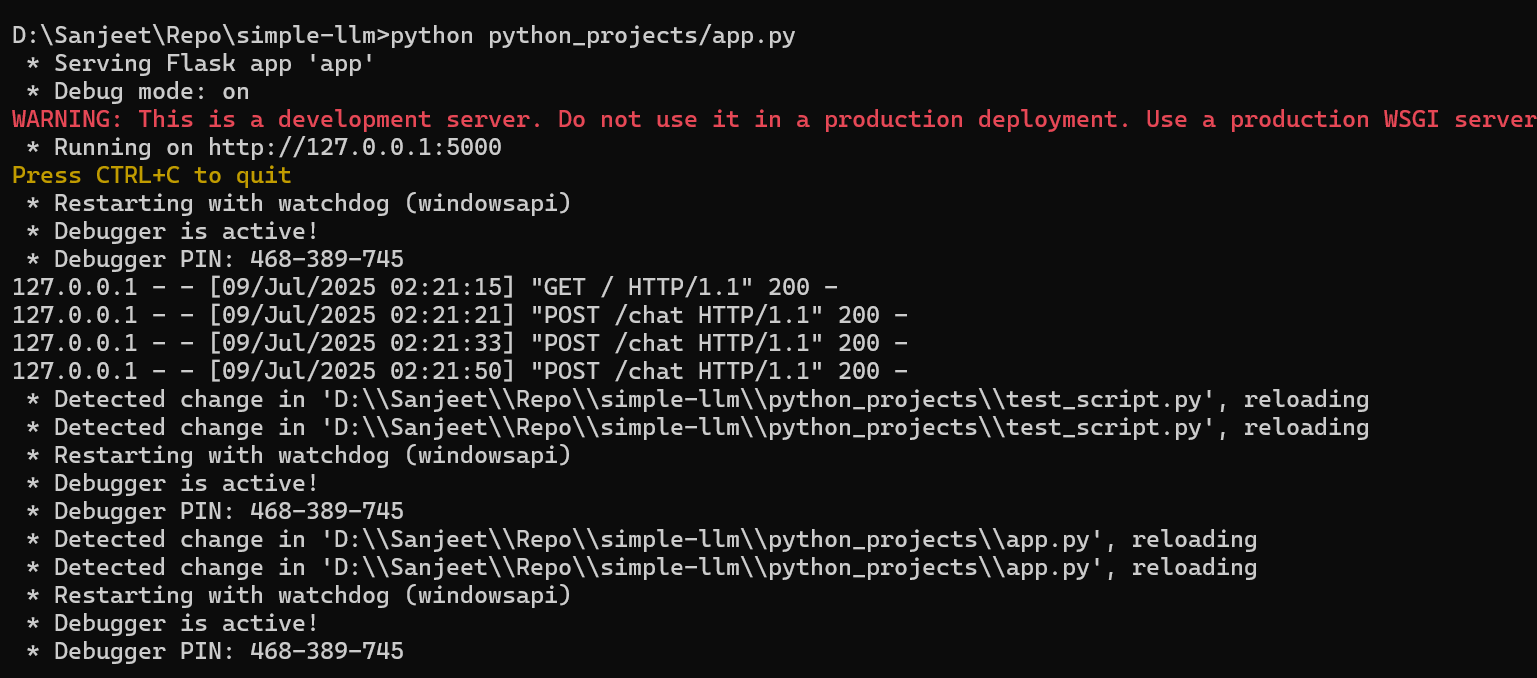
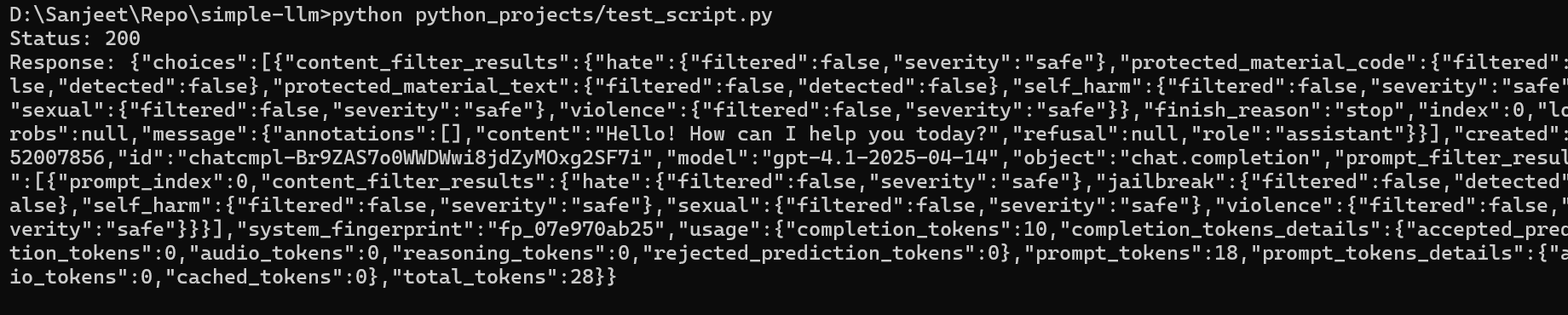
