Documentation of GUI

Description of Expenses Tracker Code

The expenses tracker code is a Tkinter-based Python application that enables users to manage their expenses efficiently. It features a user-friendly interface where users can input details such as description, amount, and category for each expense. The code calculates and displays the total expenses, organizes expenses by category, and allows users to clear all entered expenses if needed. Error handling ensures data integrity by validating input fields. Overall, the expenses tracker code provides a convenient and effective solution for users to track their spending and make informed financial decisions.

Benefits of Expenses Tracker Code

The expenses tracker code is a Tkinter-based Python application that enables users to input, track, and manage their expenses conveniently through a user-friendly interface. It offers error handling for input validation, customizable expense categories, automatic total calculation, and clear visualization of expenses. This application promotes financial discipline, assists in analyzing spending patterns, and facilitates informed financial decision-making.

Objective of Expeses Tracker Code

The objective of the expenses tracker code is to create a user-friendly application that helps individuals manage their expenses effectively. It aims to provide a simple yet comprehensive tool for users to input, track, and analyze their spending habits. By organizing expenses, calculating totals, and offering clear visualization, the application empowers users to make informed financial decisions and maintain better control over their finances.
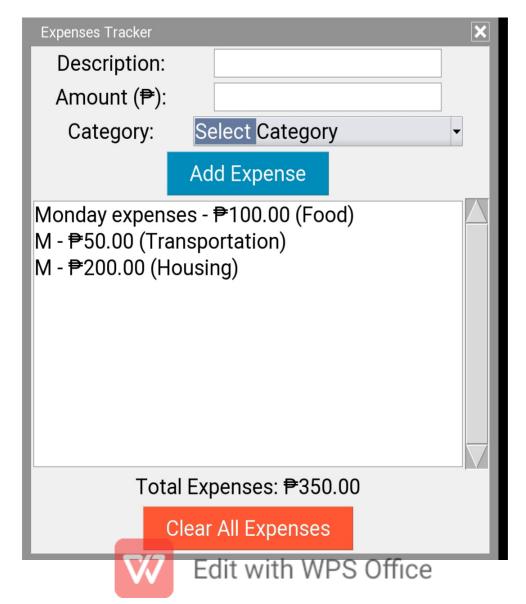
Expenses Tracker Code

```python
import tkinter as tk
from tkinter import messagebox
from tkinter import ttk

class ExpensesTracker:
    def __init__(self, master):
        self.master = master
        self.master.title("Expenses Tracker")
        self.master.configure(bg="#f0f0f0")

        self.expenses = []

        self.description_label = tk.Label(master, text="Description:", bg="#f0f0f0")
        self.description_label.grid(row=0, column=0, padx=5, pady=5)
        self.description_entry = tk.Entry(master)
        self.description_entry.grid(row=0, column=1, padx=5, pady=5)

        self.amount_label = tk.Label(master, text="Amount (₱):", bg="#f0f0f0")
        self.amount_label.grid(row=1, column=0, padx=5, pady=5)
        self.amount_entry = tk.Entry(master)
        self.amount_entry.grid(row=1, column=1, padx=5, pady=5)

        self.category_label = tk.Label(master, text="Category:", bg="#f0f0f0")
        self.category_label.grid(row=2, column=0, padx=5, pady=5)
        self.category_var = tk.StringVar(master)
        self.category_var.set("Select Category")
        self.categories = ["Food", "Transportation", "Housing", "Entertainment", "Utilities", "Other"]
        self.category_dropdown = ttk.Combobox(master, textvariable=self.category_var, values=self.categories, state="readonly")
        self.category_dropdown.grid(row=2, column=1, padx=5, pady=5, sticky="ew")
        self.category_dropdown.configure(background="#ffffff", font=("Helvetica", 8), width=15)

        self.add_button = tk.Button(master, text="Add Expense", command=self.add_expense, bg="#008CBA", fg="white")
        self.add_button.grid(row=3, columnspan=2, padx=5, pady=5)

        self.expense_listbox = tk.Listbox(master, width=38)
        self.expense_listbox.grid(row=4, columnspan=2, padx=5, pady=5, sticky="nsew")

        self.scrollbar = tk.Scrollbar(master, orient="vertical", command=self.expense_listbox.yview)
        self.scrollbar.grid(row=4, column=2, sticky="ns")
        self.expense_listbox.config(yscrollcommand=self.scrollbar.set)

        self.total_label = tk.Label(master, text="Total Expenses: ₱0.00", bg="#f0f0f0")
        self.total_label.grid(row=5, columnspan=2, padx=5, pady=5)

        self.clear_button = tk.Button(master, text="Clear All Expenses", command=self.clear_expenses, bg="#FF5733", fg="white")
        self.clear_button.grid(row=6, columnspan=2, padx=5, pady=5)

        self.master.bind("<Return>", self.add_expense)

    def add_expense(self, event=None):
        description = self.description_entry.get()
        amount = self.amount_entry.get()
        category = self.category_var.get()
        if description and amount and category != "Select Category":
            try:
                amount = float(amount)
                self.expenses.append({"description": description, "amount": amount, "category": category})
                self.update_expense_list()
                self.update_total_label()
                self.clear_fields()
            except ValueError:
                messagebox.showerror("Error", "Amount must be a number")
        else:
            messagebox.showerror("Error", "Please fill in all fields")

    def update_expense_list(self):
        self.expense_listbox.delete(0, tk.END)
        for expense in self.expenses:
            self.expense_listbox.insert(tk.END, f"{expense['description']} - ₱{expense['amount']:.2f} ({expense['category']})")
```

```python
72     def update_total_label(self):
73        total = sum(expense['amount'] for expense
    in self.expenses)
74        self.total_label.config(text=f"Total
    Expenses: ₱{total:.2f}")
75
76     def clear_expenses(self):
77        if messagebox.askyesno("Confirmation",
    "Clear all expenses?"):
78           self.expenses = []
79           self.update_expense_list()
80           self.update_total_label()
81
82     def clear_fields(self):
83        self.description_entry.delete(0, tk.END)
84        self.amount_entry.delete(0, tk.END)
85        self.category_var.set("Select Category")
86
87  if __name__ == "__main__":
88     root = tk.Tk()
89     app = ExpensesTracker(root)
90     root.mainloop()
```

