Project: Basic Calculator with GUI (Tkinter)

"07/03/2025"

Team Members: Group 07

- Xuan Anh Mai
- Ashanshi Nipuni Uggoda
- Sean sav
- Mai Vy Nguyen
- Nandana Sri Dharmarathne
- Chathurika Madushani
- Phu Si Oh

Trello Board Group 7's Project link : https://trello.com/b/H0zDp8Vh/group-7s-project

GitHub calculator project link : https://github.com/nsdharmarathne/Calculator_Project

Project Description

This project is a GUI-based **Basic Calculator** built using Python and the **Tkinter** library. The calculator allows users to perform basic arithmetic operations:

- Addition
- Subtraction
- Multiplication
- Division
- Square (x^2)
- Square Root ($\sqrt{x}$)
- Clear Entry
- Backspace
- Exit

The calculator provides a user-friendly interface with buttons for each operation.

Description of the Project:

Import Required Libraries

In [ ]:
We first import the necessary Python modules.

```
: python
import tkinter as tk
from tkinter import messagebox
import math
```

- **tkinter**: Used to create the GUI window and buttons.
- **message box**: Used to display error messages.
- **math**: Used for square root calculations.

Creating the Main Window

In [ ]:
```
# Create main window
window = tk.Tk()
window.title("Calculator - Group 07")
window.geometry("330x550")
window.resizable(False, False)  # Prevent resizing
```

We initialize the main application window.

- **"tk.Tk()."** Creates the main window.
- **"title()."** Sets the title of the window.
- **"geometry()."** Sets the window size.
- **"resizable(False, False)."** Prevents resizing.

Creating the Input Field:

A text entry field for user input.

In [ ]:
```
# Define font style
font_style = ("Poppins", 15)

# Create Entry widget
entry = tk.Entry(window, font=font_style, borderwidth=3, relief="sunken", width=22, justify="right")
entry.grid(row=0, column=0, columnspan=4, ipadx=8, ipady=8, pady=10)
```

Defining Calculator Functions:

In [ ]:
```
# Evaluate expression
def equal():
    try:
        result = eval(entry.get())
        entry.delete(0, tk.END)
        entry.insert(0, result)
    except ZeroDivisionError:
        messagebox.showerror("Error", "Cannot divide by zero!")
    except Exception:
        messagebox.showerror("Error", "Invalid Expression")
```

In [ ]:
```
# Square function (x²)
def square():
    try:
        num = float(entry.get())
        entry.delete(0, tk.END)
        entry.insert(0, num ** 2)
    except ValueError:
        messagebox.showerror("Error", "Enter a valid number!")
```

In [ ]:
```
# Square root function (√x)
def square_root():
    try:
        num = float(entry.get())
        if num < 0:
            messagebox.showerror("Error", "Cannot calculate square root of a negative number!")
        else:
            entry.delete(0, tk.END)
            entry.insert(0, math.sqrt(num))
    except ValueError:
        messagebox.showerror("Error", "Enter a valid number!")
```

Creating the Calculator Buttons

In [ ]:
```
buttons = [
    ('7', 1, 0), ('8', 1, 1), ('9', 1, 2), ('/', 1, 3),
    ('4', 2, 0), ('5', 2, 1), ('6', 2, 2), ('*', 2, 3),
    ('1', 3, 0), ('2', 3, 1), ('3', 3, 2), ('-', 3, 3),
    ('0', 4, 0), ('.', 4, 1), ('+', 4, 2), ('=', 4, 3),
]

# Create buttons dynamically
for text, row, col in buttons:
    action = lambda x=text: myclick(x) if x != "=" else equal()
    tk.Button(window, text=text, font=font_style, padx=20, pady=15, command=action).grid(row=row, column=col, padx=5, pady=5)
```

Adding Special Buttons*

We add additional buttons for special operations.

In [ ]:
```
# Special Buttons
tk.Button(window, text="C", font=font_style, padx=20, pady=15, command=clear, bg="red", fg="white").grid(row=5, column=0, padx=5, pady=5)
tk.Button(window, text="←", font=font_style, padx=20, pady=15, command=backspace, bg="gray", fg="white").grid(row=5, column=1, padx=5, pady=5)
tk.Button(window, text="x²", font=font_style, padx=20, pady=15, command=square, bg="lightblue").grid(row=5, column=2, padx=5, pady=5)
tk.Button(window, text="√x", font=font_style, padx=20, pady=15, command=square_root, bg="lightgreen").grid(row=5, column=3, padx=5, pady=5)
tk.Button(window, text="Exit", font=font_style, padx=70, pady=15, command=exit_app, bg="black", fg="white").grid(row=6, column=0, columnspan=4, padx=5, pady=10)
```

- **"Clear (C)."** Clears the input field.
- **"Backspace (←)."** Deletes the last character.
- **"Square (x^2)."** Calculates the square of the input number.
- **"Square Root ($\sqrt{x}$)."** Finds the square root of the input number.
- **"Exit."** Closes the application.

