

```

import tkinter as tk
from tkinter import filedialog
from tkinter import ttk

def categorize_transaction(description):
    if 'FOOD' in description.upper():
        return 'food'
    elif 'STUDY' in description.upper():
        return 'study'
    elif 'CLOTH' in description.upper():
        return 'cloth'
    elif 'SELF CARE' in description.upper():
        return 'Selfcare'
    elif 'STOCKMARKET' in description.upper():
        return 'Stockmarket'
    elif 'INCOME' in description.upper():
        return 'Income'
    else:
        return 'Other'

df = None

budget = 0.0

def categorize_bank_statement(text_output):
    global df
    file_path = filedialog.askopenfilename(filetypes=[("CSV Files", "*.csv")])
    if file_path:
        df = pd.read_csv(file_path)

        df['Category'] = df['Description'].apply(categorize_transaction)

        category_totals = df.groupby('Category')['Amount'].sum().reset_index()

        text_output.delete(1.0, tk.END)
        text_output.insert(tk.END, "Categorized Bank Statement:\n")
        text_output.insert(tk.END, df.to_string(index=False))

        plt.figure(figsize=(5, 3))
        plt.bar(category_totals['Category'], category_totals['Amount'])
        plt.xlabel('Category')
        plt.ylabel('Total Amount')
        plt.title('Category-wise Expense Totals')
        plt.xticks(rotation=45)
        plt.tight_layout()

        canvas = FigureCanvasTkAgg(plt.gcf(), master=window)
        canvas_widget = canvas.get_tk_widget()
        canvas_widget.grid(row=3, column=0, padx=10, pady=(10, 10))

def set_budget():
    global budget
    budget_value = budget_entry.get()
    try:
        budget = float(budget_value)
        text_output.insert(tk.END, f"\n\nBudget set to: ${budget}")
    except ValueError:
        text_output.insert(tk.END, "\n\nInvalid budget value")

def create_monthly_pie_charts():
    if df is not None:
        df['Date'] = pd.to_datetime(df['Date'])
        df['Year'] = df['Date'].dt.year
        df['Month'] = df['Date'].dt.month_name()

        df_expenses = df[df['Description'] != 'Income']

        monthly_expenses = df_expenses.groupby(['Year', 'Month', 'Category'])['Amount'].sum().reset_index()

        total_expenses = monthly_expenses.groupby(['Year', 'Month'])['Amount'].sum().reset_index()

        unique_months = df_expenses['Month'].unique()
        for month in unique_months:
            month_data = monthly_expenses[monthly_expenses['Month'] == month]
            labels = month_data['Category']
            sizes = month_data['Amount']

            plt.figure(figsize=(5, 2))
            plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)
            plt.title(f"{month} Category-wise Expenses")
            plt.axis('equal')
            plt.tight_layout()

            canvas = FigureCanvasTkAgg(plt.gcf(), master=canvas_frame)
            canvas_widget = canvas.get_tk_widget()
            canvas_widget.grid(row=unique_months.tolist().index(month), column=1, padx=10, pady=(10, 10))

        total_budget = budget
        total_actual_expenses = total_expenses['Amount'].sum()
        if total_actual_expenses > total_budget:

```

```

        text_output.insert(tk.END, f"\n\nBudget Exceeded! Total Expenses: ${total_actual_expenses:.2f}")
    else:
        text_output.insert(tk.END, f"\n\nBudget Not Exceeded. Total Expenses: ${total_actual_expenses:.2f}")

def buttonfn():

    categorize_bank_statement(text_output)
    create_monthly_pie_charts()

window = tk.Tk()
window.title("Bank Statement Categorizer")
window.config(bg='white')

big_text_style = ttk.Style()
big_text_style.configure("BigText.TLabel", font=("Helvetica", 30, "bold"), foreground="#000000")

big_text_label = ttk.Label(window, text="Expense Tracker", style="BigText.TLabel", background='#ff9900')
big_text_label.grid(row=0, column=0, columnspan=4, pady=(5, 5))

open_button = tk.Button(window, text="Open Bank Statement", command=lambda: buttonfn(), font=("Helvetica", 10, "bold"), bg='#146EB4', fg='White')
text_output = tk.Text(window, height=15, width=60, font=("Helvetica", 10, "bold"), bg='#f2f2f2', fg='Black')

budget_label = tk.Label(window, text="Set Budget:", font=("Helvetica", 10, "bold"), bg='White')
budget_entry = tk.Entry(window, font=("Helvetica", 10, "bold"), bg='#f2f2f2')
set_budget_button = tk.Button(window, text="Set Budget", command=set_budget, font=("Helvetica", 10, "bold"), bg='#146EB4', fg='White')

canvas_frame = tk.Frame(window, bg='#f2f2f2')
canvas_frame.grid(row=2, column=1, rowspan=5, columnspan=3, padx=10, pady=10)

open_button.grid(row=1, column=0, pady=10)
text_output.grid(row=2, column=0, padx=100, pady=10)

budget_label.grid(row=1, column=1, padx=0, pady=10)
budget_entry.grid(row=1, column=2, padx=0, pady=10)
set_budget_button.grid(row=1, column=3, padx=0, pady=10)
window.mainloop()

```