**EXPENSES TRACKER**

**PROGRAM:**

import os

import tkinter as tk

from tkinter import ttk

defadd\_expense():

date = date\_entry.get()

category = category\_entry.get()

amount = amount\_entry.get()

if date and category and amount:

with open("expenses.txt", "a") as file:

file.write(f"{date},{category},{amount}\n")

status\_label.config(text="Expense added successfully!", fg="green")

date\_entry.delete(0, tk.END)

category\_entry.delete(0, tk.END)

amount\_entry.delete(0, tk.END)

view\_expenses()

else:

status\_label.config(text="Please fill all the fields!", fg="red")

defdelete\_expense():

selected\_item = expenses\_tree.selection()

if selected\_item:

item\_text = expenses\_tree.item(selected\_item, "values")

date, category, amount = item\_text

with open("expenses.txt", "r") as file:

lines = file.readlines()

with open("expenses.txt", "w") as file:

for line in lines:

if line.strip() != f"{date},{category},{amount}":

file.write(line)

status\_label.config(text="Expense deleted successfully!", fg="green")

view\_expenses()

else:

status\_label.config(text="Please select an expense to delete!", fg="red")

defview\_expenses():

global expenses\_tree

if os.path.exists("expenses.txt"):

total\_expense = 0

expenses\_tree.delete(\*expenses\_tree.get\_children())

with open("expenses.txt", "r") as file:

for line in file:

date, category, amount = line.strip().split(",")

expenses\_tree.insert("", tk.END, values=(date, category, amount))

total\_expense += float(amount)

total\_label.config(text=f"Total Expense: {total\_expense:.2f}")

else:

total\_label.config(text="No expenses recorded.")

expenses\_tree.delete(\*expenses\_tree.get\_children())

# Create the main application window

root = tk.Tk()

root.title("Expense Tracker")

# Create labels and entries for adding expenses

date\_label = tk.Label(root, text="Date (YYYY-MM-DD):")

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

date\_entry = tk.Entry(root)

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

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

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

category\_entry = tk.Entry(root)

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

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

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

amount\_entry = tk.Entry(root)

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

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

add\_button.grid(row=3, column=0, columnspan=2, padx=5, pady=10)

# Create a treeview to display expenses

columns = ("Date", "Category", "Amount")

expenses\_tree = ttk.Treeview(root, columns=columns, show="headings")

expenses\_tree.heading("Date", text="Date")

expenses\_tree.heading("Category", text="Category")

expenses\_tree.heading("Amount", text="Amount")

expenses\_tree.grid(row=4, column=0, columnspan=3, padx=5, pady=5)

# Create a label to display the total expense

total\_label = tk.Label(root, text="")

total\_label.grid(row=5, column=0, columnspan=2, padx=5, pady=5)

# Create a label to show the status of expense addition and deletion

status\_label = tk.Label(root, text="", fg="green")

status\_label.grid(row=6, column=0, columnspan=2, padx=5, pady=5)

# Create buttons to view and delete expenses

view\_button = tk.Button(root, text="View Expenses", command=view\_expenses)

view\_button.grid(row=7, column=0, padx=5, pady=10)

delete\_button = tk.Button(root, text="Delete Expense", command=delete\_expense)

delete\_button.grid(row=7, column=1, padx=5, pady=10)

# Check if the 'expenses.txt' file exists; create it if it doesn't

if not os.path.exists("expenses.txt"):

with open("expenses.txt", "w"):

pass

# Display existing expenses on application start

view\_expenses()

root.mainloop()

**OUTPUT:**