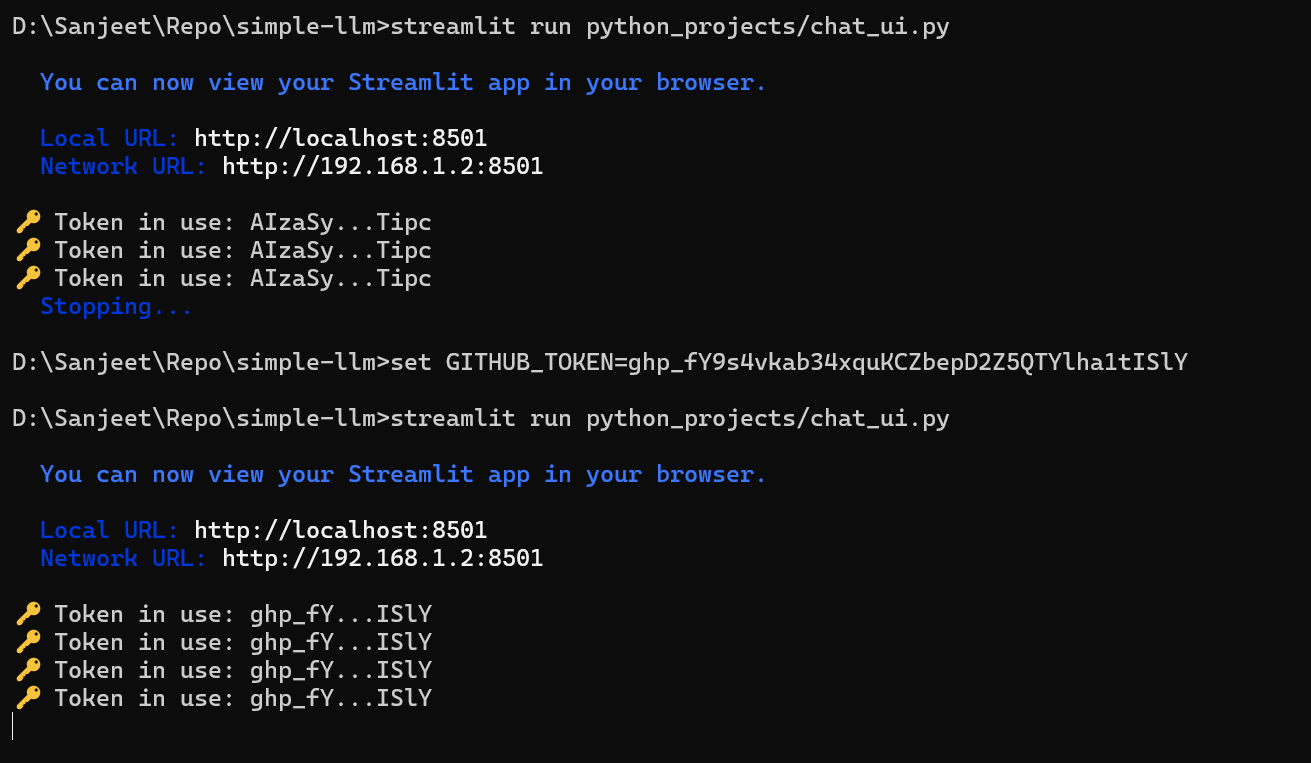
To run flask on localhost:5000



