Project Title: Expense Tracker

Author(s): Burke Havranek, Gaurav Koratagere, Ajey Sasimugunthan

Northeastern University

EECE 2140. Computing Fundamentals for Engineers

Fall 2024

Date: November 21st, 2024

Abstract

Examining the intent of this project, the goal was to ensure that people can track their expenses effectively and be able to budget their money efficiently without sacrificing that much. Looking at apps such as YNAB or pocket wallet, these services require the user to spend money on a subscription and does not implement any feature to encourage people to save more of their money. Similar to these apps, our program takes inputs from the user in order to figure out how much they are spending each month and also see how much of it goes towards their bills. One of these inputs asks the user about the money spent towards luxury items. If the user is saving money each month, our program will show them how much they could make by saving that amount each month into the S&P 500 over the course of thirty years with roughly a seven percent return. Typically, the user will see some huge number because of the power of compound interest. Not only will users understand why they should invest their excess money, but they will be more motivated to live within their means and cut out excessive spending in order to have more money towards retirement. Towards the end of this project, our group came to realize that time and lack of knowledge were the two biggest limiting factors since those two factors are needed to be able to create a platform similar to YNAB or Mint. While they may be limiting factors, the ideas and work done so far in this project serve as a foundation for an even more complete project down the line that can be used by millions of people to budget their money and also be encouraged to invest their money at the same time.

Contents

Chapter 1

Introduction

1.1 Background

Provide context or background information necessary to understand the project.

1.2 Problem Statement

A clear statement of the problem or challenge the project addresses.

1.3 Objectives

Specify the objectives or goals of the project.

1.4 Scope

Define the boundaries of the project, including what is and isns outputs here. Add screenshots and code snippets as needed.

Chapter 2

Discussion and Future Work

2.1 Development Environment

Similar to most assignments and mini projects done in lecture, this program was coded in the same Python environment through Spyder since it was familiar and fairly easy to use. One library that was utilized was matplotlib in order to create a graph for users to see the power of compound interest. It is used to display a user's growth in their money if excess income was invested in the S&P 500 over the course of thirty years.

2.2 Data Collection and Preparation

Most of the inputs were collected from the user using the input command learned in lecture. This was needed since the program relies heavily on data from the user and the program does the analysis for them. With these inputs such as monthly income for example, they were saved as floats so that they can be used in later analysis to calculate their left over monthly income and figure out if they were saving money each month or not.

