

Name: Rohan Sanjay Bhagwat

Title: Building a Graphical User Interface (GUI) using Python

Introduction:

In this case study, we will explore the development of a Graphical User Interface (GUI) Calculator using Python. The objective of the project was to create a user-friendly calculator application with a sleek interface that allows users to perform basic arithmetic operations.

Background:

The need for a GUI calculator arise from the desire to provide users with a more intuitive and visually appealing way to perform calculations. Traditional command-line calculators lack the user-friendly interface that a GUI can offer, making it challenging for non-technical users to interact with the application.

Project Goals:

1. Develop a GUI-based calculator using Python.
2. Enable basic arithmetic operations such as addition, subtraction, multiplication, and division.
3. Create a clean and visually appealing user interface.
4. Handle user inputs and display results in real-time.
5. Ensure error handling for invalid inputs.

Technologies used

- Python: The core programming language.
- Tkinter: The standard GUT toolkit for python.

Implementation:

Code:

```
import tkinter as tk
from tkinter import messagebox

calculation = ""

def add_to_calculation(symbol):
    global calculation
    calculation += str(symbol)
    text_result.delete(1.0, "end")
    text_result.insert(1.0, calculation)

def evaluate_calculation():
    global calculation
    try:
        calculation = str(eval(calculation))
        text_result.delete(1.0, "end")
        text_result.insert(1.0, calculation)
    except:
        clear_field()
        text_result.insert(1.0, "Error")

def clear_field():
    global calculation
    calculation = ""
    text_result.delete(1.0, "end")

def on_closing():
    if messagebox.askyesno(title="Quit?" , message= "Do you want to quit?"):
        root.destroy()

root = tk.Tk()
root.geometry("300x275")
root.title("GUI Calculator")
root.protocol("WM_DELETE_WINDOW", on_closing)
```

```
text_result = tk.Text(root, height= 2, width = 16, font=("Arial", 24))
text_result.grid(columnspan= 5)

btn_1 = tk.Button(root, text= "1", command = lambda: add_to_calculation(1),
width= 5, font= ("Arial",14))
btn_1.grid(row= 2, column= 1)

btn_2 = tk.Button(root, text= "2", command = lambda: add_to_calculation(2),
width= 5, font= ("Arial",14))
btn_2.grid(row= 2, column= 2)