Test github openai token



To run streamlit on localhost:8501



**Features:**

* Maintains full message history
* Exits on exit or quit
* Prints assistant responses cleanly

Let me know if you'd like to:

* Stream messages token-by-token
* Save chat logs to a file
* Add retry/backoff logic
* Turn this into a web/GUI app

**2. Use from a Flask/Streamlit Web UI**

Let me know if you want a:

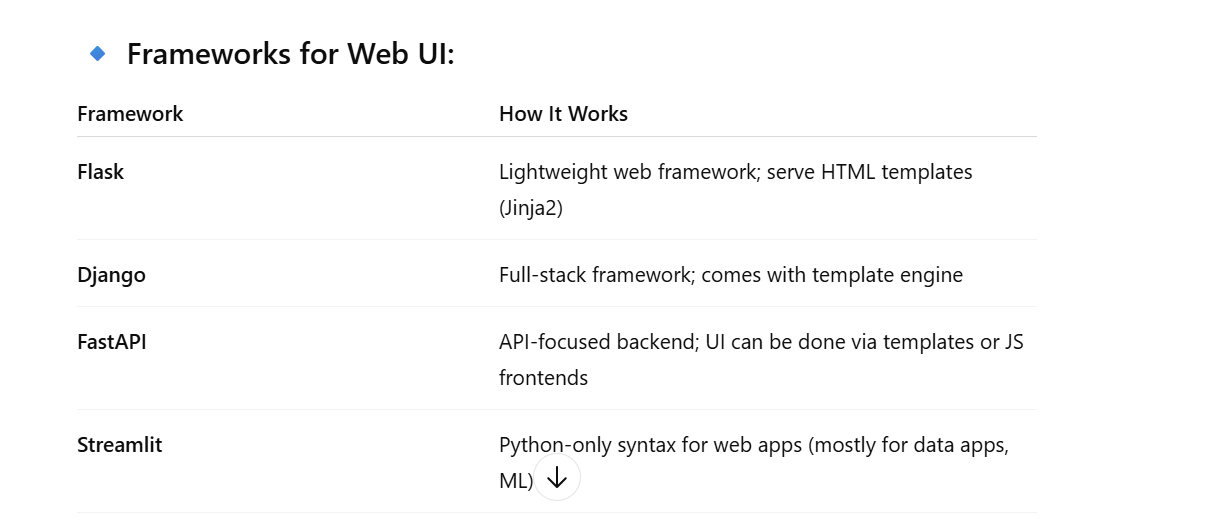
* Streamlit app (quick web interface)
* Flask backend with a chat UI
* CLI tool with argparse

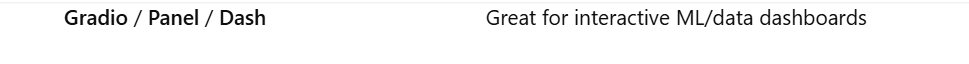
Use **raw REST API calls** in Python

n Python, frontend UI can be shown in several ways depending on your **target platform**—web, desktop, or mobile. Here's a quick breakdown:

