

****

**Midterm Project Report**

**Advanced Computer Programming**

**Finace Tracker web app**

**Student Name : Sinokuhle Dlamini**

**Student ID : 112021171**

**Teacher : DINH-TRUNG VU**

**2024-04**

# Introduction

## Github

1. **Personal Github Account**: 112011171
2. **Group Github Account**: FullyMed
3. **Group Project Repository**: Amigo
4. **List of submitted file**s:
   * Sinokuhle.py

## Topic

Build a web app to track finances

## Project Overview

In my program I have used a dataclass, pprint, operating system library, csv file and pattern matching. My program allows users to enter, edit and remove expenses and print out a list of expenses from a csv file.

# Implementation

## Class Expense

I have imported dataclass library, pprint, the os library, the csv library as well as the datetime library. This class is decorated with decorator “@dataclass” in order to initialize the dataclass library.

### Fields

The class has 3 fields namely；

Date-with the data type being datetime so that the string inputted is converted to the datetime format.

Category-which lets us know what category the expense is, and the type is string.

Amount which is float type.

And this tells us about the object “expense” that we have created. They are features that the expense we want to capture has.

### Methods

I have first created a method to create a csv file using the imported library os. Because this file is not in the operating system, I have used the “if not” statement to indicate the absence of the file and the “os.path.isfile()” to determine whether or not the path is a file then I opened and created the file ‘expense.csv’ in the write mode then wrote the fields of the class in the method.

### Upload expense.

I have defined a function that uploads the expenses into the csv file. I defined expenses as an empty list and the opened the file using again the “if os.path.isfile() “ so i could be able to write into the file using the csv.reader statement. I used next(reader) to skip the headers in the csv file and go straight into the rows. In the rows, I define the fields and place them into the row by indexing them such that date=datetime. strptime([1], format) will be the place of date. The expenses are then returned and saved into “expenses=[]” as per the “.append()” function.

**Save expense.**

This function opens the csv file and writes the expenses input by the user.It refers to the header row first then, it writes the expenses called as instantized by the upload expense function.

**pprint:**

I created a list of the menu and used the pprpint functipn to print it out then I called the csv file created and assigned the expenses values to the upload function.

**While True**

In this loop, the user is prompted to enter their choice and then their choice is validated using the match case statement. Of which :

**case 1:**

simply allows the user to enter the expense by using the input function. Then the expense is defines as a new expense, added to expenss and saved into the csv file by calling the save expense function.

## Case 2:

The list of expenses is enumerated and displayed to the user . Then index is defined as the input of the user. The index selected shoulld be within 0 and the length of the expenses printed out, hence it is defined in between 0 and “len(expenses)”. The expense will be selected, and the user will be able to edit the cateegory and the amount of the expense. Of which they are defined onto variable “new\_category “ and “new\_amount” . Then saved back into the expenses list. If successful program prints out “Expense edited successfully!”

If the index is not with the range, the program returns “ invalid index”

**Case 3**

The user selects the expense using index just as in case 2 and I have used the delete “ del expenses(index)” function to deleted the expense selected.

Then I call on the save function to save the edited expenses to the the csv file.

**Case 4**

I Open the csv file in read mode and create a loop that reads and prints out each row using a comma to seperate them.”print(\*row,sep=’,’)”

**Case \_**

This validates if the user selected any other number other than the available options. If so the program prints out “Thank you for using the app today”.

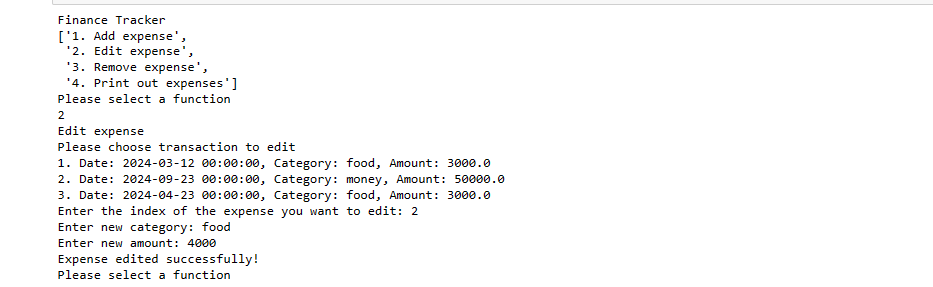
.

# Results

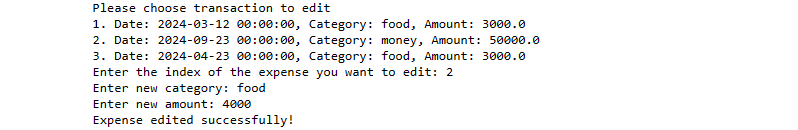
## Result 1



## Result 2



**3.3 Result 3**



**3.4 Result 4**

.

# Conclusions

From this project I have learnt to use the csv file in conjuction to other libary offered by python. By doing this practical work I have learnt to use research to learn how to use a dataclass with refence to methods within the class.