Gowtham SB
www.linkedin.com/in/sbgowtham/ Instagram - @dataengineeringtamil

🧠 What is `tkinter`?

`tkinter` is Python’s **built-in GUI library** for creating desktop apps.
You can build windows, buttons, input boxes, and more — without needing any extra installation.

🛠️ Project: Mini Add 2 Numbers Calculator

☑ What It Will Do:

- Let user enter **two numbers**
- Press a button to **add them**
- Show the **result**
- Handle **invalid inputs** (like letters)

🧾 Full Python Code

```
import tkinter as tk  # Import tkinter module

# Step 1: Create the main window
root = tk.Tk()
root.title("Add 2 Numbers Calculator")
root.geometry("300x200")  # Width x Height

# Step 2: Create input labels and entry boxes
tk.Label(root, text="Enter Number 1:").pack(pady=5)
num1_entry = tk.Entry(root)  # Input box for number 1
num1_entry.pack()

tk.Label(root, text="Enter Number 2:").pack(pady=5)
num2_entry = tk.Entry(root)  # Input box for number 2
num2_entry.pack()
```

```
# Step 3: Label to show result
result_label = tk.Label(root, text="Result will appear here")
result_label.pack(pady=10)

# Step 4: Add numbers on button click
def add_numbers():
    try:
        num1 = float(num1_entry.get())  # Get input from box 1
        num2 = float(num2_entry.get())  # Get input from box 2
        total = num1 + num2
        result_label.config(text=f"Result: {total}")  # Show result
    except ValueError:
        result_label.config(text="Please enter valid numbers")

# Step 5: Create the Add button
tk.Button(root, text="Add", command=add_numbers).pack(pady=5)

# Step 6: Run the app
root.mainloop()
```

📦 Explanation of Components

Code	Description
`tk.Tk()`	Creates the main GUI window
`tk.Label()`	Adds static text labels
`tk.Entry()`	Input box for typing numbers
`pack(pady=5)`	Adds vertical spacing (padding)
`get()`	Gets the input from `Entry` field
`Button(..., command=add_numbers)`	Runs function when button clicked

`mainloop()`	Starts and keeps the GUI open

🧪 Example Output:

1. User types: `10` and `20`
2. Clicks **Add**
3. Shows: `Result: 30.0`

🚀 To Run:

1. Save the file as `add_calculator.py`

Run it using:

```
python add_calculator.py
```

2.

🎁 Bonus Ideas:

Want to extend this calculator?

- Add Subtract/Multiply/Divide buttons
- Clear button
- Input validation messages
- Modern themes with `ttk`

✅ 📄 Resume Project Description

Project Title: *Mini Calculator GUI using Python (Tkinter)*
Tech Stack: Python, tkinter (Standard GUI Library)

Description:
Built a lightweight desktop application using Python's tkinter library that allows users to input two numbers, perform addition, and display the result in real time. Implemented robust input validation to handle incorrect or non-numeric entries. Designed a clean and responsive GUI with labels, input fields, buttons, and dynamic result display. Focused on usability, readability, and modular code for future extension to support subtraction, multiplication, and division.

🎤 💼 How to Talk About This Project in an Interview (Story Style)

> "When I started learning GUI programming, I wanted to build something very simple but useful. That's when I created a mini calculator that adds two numbers. The idea was to go beyond just console-based programs and explore how real desktop apps work – like user interaction, error handling, and GUI layouts.
>
> I used Python's built-in tkinter library – so no extra installation required. The app starts with two input fields and a button. Once the user enters two numbers and clicks the button, the sum is shown dynamically. I also added input

validation – so if someone enters 'abc', it won’t crash but instead shows a helpful message.

It may seem like a small project, but I learned a lot: structuring the GUI, linking logic to interface, handling user input, and making it beginner-friendly. It gave me confidence to later expand this into a full calculator with more operations and design enhancements.

If I had more time, I’d enhance it further with ttk styling, keyboard shortcuts, and packaging it into an .exe using tools like pyinstaller.”

About the Author

Gowtham SB is a **Data Engineering expert, educator, and content creator** with a passion for **big data technologies, as well as cloud and Gen AI** . With years of experience in the field, he has worked extensively with **cloud platforms, distributed systems, and data pipelines**, helping professionals and aspiring engineers master the art of data engineering.

Beyond his technical expertise, Gowtham is a **renowned mentor and speaker**, sharing his insights through engaging content on **YouTube and LinkedIn**. He has built one of the **largest Tamil Data Engineering communities**, guiding thousands of learners to excel in their careers.

Through his deep industry knowledge and hands-on approach, Gowtham continues to **bridge the gap between learning and real-world implementation**, empowering individuals to build **scalable, high-performance data solutions**.

Socials

🎥**YouTube** - https://www.youtube.com/@dataengineeringvideos

📷**Instagram** - https://instagram.com/dataengineeringtamil

📷**Instagram** - https://instagram.com/thedatatech.in

🤝**Connect for 1:1** - https://topmate.io/dataengineering/

💼**LinkedIn** - https://www.linkedin.com/in/sbgowtham/

🌐**Website** - https://codewithgowtham.blogspot.com

💻**GitHub** - http://github.com/Gowthamdataengineer

Gowtham SB
www.linkedin.com/in/sbgowtham/ Instagram - @dataengineeringtamil

🗨 **Whats App** - https://lnkd.in/g5JrHw8q

📧 **Email** - atozknowledge.com@gmail.com

📱 **All My Socials** - https://lnkd.in/gf8k3aCH