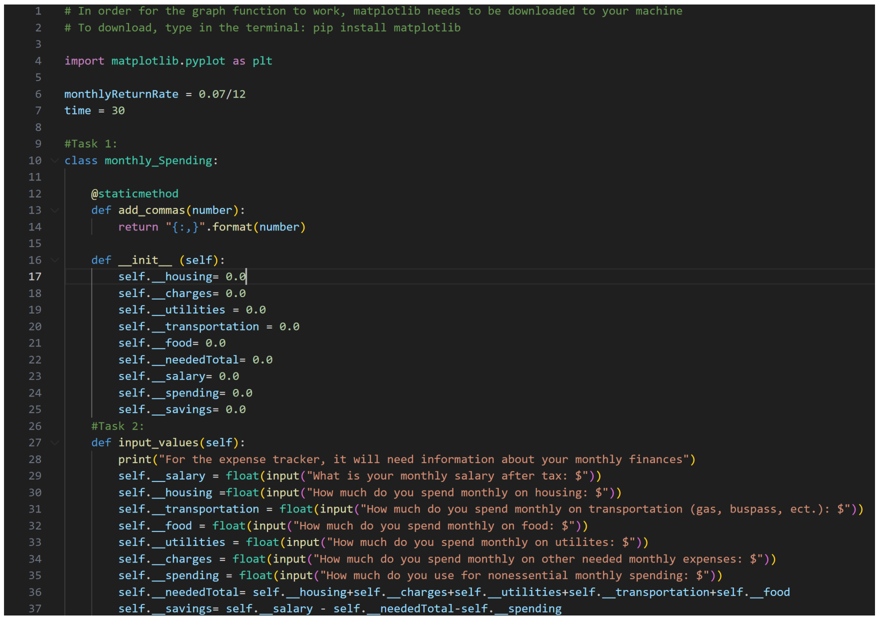
2.3 Implementation Details

The code breaks into three distinct sections, data storage, data gathering and data processing. The code starts by creating a class. In this class we defined the core information for the class. We added placeholder float values so that the data type would be interpreted as a float for even if an int is put in to make sure that none of the math breaks later on. For data gathering, a function was added inside that class to set the elements of the class to the inputs a user makes. Lastly we made functions that performed operations and displayed the operations on the data stored. The first function was just a simple graphing function, this would calculate and display the exponential growth of your money if all of the savings was put into the S&P 500. Just in case no money was being saved a saving function was needed to check and if no money was being saved, it could make a recommendation on how to possibly save. The last function in the class was a function to find simple totals of values like housing over a year, spending over a year and travel cost over a year. We also had a static method so that the numbers displayed would be shown with commas and a main function that put it all together.