**✅ 1. Web UI (Browser-based)**

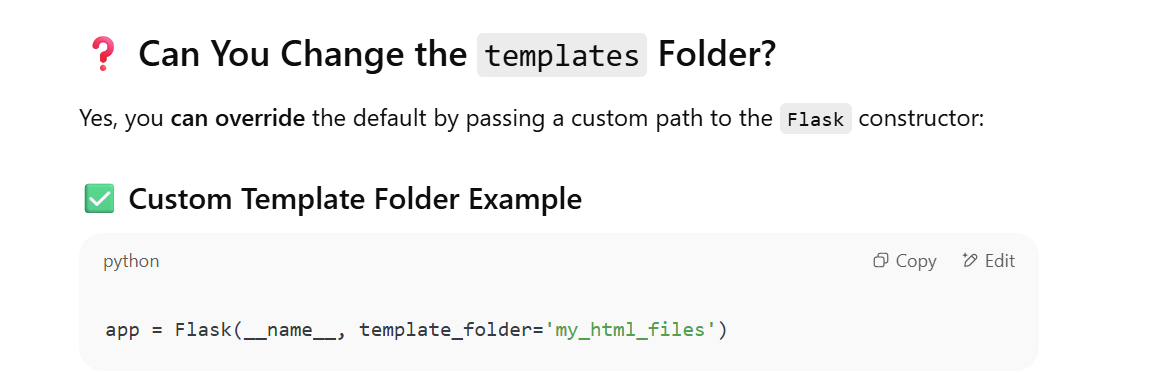
Python itself **doesn’t render frontend in the browser**, but you can use frameworks to build web apps with HTML/CSS/JS as UI.





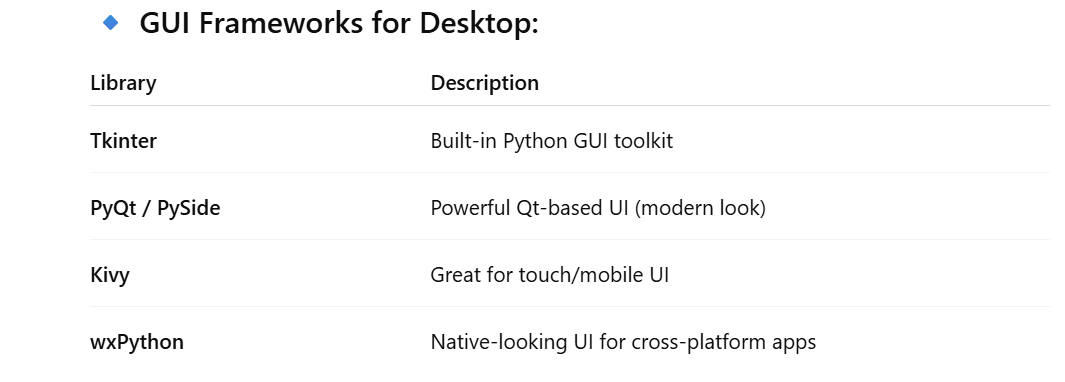


in **Flask**, HTML files (templates) are by **default** expected to be inside a folder named **templates/** — this is how Flask knows where to look when you call render\_template().



**✅ 2. Desktop GUI**

You can create actual windowed applications with buttons, textboxes, etc.

