NANA KWAKU OKAI-MENSAH

14142023

COHORT A

PYTHON PROJECT REPORT

INTRODUCTION

The youth of today struggle to keep track of the expenses they make and do not prioritize their expenses. According to the University of Illinois at Urbana-Champaign (2018), 32% of the youth struggle with money-management methods such as budgeting. The act of unconscious spending leads to overspending. My project aims to assist the youth of today in budgeting with ease with a user-friendly environment and also gives them room to prioritize their expenses as they wish. The name of the Python program I am developing to solve the problem of unconscious spending among the youth is called 'Budget Assistant.'

DESCRIPTION OF THE VARIOUS PARTS OF THE CODE

First of all, the program welcomes the user. It further asks the user how much money he or she is willing to budget (in GHC). The program provides the user with a list of options to choose from to begin budgeting. To start budgeting, the user has to create a budget list, which is the list of items. If the user picks any other option apart from the first option, "create a budget list," the program will prompt the user to choose the first option. However, the seventh option, which is to quit the program, will be granted once the user confirms his exit. The code has one main function and three inner functions. The picked option is passed into the main function as a parameter. The main function has a while loop, which runs as long as the selected option is an integer from 1 to 7. If the picked option is 1, the individual is asked to enter the item he wants to budget, its cost, the excepted quantity, and most importantly, time the item will be purchased. The time is necessary because the program uses the time an item is to be bought to fish out those items that are irrelevant when the accumulated cost of all the items exceed the user's budget.

All data in this program are stored in lists and dictionaries. The code takes into consideration case sensitivity, hence throughout the code name of the entered item is converted to lowercase to avoid confusion in the code. It also helps the program identify when an item already exists in the budget list. The second option allows the user to remove an item from the budget list at ease. Also, the third option allows the user to view his budget list. The program shows all the items in the budget list and their corresponding quantities, the accumulated cost of the items (cost of item × quantity), and the time they will be purchased.

The inner function, "prioriT," is accessed by picking option 4. The main function hints the user to immediately prioritize his items when his budget expenses exceed his initial budget. The prioriT function, when called runs only if the user is exceeding his budget. It suggests to the user which item should be purchased later from the budget list by selecting the item, which will be bought at a much later date. However, the program gives the user the freedom to pick an item to purchase later. The inner function, "viewItem_time," when called, shows the user the list of items in his budget list with more emphasis on the dates they will be purchased to aid the user make the right decision. The prioriT function comprises a while loop to ensure that the process goes on as long as the user exceeds his initial budget. The chosen item is then added to the dictionary, "laterItems,"(items to be purchased at a later date), which stores the item alongside its cost and can be accessed through option 6. After prioritizing the items in the budget list, the program suggests to the user to consider saving the rest of his money.

Lastly, the third inner function, "Save," can be called by picking option 5. When the Save function is called, the program shows the user the balance of his budget and prompts him to save that balance. Since the main function, "budget," comprises a while loop, the program continues to run until the user picks the seventh option, which ends the program.

IMPORTED MODULES USED IN THE PROGRAM

- **Time** – This module was used to run the program in a calm way to aid the user take time when reading. Usage: time.sleep(1).
- **Datetime** – This module was used to handle all data related to dates in the code. For instance, to find the present day's date. Usage: dt.date.today().
- **Operator** – This module was used to find the item with the highest number of days in the timeSpan dictionary. It was used in the prioriT function. Usage: max(timeSpan.items(), key=operator.itemgetter(1))[0]

Reference

University of Illinois at Urbana-Champaign. (2018, August 24). Many young adults lack financial literacy, economic stability, study finds. *ScienceDaily*. Retrieved February 18, 2020 from www.sciencedaily.com/releases/2018/08/180824135007.htm