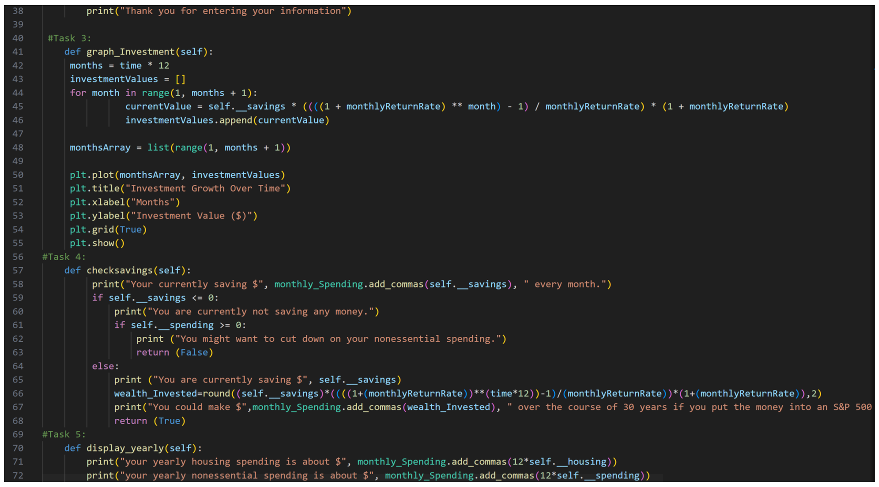
Chapter 3

Project Demonstration

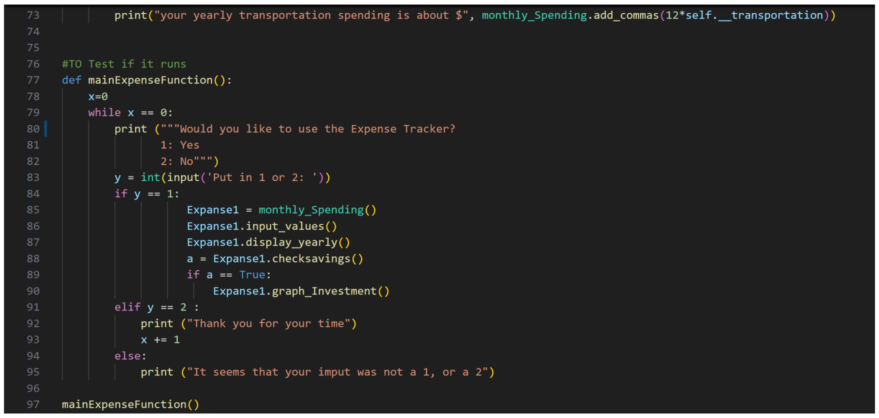
3.1 Screenshots and Code Snippets



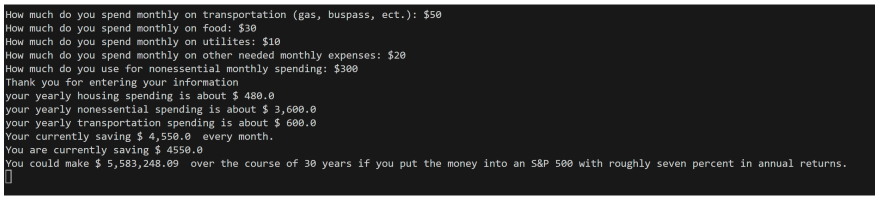
```python
# In order for the graph function to work, matplotlib needs to be downloaded to your machine
# To download, type in the terminal: pip install matplotlib

import matplotlib.pyplot as plt

monthlyReturnRate = 0.07/12
time = 30

#Task 1:
class monthly_Spending:

    @staticmethod
    def add_commas(number):
        return "{:,}".format(number)

    def __init__ (self):
        self.__housing= 0.0
        self.__charges= 0.0
        self.__utilities = 0.0
        self.__transportation = 0.0
        self.__food= 0.0
        self.__neededTotal= 0.0
        self.__salary= 0.0
        self.__spending= 0.0
        self.__savings= 0.0
    #Task 2:
    def input_values(self):
        print("For the expense tracker, it will need information about your monthly finances")
        self.__salary = float(input("What is your monthly salary after tax: $"))
        self.__housing =float(input("How much do you spend monthly on housing: $"))
        self.__transportation = float(input("How much do you spend monthly on transportation (gas, buspass, ect.): $"))
        self.__food = float(input("How much do you spend monthly on food: $"))
        self.__utilities = float(input("How much do you spend monthly on utilites: $"))
        self.__charges = float(input("How much do you spend monthly on other needed monthly expenses: $"))
        self.__spending = float(input("How much do you use for nonessential monthly spending: $"))
        self.__neededTotal= self.__housing+self.__charges+self.__utilities+self.__transportation+self.__food
        self.__savings= self.__salary - self.__neededTotal-self.__spending
```



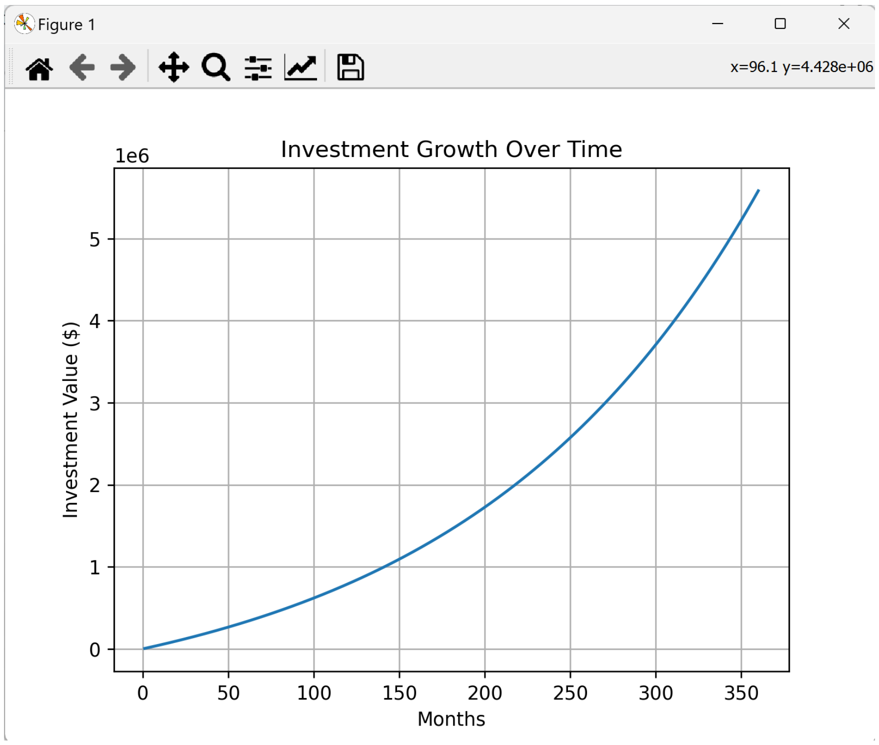
```python
        print("Thank you for entering your information")

 #Task 3:
    def graph_Investment(self):
     months = time * 12
     investmentValues = []
     for month in range(1, months + 1):
              currentValue = self.__savings * ((((1 + monthlyReturnRate) ** month) - 1) / monthlyReturnRate) * (1 + monthlyReturnRate)
              investmentValues.append(currentValue)

     monthsArray = list(range(1, months + 1))

     plt.plot(monthsArray, investmentValues)
     plt.title("Investment Growth Over Time")
     plt.xlabel("Months")
     plt.ylabel("Investment Value ($)")
     plt.grid(True)
     plt.show()
#Task 4:
    def checksavings(self):
        print("Your currently saving $", monthly_Spending.add_commas(self.__savings), " every month.")
        if self.__savings <= 0:
            print("You are currently not saving any money.")
            if self.__spending >= 0:
                print ("You might want to cut down on your nonessential spending.")
                return (False)
        else:
            print ("You are currently saving $", self.__savings)
            wealth_Invested=round((self.__savings)*((((1+(monthlyReturnRate))**(time*12))-1)/(monthlyReturnRate))*(1+(monthlyReturnRate)),2)
            print("You could make $",monthly_Spending.add_commas(wealth_Invested), " over the course of 30 years if you put the money into an S&P 500
            return (True)
#Task 5:
    def display_yearly(self):
        print("your yearly housing spending is about $", monthly_Spending.add_commas(12*self.__housing))
        print("your yearly nonessential spending is about $", monthly_Spending.add_commas(12*self.__spending))
```