btn_3 = tk.Button(root, text= "3", command = lambda: add_to_calculation(3),
width= 5, font= ("Arial",14))
btn_3.grid(row= 2, column= 3)

btn_4 = tk.Button(root, text= "4", command = lambda: add_to_calculation(4),
width= 5, font= ("Arial",14))
btn_4.grid(row= 3, column= 1)

btn_5 = tk.Button(root, text= "5", command = lambda: add_to_calculation(5),
width= 5, font= ("Arial",14))
btn_5.grid(row= 3, column= 2)

btn_6 = tk.Button(root, text= "6", command = lambda: add_to_calculation(6),
width= 5, font= ("Arial",14))
btn_6.grid(row= 3, column= 3)

btn_7 = tk.Button(root, text= "7", command = lambda: add_to_calculation(7),
width= 5, font= ("Arial",14))
btn_7.grid(row= 4, column= 1)

btn_8 = tk.Button(root, text= "8", command = lambda: add_to_calculation(8),
width= 5, font= ("Arial",14))
btn_8.grid(row= 4, column= 2)

btn_9 = tk.Button(root, text= "9", command = lambda: add_to_calculation(9),
width= 5, font= ("Arial",14))
btn_9.grid(row= 4, column= 3)

btn_0 = tk.Button(root, text= "0", command = lambda: add_to_calculation(0),
width= 5, font= ("Arial",14))
btn_0.grid(row= 5, column= 2)

btn_plus = tk.Button(root, text= "+", command = lambda:
add_to_calculation("+"), width= 5, font= ("Arial",14))
btn_plus.grid(row= 2, column= 4)
```

```
btn_minus = tk.Button(root, text= "-", command = lambda: add_to_calculation("-"), width= 5, font= ("Arial",14))
btn_minus.grid(row= 3, column= 4)

btn_mul = tk.Button(root, text= "*", command = lambda: add_to_calculation("*"), width= 5, font= ("Arial",14))
btn_mul.grid(row= 4, column= 4)

btn_div = tk.Button(root, text= "/", command = lambda: add_to_calculation("/"), width= 5, font= ("Arial",14))
btn_div.grid(row= 5, column= 4)

btn_open = tk.Button(root, text= "(", command = lambda: add_to_calculation("("), width= 5, font= ("Arial",14))
btn_open.grid(row= 5, column= 1)

btn_close = tk.Button(root, text= ")", command = lambda: add_to_calculation(")"), width= 5, font= ("Arial",14))
btn_close.grid(row= 5, column= 3)

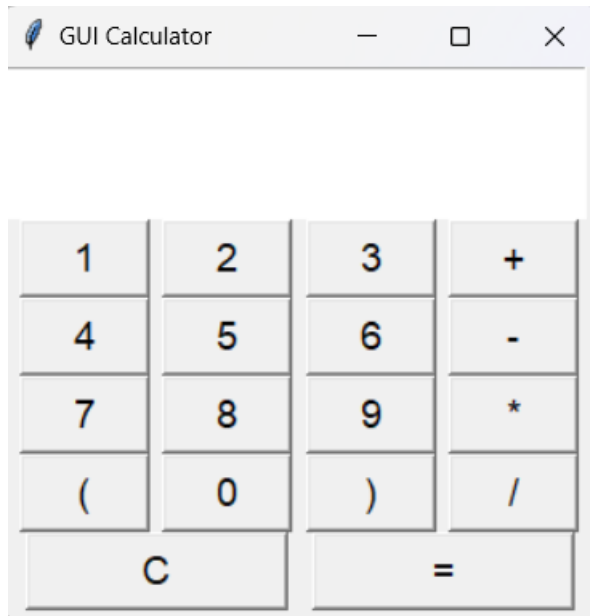
btn_clear = tk.Button(root, text= "C", command = clear_field, width= 11, font= ("Arial",14))
btn_clear.grid(row= 6, column= 1, columnspan= 2)

btn_equals = tk.Button(root, text= "=", command = evaluate_calculation, width= 11, font= ("Arial",14))
btn_equals.grid(row= 6, column= 3, columnspan= 2)

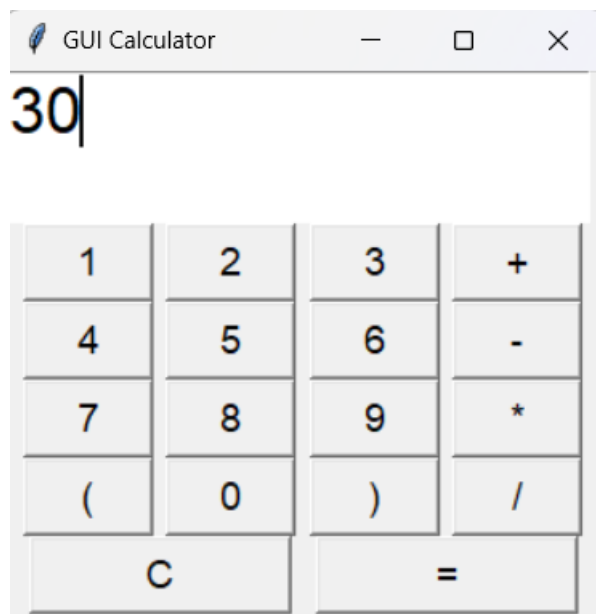
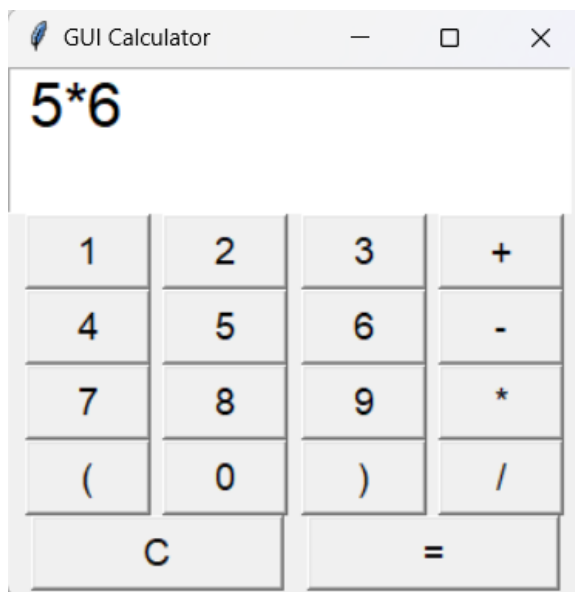
root.mainloop()
```