OUTPUT

Expenses Tracker   ✕

Description: [   ]

Amount (₱): [   ]

Category: [Select Category   ▼]

[Add Expense]

Monday expenses - ₱100.00 (Food)
M - ₱50.00 (Transportation)
M - ₱200.00 (Housing)

Total Expenses: ₱350.00

[Clear All Expenses]

PERSONAL INFORMATION

Name :   Kim Palotes Tarro

Contact Number : 09063444417

Email Adress :   ktarro@ssct.edu.ph

Date of Birth :   April 23 2000

Place of Birth :   Novaliches Quezon City

Adress :   Timamana Tubod Surigao Del Norte

Age :.   24

Nationality :   Filipino

Religion :   Roman Catholic

Civil Status :   Single

Mother's Name : Lenny Magbanua Palotes

Father's Name : Advento Vertudazo Tarro Jr.

Education Background

Elementary :   Timamana Elementary School

Adress :   Timamana, Tubod Surigao Del Norte

Junior High School : Timamana National High School

Adress :   Timamana, Tubod Surigao Del Norte

Senior High School : Colegio de Santo Ñino de Jasaan

Adress :   Jasaan, Misamis Oriental

Purpose of Expenses Tracker Code

The purpose of the expenses tracker code is to provide individuals with a tool to effectively manage their finances by tracking and organizing their expenses. By recording details such as description, amount, and category for each expense, the code helps users understand their spending habits, identify areas for potential savings, and adhere to budgetary goals. Additionally, the expenses tracker code promotes financial awareness, accountability, and discipline, empowering users to make informed financial decisions and achieve long-term financial stability.

Significance of Expenses Tracker code

- **Financial Awareness**: It promotes awareness of spending habits by providing a platform to track expenses systematically, helping users understand where their money is going.

- **Budgeting**: By tracking expenses, users can establish and stick to a budget more effectively, ensuring they allocate funds appropriately to different categories.

- **Financial Discipline**: The code encourages financial discipline by allowing users to monitor their spending in real-time, potentially curbing unnecessary purchases and promoting responsible financial behavior.

- **Goal Setting**: Users can set financial goals more effectively by understanding their spending patterns and identifying areas where they can save or allocate funds towards specific objectives.

- **Data-Driven Decisions**: With clear visualization and total calculations, users can make data-driven decisions regarding their finances, identifying trends, and areas for improvement.

- **Long-Term Financial Health**: By providing insights into spending habits and facilitating budgeting, the expenses tracker code contributes to the long-term financial health and stability of individuals.