Running the Calculator

In [ ]:
```
# Run the application
window.mainloop()
```

Sample Output (GUI)

Conclusion

This **"Tkinter-based GUI Calculator"** allows users to perform basic arithmetic calculations interactively. The interface is designed for ease of use and provides error handling for invalid inputs.

Future Improvements

- Add trigonometric functions" (sin, cos, tan).
- Implement a **"dark mode theme"**.
- Add **"memory functions"** (M+, M-, MR).

Final Code

In [23]:
```
import tkinter as tk
from tkinter import messagebox
import math

# Create main window
window = tk.Tk()
window.title("Calculator Group - 07")
window.geometry("330x550")
window.resizable(False, False)  # Prevent resizing

# Define font style
font_style = ("Poppins", 15)

# Create Entry widget for input
entry = tk.Entry(window, font=font_style, borderwidth=3, relief="sunken", width=22, justify="right")
entry.grid(row=0, column=0, columnspan=4, ipadx=8, ipady=8, pady=10)


# Function to insert number/symbol into entry field
def myclick(value):
    entry.insert(tk.END, value)


# Function to evaluate the expression
def equal():
    try:
        result = eval(entry.get())
        entry.delete(0, tk.END)
        entry.insert(0, result)
    except ZeroDivisionError:
        messagebox.showerror("Error", "Cannot divide by zero!")
    except Exception:
        messagebox.showerror("Error", "Invalid Expression")

# Function to clear entry field
def clear():
    entry.delete(0, tk.END)


# Function to delete last character (backspace)
def backspace():
    current_text = entry.get()
    entry.delete(len(current_text) - 1)


# Function to calculate square of the number
def square():
    try:
        num = float(entry.get())
        entry.delete(0, tk.END)
        entry.insert(0, num ** 2)
    except ValueError:
        messagebox.showerror("Error", "Enter a valid number!")


# Function to calculate square root
def square_root():
    try:
        num = float(entry.get())
        if num < 0:
            messagebox.showerror("Error", "Cannot calculate square root of a negative number!")
        else:
            entry.delete(0, tk.END)
            entry.insert(0, math.sqrt(num))
    except ValueError:
        messagebox.showerror("Error", "Enter a valid number!")


# Function to exit the application
def exit_app():
    window.destroy()


# Button Layout
buttons = [
    ('7', 1, 0), ('8', 1, 1), ('9', 1, 2), ('/', 1, 3),
    ('4', 2, 0), ('5', 2, 1), ('6', 2, 2), ('*', 2, 3),
    ('1', 3, 0), ('2', 3, 1), ('3', 3, 2), ('-', 3, 3),
    ('0', 4, 0), ('.', 4, 1), ('+', 4, 2), ('=', 4, 3),
]

# Create buttons dynamically
for text, row, col in buttons:
    action = lambda x=text: myclick(x) if x != "=" else equal()
    tk.Button(window, text=text, font=font_style, padx=20, pady=15, command=action).grid(row=row, column=col, padx=5, pady=5)

# Special Buttons
tk.Button(window, text="C", font=font_style, padx=20, pady=15, command=clear, bg="red", fg="white").grid(row=5, column=0, padx=5, pady=5)
tk.Button(window, text="←", font=font_style, padx=20, pady=15, command=backspace, bg="gray", fg="white").grid(row=5, column=1, padx=5, pady=5)
tk.Button(window, text="x²", font=font_style, padx=20, pady=15, command=square, bg="lightblue").grid(row=5, column=2, padx=5, pady=5)
tk.Button(window, text="√x", font=font_style, padx=20, pady=15, command=square_root, bg="lightgreen").grid(row=5, column=3, padx=5, pady=5)
tk.Button(window, text="Exit", font=font_style, padx=70, pady=15, command=exit_app, bg="black", fg="white").grid(row=6, column=0, columnspan=4, padx=5, pady=10)
```