Output:

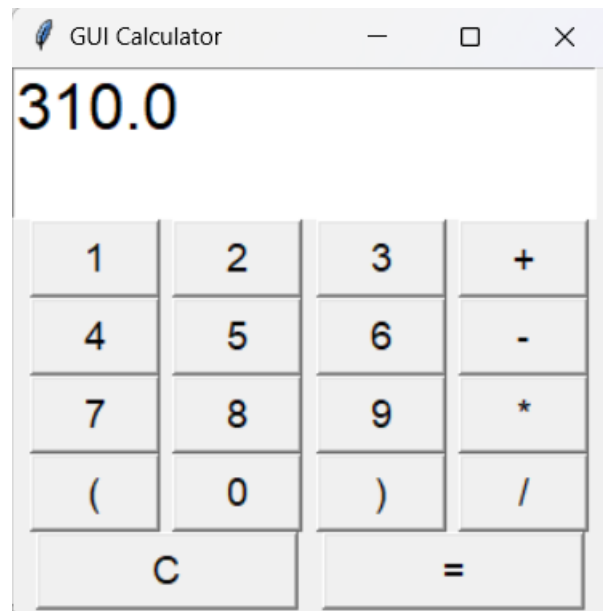
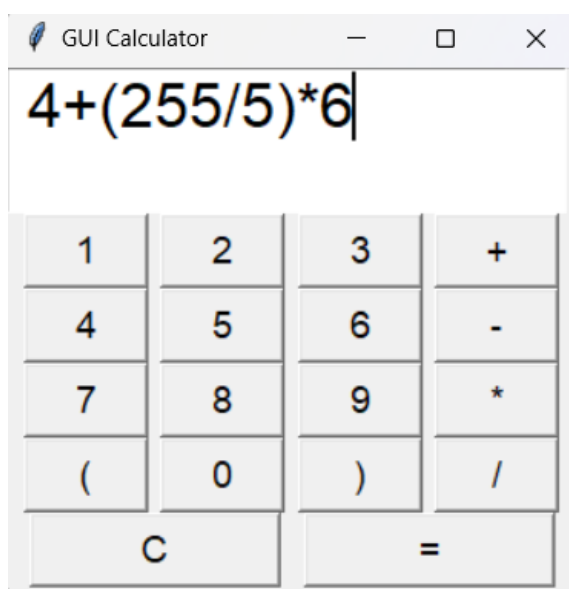
Graphical User Interface:



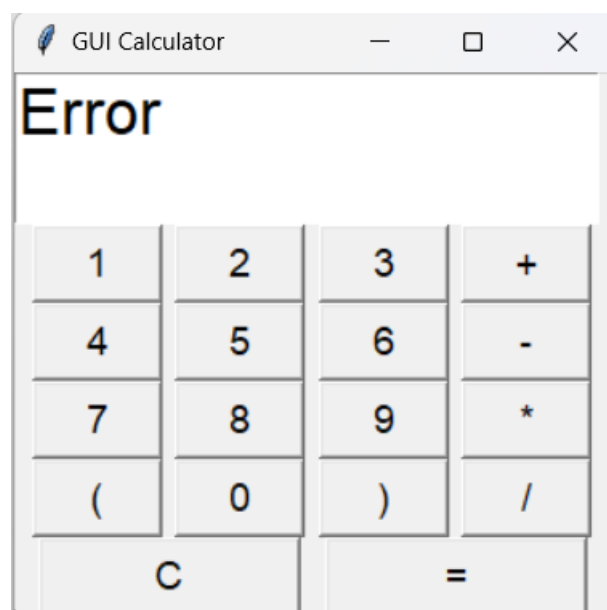
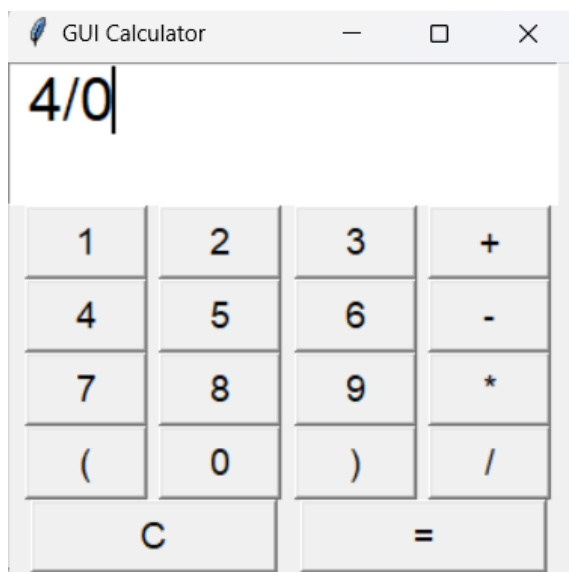
Arithmetic operations:



