Expense Tracker

PyCK Project Report

Aziz Shameem

20d070020

Contents

1 Introduction

1.1 Importance

There is a popular saying that "the easiest way of becoming bankrupt is by not paying attention to your expenses."People purchase many things over the course of a month. It may just be a packet of gum, a newspaper, an ice cream, or even clothing and household items. At the end of the month, when you evaluate your purchases, you may discover you've made more than 50 purchases just in one month. At that rate if you do not track your expenses, it's easy to go overboard, beyond your income.

Why Track Your Spending

Being aware of our spending habits is the best way of utilizing our money. When you know how much money you spend, it's easy to balance your income with your spending and even save for the future.

For this, an Expense Tracker designed to store your expenses and balance may come in handy, making it easy to store your finances in one place, and much less tedious than maintaining a hand-written table.

1.2 Problem Statement

To create an online tool, complete with a Graphical User Interface, that takes in initial balance and entries in expenses, and stores them. Also, can be used to generate a bar-plot of the expenses.

1.3 Python Libraries Used

- Tkinter : Used for creating the GUI
- Pandas : Used for storing the data obtained in a table
- MatPlotLib : Used for generating bar-plots.
- Hashlib : Used to create hashes for passwords in the table.

2 Implementation

2.1 GUI

The Graphical User Interface(GUI) is implemented using the Python Library 'Tkinter'. It consists of four frames, that are made to appear and disappear by pushing buttons. The elements of the GUI : Labels, Buttons, Entries and the Frame itself are designed using the available tkinter function, and even spaced out on a constant dimension layout. The colours are chosen so that the elements can be distinguished easily, without providing much contrast.

The main problem faced in this phase was of storing the obtained entries in a table, and referring to the values stored later, at will. This was accomplished using another Python library, Pandas, along with some dictionaries in the program itself.
Also, the plot had to be cleverly position/scaled so that it fits in the layout, while also providing enough room for a 'back' button.

2.2 Data Analysis

Data Analysis is a major part of the project. The data obtained has to be stored in a particular format, to be made accessible for plotting later.
Two Python libraries are used to accomplish this, Pandas and MatPlotLib.
Pandas is used to write the data in a previously made table, and to access the required contents of the table later.
MatPlotLib is used to plot the data.(Pandas cannot be used directly to plot the data on a tkinter GUI)