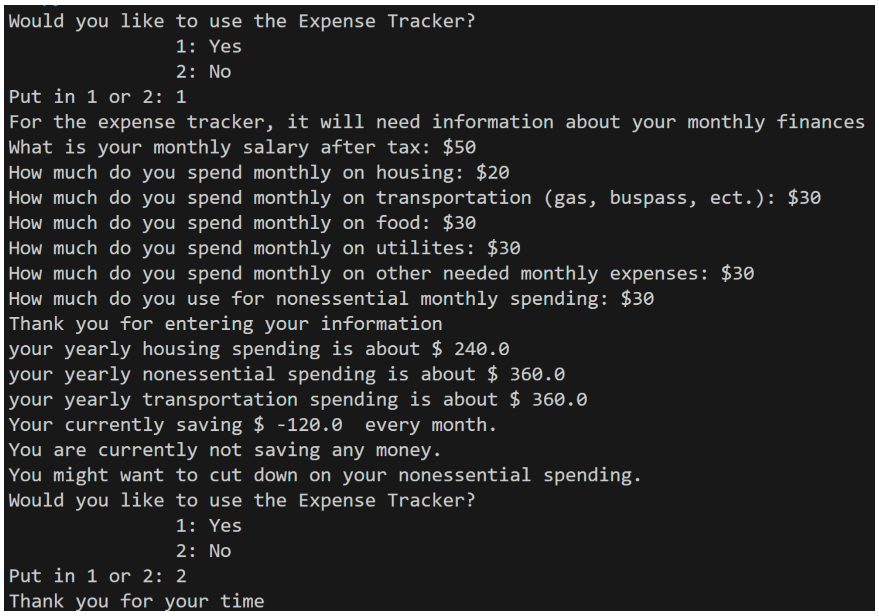


```python
        print("your yearly transportation spending is about $", monthly_Spending.add_commas(12*self.__transportation))


#TO Test if it runs
def mainExpenseFunction():
    x=0
    while x == 0:
        print ("""Would you like to use the Expense Tracker?
               1: Yes
               2: No""")
        y = int(input('Put in 1 or 2: '))
        if y == 1:
                Expanse1 = monthly_Spending()
                Expanse1.input_values()
                Expanse1.display_yearly()
                a = Expanse1.checksavings()
                if a == True:
                    Expanse1.graph_Investment()
        elif y == 2 :
            print ("Thank you for your time")
            x += 1
        else:
            print ("It seems that your imput was not a 1, or a 2")

mainExpenseFunction()
```



```
How much do you spend monthly on transportation (gas, buspass, ect.): $50
How much do you spend monthly on food: $30
How much do you spend monthly on utilites: $10
How much do you spend monthly on other needed monthly expenses: $20
How much do you use for nonessential monthly spending: $300
Thank you for entering your information
your yearly housing spending is about $ 480.0
your yearly nonessential spending is about $ 3,600.0
your yearly transportation spending is about $ 600.0
Your currently saving $ 4,550.0  every month.
You are currently saving $ 4550.0
You could make $ 5,583,248.09  over the course of 30 years if you put the money into an S&P 500 with roughly seven percent in annual returns.
```





```
Would you like to use the Expense Tracker?
               1: Yes
               2: No
Put in 1 or 2: 1
For the expense tracker, it will need information about your monthly finances
What is your monthly salary after tax: $50
How much do you spend monthly on housing: $20
How much do you spend monthly on transportation (gas, buspass, ect.): $30
How much do you spend monthly on food: $30
How much do you spend monthly on utilites: $30
How much do you spend monthly on other needed monthly expenses: $30
How much do you use for nonessential monthly spending: $30
Thank you for entering your information
your yearly housing spending is about $ 240.0
your yearly nonessential spending is about $ 360.0
your yearly transportation spending is about $ 360.0
Your currently saving $ -120.0  every month.
You are currently not saving any money.
You might want to cut down on your nonessential spending.
Would you like to use the Expense Tracker?
               1: Yes
               2: No
Put in 1 or 2: 2
Thank you for your time
```