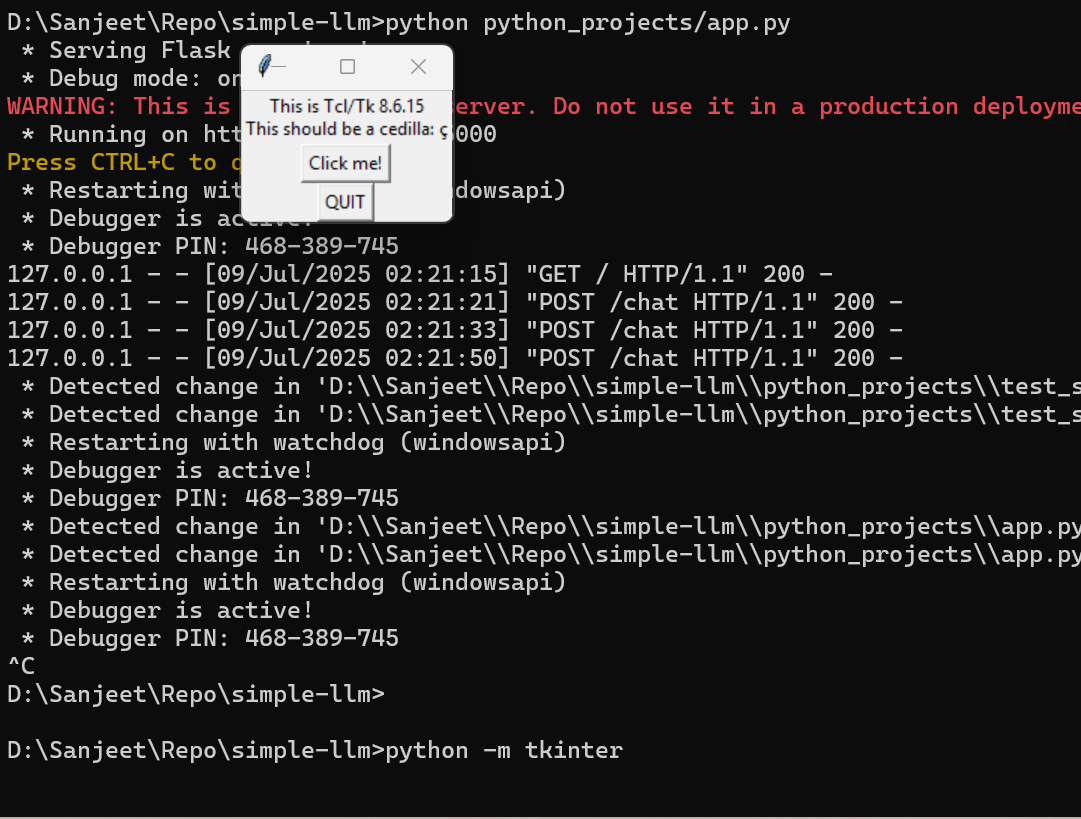




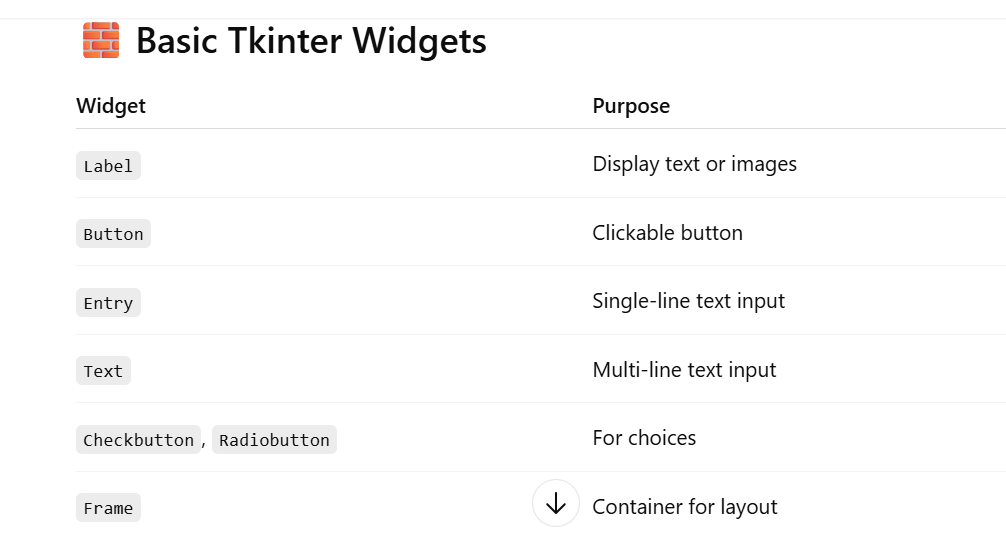
**Tkinter comes built-in with standard Python installations on most systems (Windows, macOS, Linux).**

