

import json

import os

import tkinter as tk

from tkinter import messagebox

from datetime import datetime

# Initial data setup

data\_file = "budget\_data.json"

# Load data or initialize an empty data structure

if os.path.exists(data\_file):

with open(data\_file, "r") as f:

data = json.load(f)

else:

data = {

"income": 0.0,

"budget": 0.0,

"expenses": []

}

# Save data function

def save\_data():

with open(data\_file, "w") as f:

json.dump(data, f, indent=4)

# Helper function to format currency

def format\_currency(amount):

return f"${amount:,.2f}"

# Set monthly income

def set\_income():

income = float(income\_entry.get())

data["income"] = income

save\_data()

messagebox.showinfo("Success", f"Monthly income set to {format\_currency(income)}.")

# Set monthly budget

def set\_budget():

budget = float(budget\_entry.get())

data["budget"] = budget

save\_data()

messagebox.showinfo("Success", f"Monthly budget set to {format\_currency(budget)}.")

# Add an expense

def add\_expense():

category = category\_entry.get()

amount = float(amount\_entry.get())

date = datetime.now().strftime("%Y-%m-%d")

expense = {

"category": category,

"amount": amount,

"date": date

}

data["expenses"].append(expense)

save\_data()

messagebox.showinfo("Success", f"Added expense: {format\_currency(amount)} for {category} on {date}")

check\_budget()

# Check if budget is exceeded

def check\_budget():

total\_expenses = sum(expense["amount"] for expense in data["expenses"])

if total\_expenses > data["budget"]:

messagebox.showwarning("Budget Exceeded", f"You have exceeded your budget by {format\_currency(total\_expenses - data['budget'])}!")

# Show summary of expenses

def show\_summary():

total\_expenses = sum(expense["amount"] for expense in data["expenses"])

summary = f"--- Monthly Summary ---\n"

summary += f"Monthly Income: {format\_currency(data['income'])}\n"

summary += f"Monthly Budget: {format\_currency(data['budget'])}\n"

summary += f"Total Expenses: {format\_currency(total\_expenses)}\n"

summary += f"Remaining Budget: {format\_currency(data['budget'] - total\_expenses)}\n\n"

categories = {}

for expense in data["expenses"]:

if expense["category"] in categories:

categories[expense["category"]] += expense["amount"]

else:

categories[expense["category"]] = expense["amount"]

summary += "Expenses by Category:\n"

for category, amount in categories.items():

summary += f"{category}: {format\_currency(amount)}\n"

messagebox.showinfo("Summary", summary)

# Delete all recorded data

def delete\_all\_data():

confirm = messagebox.askyesno("Confirm Deletion", "Are you sure you want to delete all recorded data?")

if confirm:

data["income"] = 0.0

data["budget"] = 0.0

data["expenses"] = []

save\_data()

messagebox.showinfo("Success", "All data has been deleted.")

# Create the main window

root = tk.Tk()

root.title("BudgetWise")

# Layout

frame = tk.Frame(root)

frame.pack(pady=20)

# Income entry

income\_label = tk.Label(frame, text="Monthly Income:")

income\_label.grid(row=0, column=0, padx=10, pady=5)

income\_entry = tk.Entry(frame)

income\_entry.grid(row=0, column=1, padx=10, pady=5)

# Budget entry

budget\_label = tk.Label(frame, text="Monthly Budget:")

budget\_label.grid(row=1, column=0, padx=10, pady=5)

budget\_entry = tk.Entry(frame)

budget\_entry.grid(row=1, column=1, padx=10, pady=5)

# Category entry

category\_label = tk.Label(frame, text="Expense Category:")

category\_label.grid(row=2, column=0, padx=10, pady=5)

category\_entry = tk.Entry(frame)

category\_entry.grid(row=2, column=1, padx=10, pady=5)

# Amount entry

amount\_label = tk.Label(frame, text="Expense Amount:")

amount\_label.grid(row=3, column=0, padx=10, pady=5)

amount\_entry = tk.Entry(frame)

amount\_entry.grid(row=3, column=1, padx=10, pady=5)

# Buttons

button\_frame = tk.Frame(root)

button\_frame.pack(pady=20)

set\_income\_button = tk.Button(button\_frame, text="Set Income", command=set\_income)

set\_income\_button.grid(row=0, column=0, padx=10)

set\_budget\_button = tk.Button(button\_frame, text="Set Budget", command=set\_budget)

set\_budget\_button.grid(row=0, column=1, padx=10)

add\_expense\_button = tk.Button(button\_frame, text="Add Expense", command=add\_expense)

add\_expense\_button.grid(row=0, column=2, padx=10)

show\_summary\_button = tk.Button(button\_frame, text="Show Summary", command=show\_summary)

show\_summary\_button.grid(row=1, column=0, padx=10)

delete\_data\_button = tk.Button(button\_frame, text="Delete All Data", command=delete\_all\_data)

delete\_data\_button.grid(row=1, column=1, padx=10)

exit\_button = tk.Button(button\_frame, text="Exit", command=root.quit)

exit\_button.grid(row=1, column=2, padx=10)

# Start the application

root.mainloop()