

SRINIVAS UNIVERSITY
INSTITUTE OF ENGINEERING & TECHNOLOGY
MUKKA, MANGALURU



SPEND ANALYSER TOOL
(USING PYTHON)

For the Academic year 2024-2025

Submitted by

1. MADHU GOWDA K S	01SU23AI043
2. MADHU DUNDAPPA	01SU23AI042
3. MANSIHA ACHARYA	01SU23A1046
4. SUNITHA	01SU23AI095

Submitted to

Ms.Priyanka Professor



CONTENT

❖ **ABSTRACT**

❖ **INTRODUCTION**

❖ **SOURCE CODE**

❖ **OUTPUT**

❖ **CONCLUSION**

❖ **FUTURE SCOPE**

❖ **REFERENCE**

ABSTRACT

The Expense Analyzer project is a Python-based application designed to help users track, manage, and analyze their expenses efficiently. The application offers functionalities such as adding new expenses with details like name, amount, and category, viewing a list of all expenses, calculating the total expenses incurred, and analyzing expenses by category.

Users can interact with the application through a command-line interface, which provides an intuitive menu-driven system. Additionally, the project utilizes the Matplotlib library to generate informative visualizations, such as bar charts, to represent expenses across different categories.

The Expense Analyzer project serves as a practical tool for individuals seeking to gain insights into their spending habits and make informed financial decisions.

Through its implementation, users can enhance their proficiency in Python programming, object-oriented design, user interface development, and data visualization techniques

INTRODUCTION

In an era marked by the relentless pursuit of efficiency and optimization, personal finance management stands as a cornerstone of responsible living. In this digital age, where transactions occur seamlessly with a swipe or a click, the need for effective expense tracking and analysis has never been more pronounced. Recognizing this imperative, the Expense Analyzer project emerges as a robust solution crafted with Python—a versatile programming language renowned for its simplicity and power.

At its core, the Expense Analyzer project seeks to empower individuals with the tools necessary to navigate the complex landscape of personal finances. By offering a comprehensive suite of features encompassing expense recording, visualization, and analysis, this project endeavors to demystify the process of financial management, transforming it from an arduous chore into an enlightening journey of self-awareness and empowerment.

In this introductory exploration, we delve into the genesis of the Expense Analyzer project, its underlying motivations, and the myriad benefits it confers upon its users. From its humble beginnings as a conceptualization born of necessity to its realization as a fully-fledged Python application, the journey of the Expense Analyzer project epitomizes the symbiotic relationship between technology and human endeavor.

Through this project, we aim to explore the intersection of mathematics, programming, and visual art. Fractal geometry, with its emphasis on self-replication and scalability, offers a captivating playground for computational creativity. By implementing this fractal tree code, we not only gain practical experience in algorithmic design but also deepen our appreciation for the elegance of recursive algorithms and their ability to generate complex, organic forms.

Join us as we embark on a captivating journey through the realms of personal finance, guided by the ingenuity of Python programming and the boundless possibilities it offers. Through the lens of the Expense Analyzer project, we unravel the intricacies of expense tracking, unveil the secrets hidden within financial data, and empower individuals to take control of their financial destinies with confidence and clarity.

SOURCE CODE :

```
import matplotlib.pyplot as plt

class Expense:
    def __init__(self, name, amount, category):
        self.name = name
        self.amount = amount
        self.category = category

class ExpenseTracker:
    def __init__(self):
        self.expenses = []

    def add_expense(self, expense):
        self.expenses.append(expense)

    def view_expenses(self):
        if not self.expenses:
            print("No expenses recorded yet.")
        else:
            print("List of expenses:")
            for index, expense in enumerate(self.expenses, start=1):
                print(f"{index}. {expense.name}: ${expense.amount} ({expense.category})")

    def total_expenses(self):
        return sum(expense.amount for expense in self.expenses)

    def expenses_by_category(self):
```

```
categories = {}  
for expense in self.expenses:  
    categories[expense.category] = categories.get(expense.category, 0) + expense.amount  
return categories
```

```
def plot_expenses_by_category(self):  
    expenses_by_category = self.expenses_by_category()  
    categories = list(expenses_by_category.keys())  
    amounts = list(expenses_by_category.values())  
  
    plt.figure(figsize=(10, 6))  
    plt.bar(categories, amounts, color='skyblue')  
    plt.xlabel('Categories')  
    plt.ylabel('Amount ($)')  
    plt.title('Expenses by Category')  
    plt.xticks(rotation=45, ha='right')  
    plt.tight_layout()  
    plt.show()
```

```
def main():  
    tracker = ExpenseTracker()  
  
    while True:  
        print("\nExpense Tracker Menu:")  
        print("1. Add Expense")  
        print("2. View Expenses")  
        print("3. View Total Expenses")  
        print("4. View Expenses by Category")  
        print("5. View Expenses by Category (Graph)")  
        print("6. Quit")
```

```
choice = input("Enter your choice: ")

if choice == "1":
    name = input("Enter expense name: ")
    amount = float(input("Enter expense amount: "))
    category = input("Enter expense category: ")
    expense = Expense(name, amount, category)
    tracker.add_expense(expense)
    print("Expense added successfully!")
elif choice == "2":
    tracker.view_expenses()
elif choice == "3":
    total = tracker.total_expenses()
    print(f"Total expenses: ${total}")
elif choice == "4":
    expenses_by_category = tracker.expenses_by_category()
    print("Expenses by Category:")
    for category, amount in expenses_by_category.items():
        print(f"{category}: ${amount}")
elif choice == "5":
    tracker.plot_expenses_by_category()
elif choice == "6":
    print("Exiting...")
    break
else:
    print("Invalid choice. Please try again.")

if __name__ == "__main__":
    main()
```