Chapter 4

Discussion and Future Work

At the conclusion of the project, our team was able to see how this program can be much more unique compared to other apps such as Mint or YNAB because of the features we have implemented. Having a feature that can show users how much they could have made if they invested their excess income into the S&P 500 serves as an additional reason for our users to invest. This program can serve as the foundation for a future app or website that revolves around helping the common person become wealthy after retirement. A future idea that our team has discussed was the possibility of creating a user interface for users to actually use the program. Not to mention, having more features that can look at the person's car payment and let them know if they have a car they cannot afford and whether they should downgrade so that they can save more money for investing. As alluded to before, two of the biggest limiting factors for our team is our lack of experience as well as time and money. Because this is an introductory coding class and our team is relatively new to coding Python, we are not able to execute an app similar to Ming and it limits our ability to create a basic user interface since those skills were not explicitly taught in lecture as well. In addition, the only option for investing that has been implemented in our code so far is the S&P 500 with a rough estimate of the annual returns based on the past thirty years. If there is a way that we can implement live data of stocks changing in real-time that can be updated in our code, then our users can see what stocks they can invest their excess money in. Not to mention, a future feature of this program would be to assess what our user should do with excess money. For example, a user with a ton of excess money could be recommended to buy real estate near their area with some information on how much they can put down and how much they can expect to pay monthly on the mortgage. Essentially, our findings show that we have a good foundation for a potential future app that can compete with the likes of Mint and YNAB.

Chapter 5

Conclusion

Comparing our finished product to other well-known services such as YNAB or Mint, it does not quite compare as those services have much more experienced individuals along with more time and money in order to create a product that people like. Our program serves more like the foundation to something bigger down the line that can challenge these services as some of them cost money and may not adhere to their users' privacy as well. A key feature that our program has that is not necessarily seen in competitors is how compound interest can help someone build wealth. Even though our program can help users track expenses and budget similar to other services, we are also able to encourage and motivate users to save more money. If the user saves money each month, we show them how much they could make if they put that same amount of money each month towards the S&P 500 over the course of thirty years which results in a large sum of money. Not only does users see how much money they could make by investing, but it serves as an additional reason for people to live within or below their means in order to put more money away for investing. Looking at whether the project's objectives were accomplished or not, most of them were completed to some extent and can be improved in the future as we gain more knowledge and improve our coding skills. When it comes to our program being a tool that can help users cut out unnecessary spending, this objective was reasonably accomplished as it alerts the user when they are spending above their means. In terms of being effective in tracking expenses, this objective will be better achieved in the future as we lack the knowledge to effectively show the users a summary of their expenses but are able to lead them in the right direction. By showing users how much they can make if they save a certain amount of money each month, our program is different from other apps since we help them build wealth through compound interest. Because we are able to differentiate ourselves from other applications with this key difference, it shows how our project can be the foundation to an even bigger app or website down the line that can effectively help users see where they can cut out expenses in order to invest more money in funds such as the S&P 500 to live comfortably after retirement.

Bibliography

[1] Features-YNAB(n.d.).Www.ynab.com.`https://www.ynab.com/features`

[2] matplotlib. (n.d.) *Pyplot tutorial — Matplotlib 3.8.0 documentation*. Matplotlib.org. `https://matplotlib.org/stable/tutorials/pyplot.html`

[3] Mint. (2024)*Budget Tracker & Planner-Free Online Money Management-Mint*. Mint.intuit.com.`https://mint.intuit.com/`

[4] PocketGuard. (n.d.). *PocketGuard-Your Personal Money & Bill Organizer*. PocketGuard. `https://pocketguard.com/`