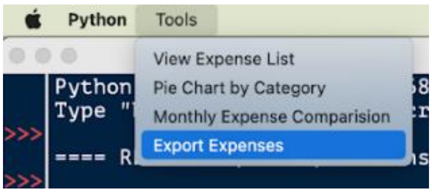
- View bar graph of expenses of a chosen year

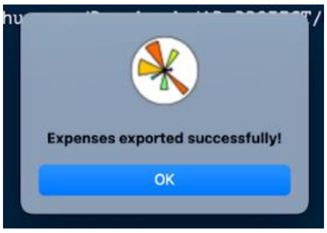


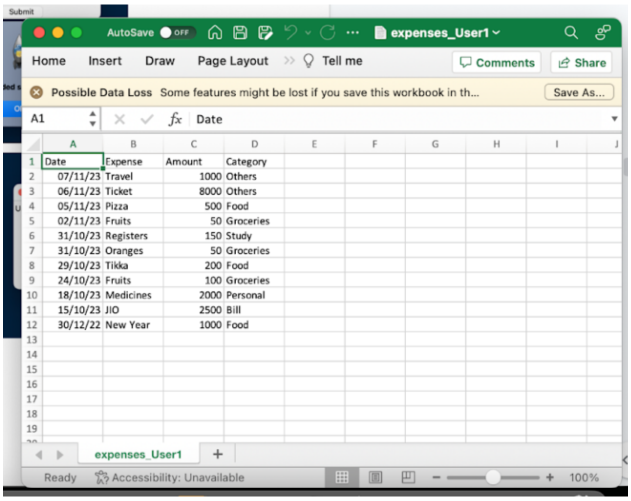




- Exporting expenses to a file




Expenses exported successfully!

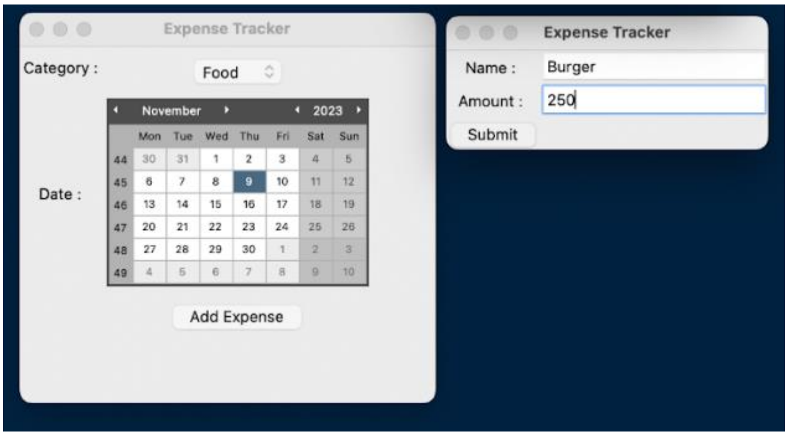


Possible Data Loss Some features might be lost if you save this workbook in th...

A1 Date

	A	B	C	D
1	Date	Expense	Amount	Category
2	07/11/23	Travel	1000	Others
3	06/11/23	Ticket	8000	Others
4	05/11/23	Pizza	500	Food
5	02/11/23	Fruits	50	Groceries
6	31/10/23	Registers	150	Study
7	31/10/23	Oranges	50	Groceries
8	29/10/23	Tikka	200	Food
9	24/10/23	Fruits	100	Groceries
10	18/10/23	Medicines	2000	Personal
11	15/10/23	JIO	2500	Bill
12	30/12/22	New Year	1000	Food

- Adding an expense



Category : Food

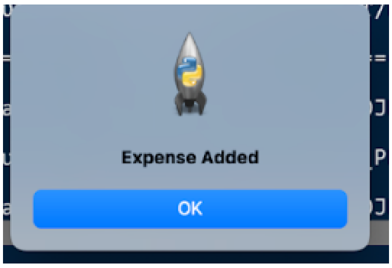
Date :

November 2023

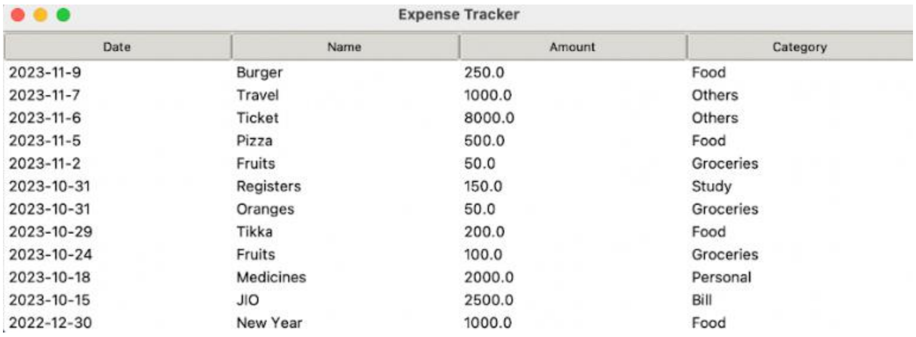
	Mon	Tue	Wed	Thu	Fri	Sat	Sun
44	30	31	1	2	3	4	5
45	6	7	8	9	10	11	12
46	13	14	15	16	17	18	19
47	20	21	22	23	24	25	26
48	27	28	29	30	1	2	3
49	4	5	6	7	8	9	10