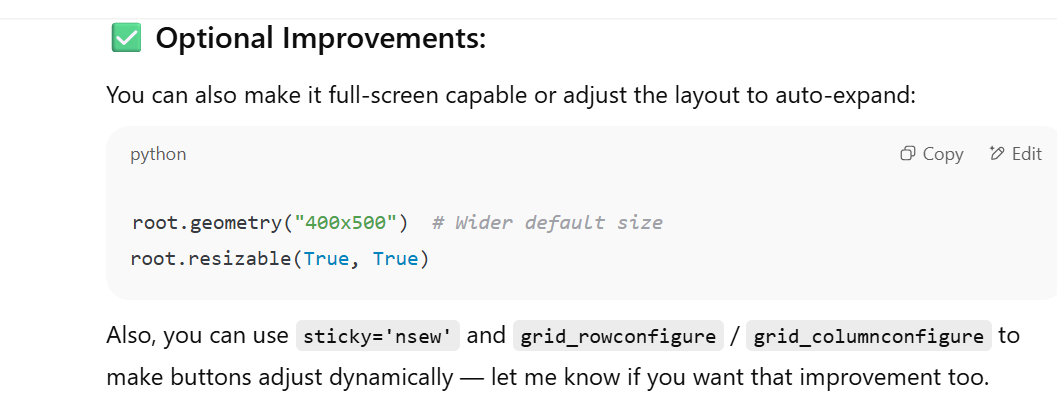
**Tkinter is Python’s standard library for building simple desktop GUI apps (buttons, windows, labels, text boxes, etc.).**

**python -m tkinter**

****

**If a small window pops up titled "Tk", then Tkinter is installed ✅**

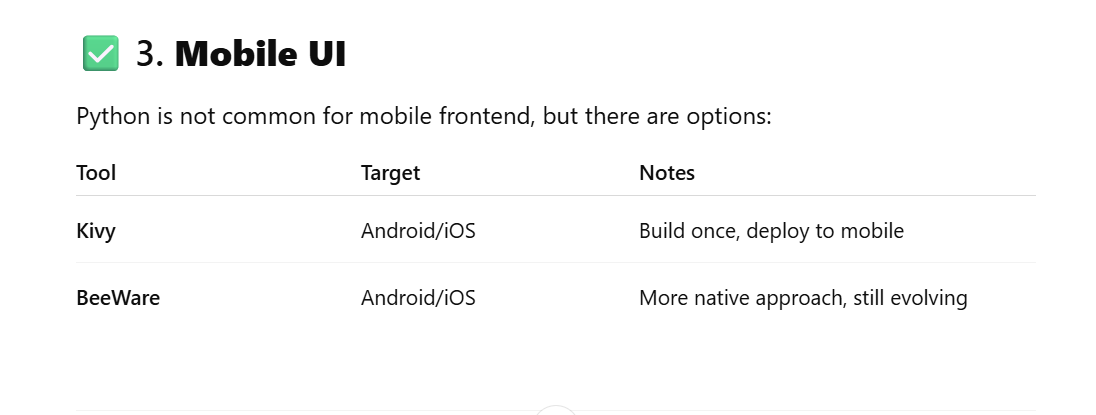
* **You don’t need to install anything extra if you already have Python.**
* **Use pack(), grid(), or place() to control layout.**
* **Run .py file in your terminal or IDE (like VS Code or IDLE) to see the window.**
* specific GUI example (like a calculator, login form, or chatbot UI)?

****

****

**✅ Features**

* **✅ Number input and operators**
* **✅ Real-time display in input box**
* **✅ = to evaluate expression**
* **✅ C button to clear everything**

**✅ 4. Jupyter Notebook UI**

If you're doing data science or ML, use:

* **IPyWidgets** for sliders/buttons
* **Streamlit** / **Gradio** for interactive dashboards from notebooks