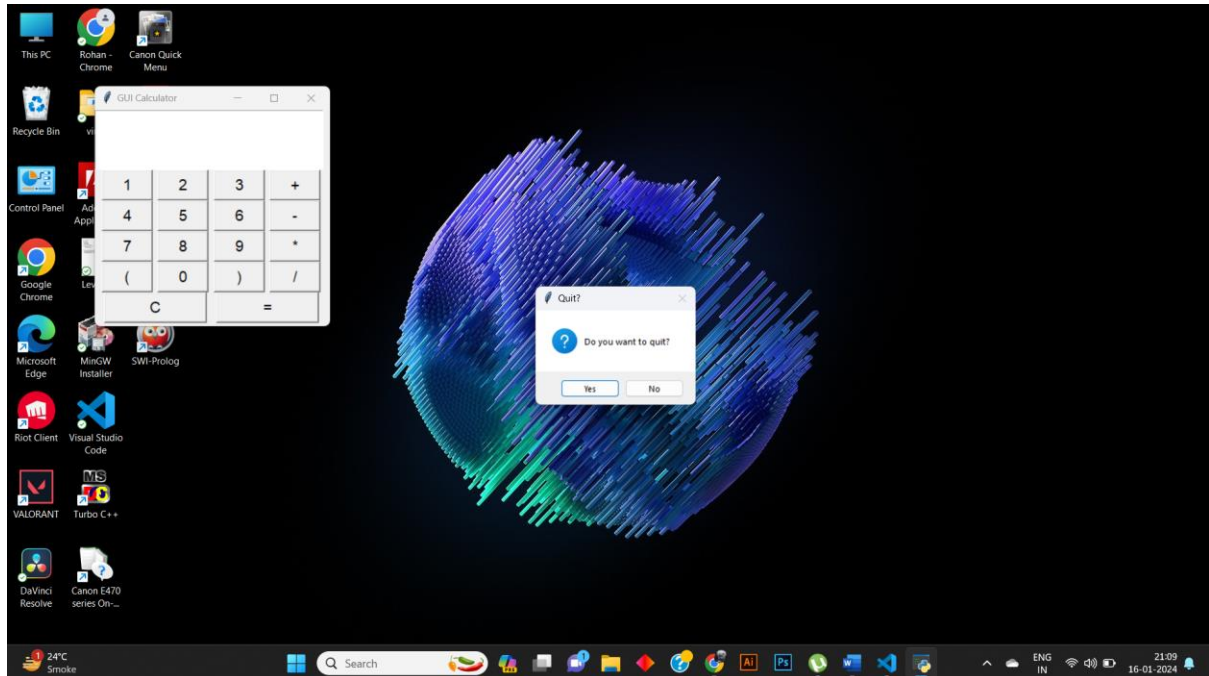
Complex Arithmetic operations:



Error Handling:



Closing function:



Conclusion:

The GUI calculator project successfully met its objectives by providing a user-friendly interface for basic arithmetic operations. The combination of Python and Tkinter proved to be effective for rapid development. The project serves as a foundation for future enhancements, such as adding advanced mathematical functions and improving the overall user experience.