Name : Burger

Amount : 250



Expense Added

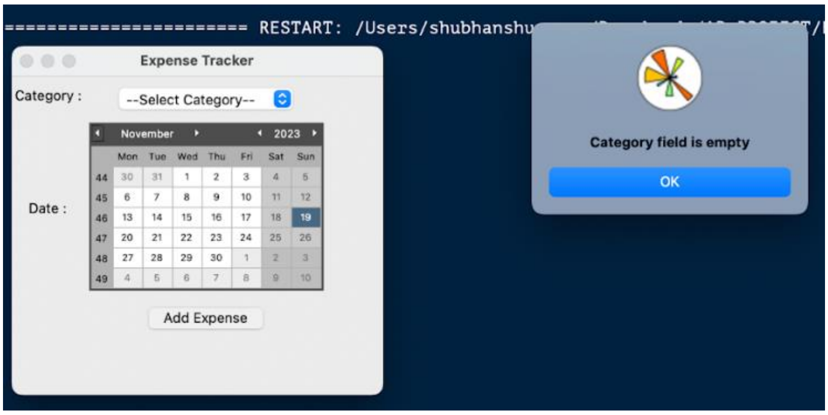


Expense Tracker

Date	Name	Amount	Category
2023-11-9	Burger	250.0	Food
2023-11-7	Travel	1000.0	Others
2023-11-6	Ticket	8000.0	Others
2023-11-5	Pizza	500.0	Food
2023-11-2	Fruits	50.0	Groceries
2023-10-31	Registers	150.0	Study
2023-10-31	Oranges	50.0	Groceries
2023-10-29	Tikka	200.0	Food
2023-10-24	Fruits	100.0	Groceries
2023-10-18	Medicines	2000.0	Personal
2023-10-15	JIO	2500.0	Bill
2022-12-30	New Year	1000.0	Food

- Exception handling

  - Not selecting any field while adding an expense



```
======================= RESTART: /Users/shubhanshu
```

Expense Tracker

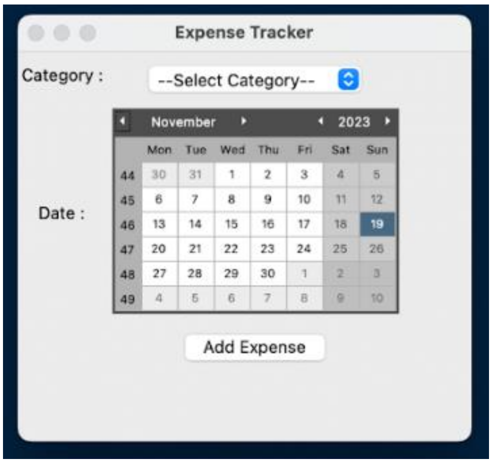
Category : --Select Category--

Date :

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
44	30	31	1	2	3	4	5
45	6	7	8	9	10	11	12
46	13	14	15	16	17	18	19
47	20	21	22	23	24	25	26
48	27	28	29	30	1	2	3
49	4	5	6	7	8	9	10

Category field is empty

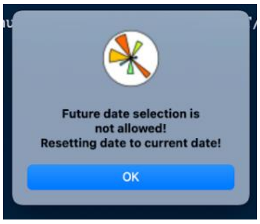
- Selecting a future date to enter an expense



Category : --Select Category--

Date :

	November					2023	
	Mon	Tue	Wed	Thu	Fri	Sat	Sun
44	30	31	1	2	3	4	5
45	6	7	8	9	10	11	12
46	13	14	15	16	17	18	19
47	20	21	22	23	24	25	26
48	27	28	29	30	1	2	3
49	4	5	6	7	8	9	10



Future date selection is
not allowed!
Resetting date to current date!

Results

The Expense Tracker application has delivered a robust and user-friendly platform for tracking and managing personal finances. Through its intuitive interface, users can seamlessly record their expenses, categorize spending, and visualize their financial data. The application's key outcomes include:

- Simplified Expense Tracking: Users can effortlessly input and categorize their costs, enabling a greater prepared method to economic control.
- Insightful Visualization: The integration of matplotlib facilitates the introduction of complete pie charts and bar graphs, imparting customers a clean review of their spending styles.
- User-Centric Design: Tkinter and tkcalendar have contributed to an intuitive and responsive person interface, taking into consideration easy navigation and efficient statistics entry.

Conclusion

The Expense Tracker utility serves as an efficient and user-friendly solution for dealing with personal finances. Through its complete functionalities and consumer-centric layout, it addresses the critical need for individuals and small organizations to monitor their expenses efficiently. Moving ahead, capability enhancements could consist of additional features for finances planning and goal setting.

References

- Matplotlib Pyplot (w3schools.com) for plotting charts and graphs
- Python File Open (w3schools.com) for file handling
- Python Try Except (w3schools.com) for exception handling
- tkinter — Python interface to Tcl/Tk — Python 3.12.0 documentation for frontend
- tkinter - How can we restrict future date selection from tkCalender Date Entry picker in Python? - Stack Overflow for calendar options
- Python SQLite - GeeksforGeeks for database operations