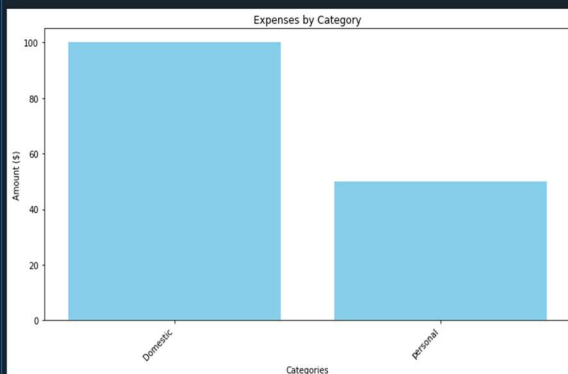
OUTPUT

```
In [1]: runfile('D:/SEMINAR PROJECTS/CS MINI PROJECT EXPENSE ANALYSER.py', wdir='D:/SEMINAR PROJECTS')
```

```
Expense Tracker Menu:  
1. Add Expense  
2. View Expenses  
3. View Total Expenses  
4. View Expenses by Category  
5. View Expenses by Category (Graph)  
6. Quit  
Enter your choice: 1  
Enter expense name: Vegetables  
Enter expense amount: 100  
Enter expense category: Domestic  
Expense added successfully!
```

```
Expense Tracker Menu:  
1. Add Expense  
2. View Expenses  
3. View Total Expenses  
4. View Expenses by Category  
5. View Expenses by Category (Graph)  
6. Quit  
Enter your choice: 1  
Enter expense name: Chats  
Enter expense amount: 50  
Enter expense category: personal  
Expense added successfully!
```

```
Expense Tracker Menu:  
1. Add Expense  
2. View Expenses  
3. View Total Expenses  
4. View Expenses by Category  
5. View Expenses by Category (Graph)  
6. Quit  
Enter your choice: 5
```



FUTURE SCOPE

The Expense Analyzer project lays a solid foundation for future enhancements and expansions, offering a myriad of opportunities to further elevate its functionality, usability, and impact. As technology evolves and user needs evolve alongside it, there are several avenues through which the project can continue to grow and adapt to meet emerging demands. Here are some potential areas for future development:

1. ****Enhanced User Interface****: While the current version of the Expense Analyzer project features a command-line interface, there is ample scope for developing a more intuitive and user-friendly graphical user interface (GUI). Transitioning to a GUI framework such as Tkinter or PyQt would enable users to interact with the application more seamlessly, thereby enhancing overall usability.
2. ****Data Persistence and Backup****: Implementing data persistence mechanisms such as database integration or file-based storage would allow users to store their expense data securely and access it across sessions. Additionally, incorporating automated backup functionalities would safeguard against data loss and provide peace of mind to users.
3. ****Advanced Analytics****: Introducing advanced analytical features such as trend analysis, predictive modeling, and budget forecasting would enable users to gain deeper insights into their spending patterns and make informed financial decisions. Leveraging machine learning algorithms could help identify trends and patterns in expense data, facilitating proactive financial planning.
4. ****Integration with Financial Services****: Integrating the Expense Analyzer project with external financial services and APIs could provide users with real-time access to their bank accounts, credit card transactions, and investment portfolios. This integration would streamline expense tracking and ensure that users have a comprehensive view of their financial landscape..
5. ****Advanced Analytics****: Introducing advanced analytical features such as trend analysis, predictive modeling, and budget forecasting would enable users to gain deeper insights into their spending patterns and make informed financial decisions. Leveraging machine learning algorithms could help identify trends and patterns in expense data, facilitating proactive financial planning.

CONCLUSION

The Expense Analyzer project represents a significant step towards empowering individuals with the tools and insights necessary to navigate the complex landscape of personal finance. Through its intuitive interface, robust functionality, and data-driven approach, the project has emerged as a valuable asset for users seeking to gain control over their financial destinies.

Throughout this journey, we have witnessed the transformative power of Python programming, which has served as the backbone of the project, enabling seamless interaction, efficient data processing, and insightful analysis. From its inception as a humble concept to its realization as a fully-fledged application, the Expense Analyzer project stands as a testament to the ingenuity and versatility of Python in addressing real-world challenges.

As we reflect on the accomplishments of the Expense Analyzer project, it becomes evident that its impact extends far beyond mere expense tracking. By fostering a culture of financial awareness, responsibility, and empowerment, the project has empowered individuals to make informed decisions, cultivate healthy spending habits, and work towards financial wellness.

Looking ahead, the future holds immense promise for the Expense Analyzer project. With opportunities for expansion, innovation, and collaboration abound, the project is poised to evolve in response to emerging trends, user feedback, and technological advancements. By embracing these opportunities and staying true to its mission of promoting financial literacy and empowerment, the Expense Analyzer project will continue to thrive and make a meaningful difference in the lives of its users.

REFERENCE

- ❖ <https://wikipedia.org>
- ❖ <https://chatgpt.com>
- ❖ <https://google.com>