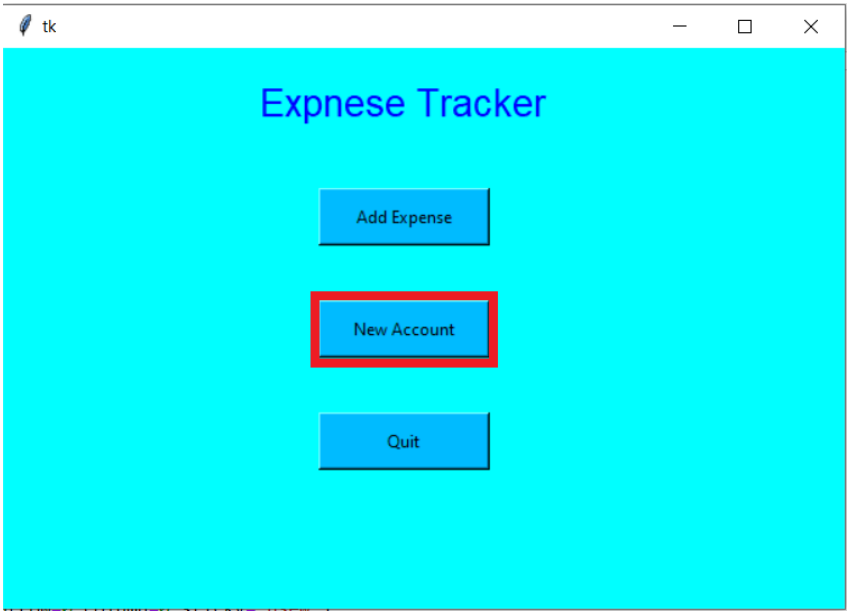
2.3 Hashing

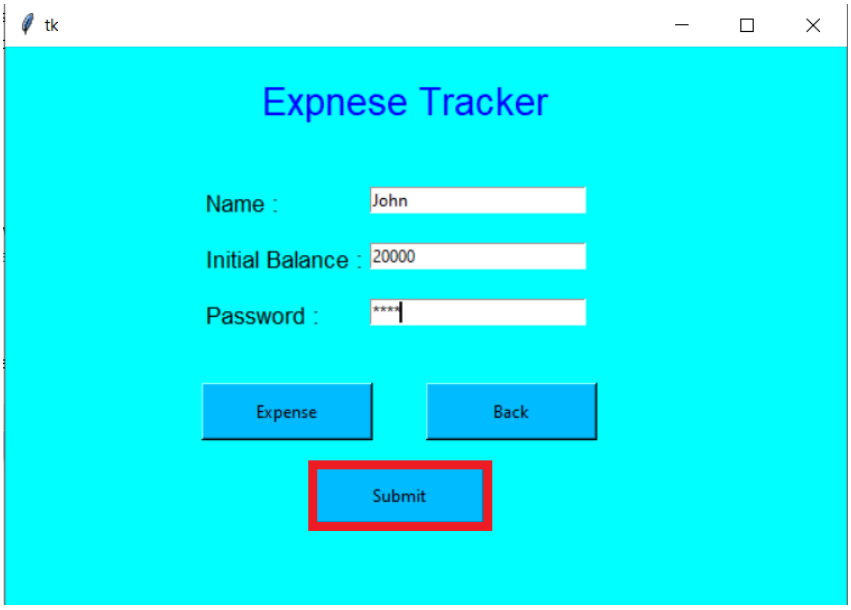
Many firms hash data in order to store them in an encoded manner, making it difficult for hackers and other unauthorized users from accessing private data. It is a many-one transformation, thus it is not easy to get the original password back from its hashed counter-part. Some firms also add *salts*, i.e. additional strings to the password entered before hashing, adding an additional layer of protection.

In this project, hashing of password strings is implemented using the Python library Hashlib. The hashed string is stored in the table. When a password is entered by the user, it is converted and checked for correctness from the table.

