Personal Finance Tracker

Overview:

Create a personal finance tracker that helps users track their income and expenses. The application will allow users to add transactions, categorize them, and generate reports showing their spending habits over time.

Key Skills:

- GUI development using Tkinter
- Data storage using SQLite
- Data visualization using Matplotlib

Code Snippet:

```
import tkinter as tk

from tkinter import ttk

import sqlite3

import matplotlib.pyplot as plt


# Database setup

conn = sqlite3.connect('finance.db')

c = conn.cursor()

c.execute('''CREATE TABLE IF NOT EXISTS transactions

             (id INTEGER PRIMARY KEY, type TEXT, category TEXT, amount REAL, date TEXT)''')

conn.commit()


# Function to add transaction
```

```
def add_transaction():

    t_type = type_var.get()

    category = category_var.get()

    amount = float(amount_var.get())

    date = date_var.get()

    c.execute("INSERT INTO transactions (type, category, amount, date) VALUES (?, ?, ?, ?)",

              (t_type, category, amount, date))

    conn.commit()

    load_transactions()



# Function to load transactions

def load_transactions():

    for row in tree.get_children():

        tree.delete(row)

    for row in c.execute("SELECT * FROM transactions"):

        tree.insert("", tk.END, values=row)



# Function to visualize data

def visualize_data():

    c.execute("SELECT category, SUM(amount) FROM transactions GROUP BY category")

    data = c.fetchall()

    categories = [row[0] for row in data]

    amounts = [row[1] for row in data]



    plt.figure(figsize=(10, 5))

    plt.bar(categories, amounts)
    plt.xlabel('Category')

    plt.ylabel('Amount')

    plt.title('Spending by Category')

    plt.show()



# GUI setup

root = tk.Tk()

root.title("Personal Finance Tracker")



type_var = tk.StringVar()

category_var = tk.StringVar()

amount_var = tk.StringVar()

date_var = tk.StringVar()



tk.Label(root, text="Type").grid(row=0, column=0)

tk.Entry(root, textvariable=type_var).grid(row=0, column=1)

tk.Label(root, text="Category").grid(row=1, column=0)

tk.Entry(root, textvariable=category_var).grid(row=1, column=1)

tk.Label(root, text="Amount").grid(row=2, column=0)

tk.Entry(root, textvariable=amount_var).grid(row=2, column=1)

tk.Label(root, text="Date").grid(row=3, column=0)

tk.Entry(root, textvariable=date_var).grid(row=3, column=1)

tk.Button(root, text="Add Transaction", command=add_transaction).grid(row=4, column=0, columnspan=2)



tree = ttk.Treeview(root, columns=("ID", "Type", "Category", "Amount", "Date"), show='headings')

tree.heading("ID", text="ID")
tree.heading("Type", text="Type")
tree.heading("Category", text="Category")
tree.heading("Amount", text="Amount")
tree.heading("Date", text="Date")
tree.grid(row=5, column=0, columnspan=2)

tk.Button(root, text="Visualize Data", command=visualize_data).grid(row=6, column=0, columnspan=2)

load_transactions()
root.mainloop()
```