## **🧠 What is Streamlit?**

**Streamlit** is a Python library to create **web apps for data projects** — without needing HTML, CSS, or JavaScript.

It turns your Python scripts into interactive web apps **just by running a script**.

## **✅ Step-by-Step Guide to Streamlit (Beginner Friendly)**

### **✅ Step 1: Install Streamlit**

Open terminal or command prompt and run:

pip install streamlit

### **✅ Step 2: Create Your First App**

Create a Python file called app.py:

# app.py

import streamlit as st

st.title("📊 My First Streamlit App")

st.header("Welcome to the Demo App")

st.write("This is a simple app built with Streamlit!")

# Interactive element

name = st.text\_input("What's your name?")

if name:

st.success(f"Hello, {name}! 👋")

# Slider example

age = st.slider("Select your age", 1, 100, 25)

st.write("You selected:", age)

### **✅ Step 3: Run the App**

In the terminal, run:

streamlit run app.py

Your default browser will open at [http://localhost:8501](http://localhost:8501/)

## **🚀 How It Works (Like Magic)**

* st.title(), st.write() etc. are **Streamlit commands** that display content.
* Input elements like st.text\_input() create **interactive UI** (like forms, sliders).
* You **write Python**, and Streamlit converts it to a **web app**.

## **📦 Bonus: Add a Simple Calculator App**

Replace your app.py with this code:

import streamlit as st

st.title("🧮 Simple Calculator")

num1 = st.number\_input("Enter first number")

num2 = st.number\_input("Enter second number")

operation = st.selectbox("Choose operation", ["Add", "Subtract", "Multiply", "Divide"])

if st.button("Calculate"):

if operation == "Add":

result = num1 + num2

elif operation == "Subtract":

result = num1 - num2

elif operation == "Multiply":

result = num1 \* num2

elif operation == "Divide":

result = num1 / num2 if num2 != 0 else "Cannot divide by zero"

st.success(f"Result: {result}")

## **✅ What You Learned**

| **Topic** | **Description** |
| --- | --- |
| st.write() | Write text, variables |
| st.text\_input() | Take user input |
| st.number\_input() | Numeric input |
| st.selectbox() | Dropdown selection |
| st.button() | Run logic on click |
| st.success() | Display results nicely |

## **🛠️ What Can You Build?**

* Data dashboards 📊
* ML model apps 🤖
* Image/video tools 📸
* Chatbots 💬
* Resume analyzers 📄
* SQL explorers 🔍

### **📝 Resume Project Description – *Chat with PDF AI Web App***

**Project Title:** Chat with PDF – AI-powered Document Q&A Web App  
 **Tech Stack:** Python, Streamlit, LangChain, OpenAI API, FAISS, PyPDF

**Description:**Developed a Streamlit-based web application that enables users to interact with PDF documents via natural language queries. Utilized LangChain to split and embed PDF text, stored embeddings in a FAISS vector database, and used OpenAI GPT models for semantic question-answering. The application supports real-time Q&A on document content, offering a practical tool for research, legal, and academic use cases.

**Key Highlights:**

* Implemented document chunking and embedding using LangChain.
* Integrated FAISS for fast similarity search across document segments.
* Created an intuitive frontend using Streamlit to upload and chat with PDF files.
* Ensured seamless user interaction with OpenAI-powered LLM responses.
* Designed to run locally without needing full-stack web development skills.

### **🎙️ Interview Story – How to Talk About It Confidently**

"One of the projects I’m most excited about is something I built called **Chat with PDF**. The idea came from a real-world problem — we often have long documents like legal contracts, research papers, or customer policies, and reading through them manually is time-consuming. So I thought — why not build a small app where anyone could upload a PDF and just **ask questions** like chatting with an assistant?

I used **Streamlit** to build a lightweight frontend — it’s perfect because I didn’t need to write any HTML or JavaScript. For the AI part, I used **LangChain**, which made it super easy to split the PDF into chunks and convert those into embeddings using **OpenAI**. Then I stored those embeddings in a **FAISS vector database** for fast retrieval.

So the flow is: user uploads a PDF, the app extracts and embeds the content, and when they type a question, it finds the most relevant chunks and feeds that to a GPT model for the answer.

It was exciting because I learned how to stitch multiple tools together — Streamlit, OpenAI, FAISS — and deliver a useful, working app that can run even on a local machine. In fact, during the demo, I uploaded a 20-page resume guide and asked things like 'What are the top 3 resume mistakes?' and got accurate, meaningful responses.

I see this kind of AI app becoming a real productivity booster — and it taught me not just how to build AI pipelines, but how to **make them usable** for non-technical users."

### **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/](https://instagram.com/dataengineeringtamil)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

💬𝐖𝐡𝐚𝐭𝐬 𝐀𝐩𝐩 - https://lnkd.in/g5JrHw8q

📧𝐄𝐦𝐚𝐢𝐥 - atozknowledge.com@gmail.com

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