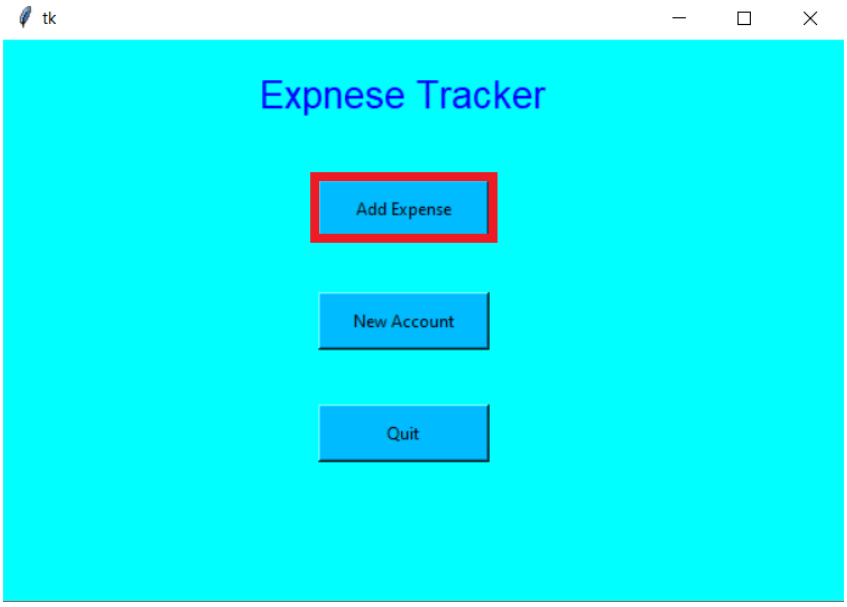
3 Sample Run

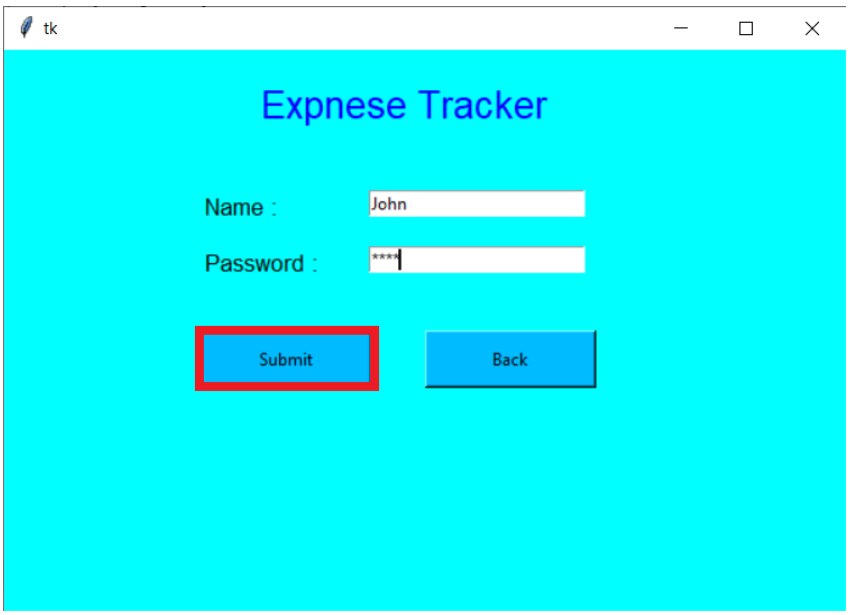
3.1 Adding New Entries

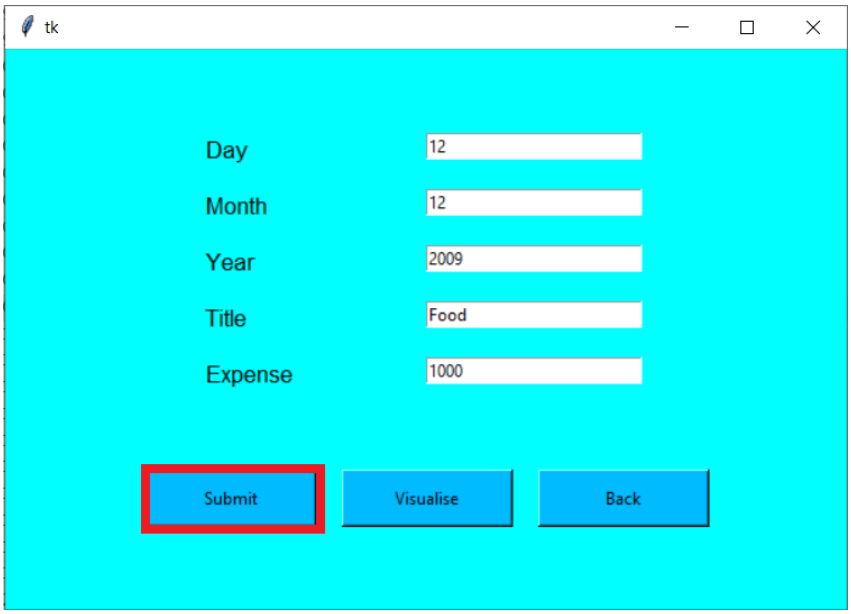




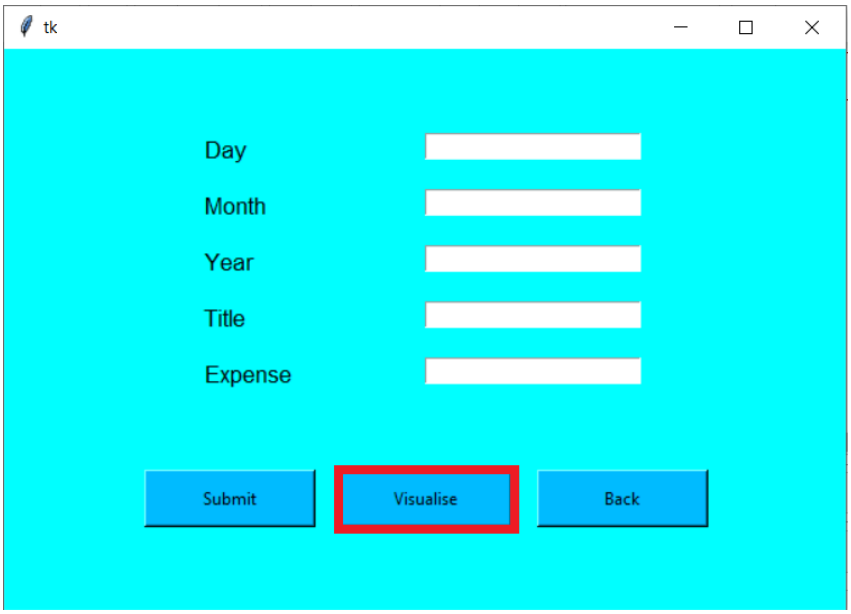
3.2 Adding expenses

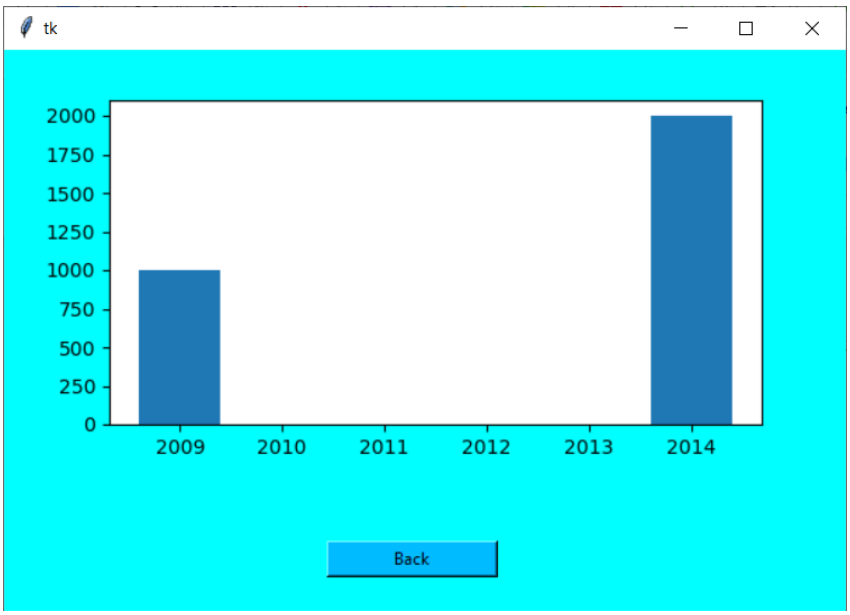






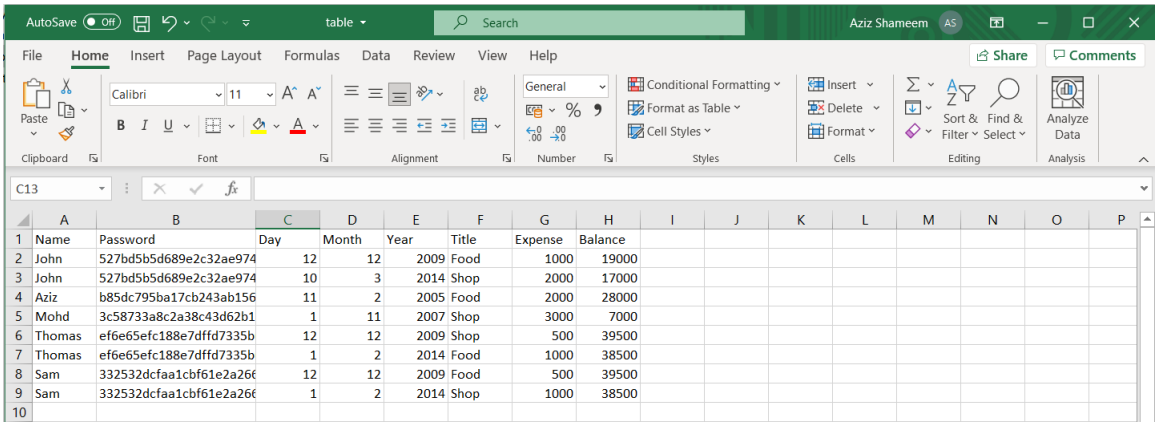
3.3 Generate Plots





3.4 Back-end

This is a sample screen shot of the table, where all the data obtained from the program is stored.



	A	B	C	D	E	F	G	H
1	Name	Password	Day	Month	Year	Title	Expense	Balance
2	John	527bd5b5d689e2c32ae974	12	12	2009	Food	1000	19000
3	John	527bd5b5d689e2c32ae974	10	3	2014	Shop	2000	17000
4	Aziz	b85dc795ba17cb243ab156	11	2	2005	Food	2000	28000
5	Mohd	3c58733a8c2a38c43d62b1	1	11	2007	Shop	3000	7000
6	Thomas	ef6e65efc188e7dffd7335b	12	12	2009	Shop	500	39500
7	Thomas	ef6e65efc188e7dffd7335b	1	2	2014	Food	1000	38500
8	Sam	332532dcfaa1cbf61e2a26	12	12	2009	Food	500	39500
9	Sam	332532dcfaa1cbf61e2a26	1	2	2014	Shop	1000	38500
10								

4 Conclusion

1. The program implemented here can further be extended to include systems that can predict changes in prices, thus can be used to better investments.
2. Can be turned into a mobile application, if required.
3. Limitation : Plot is not so clear for a singular entry.
4. References :
   - tinyurl.com/PyCK-course
   - docs.python.org/
   - javatpoint.com
   - stackoverflow.com
   - projectgurukul.org/
   - geeksforgeeks.org/python-gui-tkinter/
   - geeksforgeeks.org/python-hash-method/
   - searchsqlserver.techtarget.com