**Project – Programming refresher**

import csv

from datetime import datetime

*# Initialize the dictionary for storing expenses*

fin\_dict = {}  # Dictionary to store expenses, keyed by name

*# Function to save expenses to CSV*

def save\_expenses\_to\_csv(fin\_dict, filename='expenses.csv'):

    with open(filename, mode='a', newline='') as file:

        writer = csv.writer(file)

*# Write each person's expenses to the CSV file*

        for Log\_no, expense in fin\_dict.items():

            writer.writerow([Log\_no, expense['date'], expense['category'], expense['amount'], expense['description']])

# Function to load expenses from CSV

def load\_expenses\_from\_csv(filename='expenses.csv'):

    expenses = {}

    try:

        with open(filename, mode='r') as file:

            reader = csv.reader(file)

            next(reader, None)  # Skip the header row

            # Read each row and add it to fin\_dict

            for row in reader:

                Log\_no = row[0]

                expense = {

                    'date': row[1],

                    'category': row[2],

                    'amount': float(row[3]),

                    'description': row[4]

                }

                expenses[Log\_no] = expense

    except FileNotFoundError:

        print(f"File '{filename}' not found. Starting with an empty list.")

    return expenses

*# Function to fill the fin\_dict with expense data*

def filldict(Log\_no, expense\_data):

    fin\_dict[Log\_no] = expense\_data

*# Function to add a new expense and save to CSV*

def main\_func():

    Ans = "Y"

*# Loading existing data from CSV at the start*

    global fin\_dict

    fin\_dict = load\_expenses\_from\_csv()

    while Ans != 'N':

        Ans = input("Do you wish to continue putting details? (Y/N): ")

        if Ans == 'Y':

            Log\_no = input("Entry No: ")

            Date = input("Date of bill (YYYY-MM-DD): ")

            cat = input("Category of expense (food/travel): ")

            amn = input("Amount paid: ")

            desc = input("Give short description (in one line): ")

*# Created a fresh dictionary instance for Emp\_exp for each expense entry*

            Emp\_exp = {'date': Date, 'category': cat, 'amount': float(amn), 'description': desc}

*# Added the new expense to the fin\_dict, keyed by name*

            filldict(Log\_no, Emp\_exp)

*# Saving the updated fin\_dict to the CSV file*

    save\_expenses\_to\_csv(fin\_dict)

*# Function to view all expenses*

def view\_exp():

    print("\n--- All Expenses ---")

    for Log\_no, expense in fin\_dict.items():

        print(f"{Log\_no}: Date: {expense['date']}, Category: {expense['category']}, Amount: {expense['amount']}, Description: {expense['description']}")

    check\_missing\_data()

*# Function to check for missing details*

def missing\_detail(P):

    for key in P:

        if P[key] == '' or P[key] is None:

            resp = input(f"Is '{key}' the key for which data is missing? (Y/N): ")

            if resp == 'Y':

                if key == 'amount':

                    new\_value = float(input(f"Please enter the value for '{key}"))

                    P[key] = new\_value

                    continue

                new\_value = input(f"Please enter the value for '{key}': ")

                P[key] = new\_value

                print(f"Updated '{key}' with value: {new\_value}")

                break

def change\_detail(P):

    for key in P:

        resp = input(f"Is '{key}' the key for which you want to change the data for? (Y/N): ")

        if resp == 'Y':

            if key == 'amount':

                new\_value = float(input(f"Please enter the value for '{key}"))

                P[key] = new\_value

                continue

            new\_value = input(f"Please enter the value for '{key}': ")

            P[key] = new\_value

            print(f"Updated '{key}' with value: {new\_value}")

            break  # After updating, exit the loop and return

*# Function to handle missing values in the dictionary*

def check\_missing\_data():

    Ans = input("Is any value in the data missing? (Y/N): ")

    while Ans != "N":

        for Log\_no, data in fin\_dict.items():

            print(f"Checking details for entry number {Log\_no}")

            missing\_detail(data)

        Ans = input("Is any value in the data missing? (Y/N): ")

    Ans1 = input("Is there any value that you wish to change? (Y/N): ")

    while Ans1 != "N":

        for Log\_no, data in fin\_dict.items():

            afp = input(f"Do you wish to change data for {Log\_no}? (Y/N): ")

            if afp == "Y":

                change\_detail(data)

        Ans1 = input("Is there any data value yet to change?(Y/N): ")

    print("Final data:", fin\_dict)

*# Function to set a monthly budget*

def monthlybudget():

    budget = float(input("Enter a monthly budget you want to set for yourself: "))

    return budget

*# Function to track the budget*

def tallybudget(budget, fin\_dict):

    tot\_exp = 0

*# Calculate total expenses*

    for key in fin\_dict:

        sum\_exp = fin\_dict[key]['amount']

        tot\_exp += sum\_exp

    print(f"Your total expense is coming out to be: {tot\_exp}")

*# Compare expenses with the budget*

    if tot\_exp > budget:

        print(f"You have exceeded your monthly budget by {tot\_exp - budget}")

    elif tot\_exp < budget:

        print(f"You still have {budget - tot\_exp} left from your monthly budget")

    else:

        print("You have spent the exact amount as you wanted your monthly expense to be")

*# Main function for user interaction options*

def options\_det():

    while True:

        opt\_num = int(input("Enter 1 to add expense, 2 to view expense, 3 to track budget, 4 to save expenses and 5 to exit: "))

        if opt\_num == 1:

            main\_func()

        elif opt\_num == 2:

            view\_exp()

        elif opt\_num == 3:

            budget = monthlybudget()

            tallybudget(budget, fin\_dict)

        elif opt\_num == 4:

            save\_expenses\_to\_csv(fin\_dict)

        elif opt\_num == 5:

            print("You have exited!")

            break

# Run the program

if \_\_name\_\_ == "\_\_main\_\_":

    options\_det()