Integrating Gemini APIs with Python and Building a Chatbot App with Streamlit

Open Lab | Digital Summit 2024

Goal

In this OpenLab, you will learn to create a simple, user-friendly chat bot using Streamlit and powered by Google Cloud's Gemini API. Here's what you'll be working on,

- **Streamlit** is an easy-to-use platform that helps you build interactive applications quickly. With it, you'll create a chatbot where users can ask questions and get accurate answers
- The chatbot will use **Gemini's AI**, a powerful language model by Google, to understand the user's questions and provide intelligent, helpful responses

By the end of the OpenLab, you'll have built a chatbot that anyone can use to ask questions and get smart answers, all with a simple interface!

Pre-Requisites

The following installations are required to complete this lab and run successfully,

- Google Account
- Python Installation
- Any Text Editor(VS Code/Pycharm/Notepad++)

Technology Involved

- Python
- Streamlit (HTML + CSS)

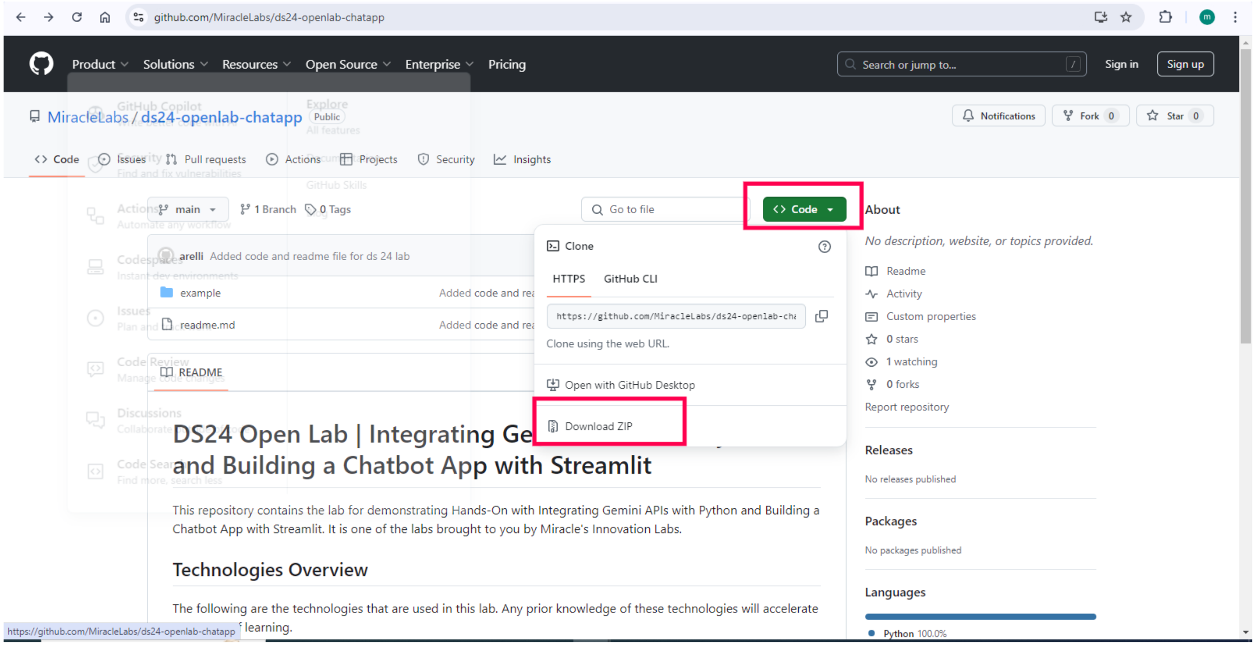
Lab Steps

Let's get started with the lab!

Step #1 | Download Code Repository

Get Started

Download the code from the following GitHub Repo link, https://github.com/MiracleLabs/ds24-openlab-chatapp



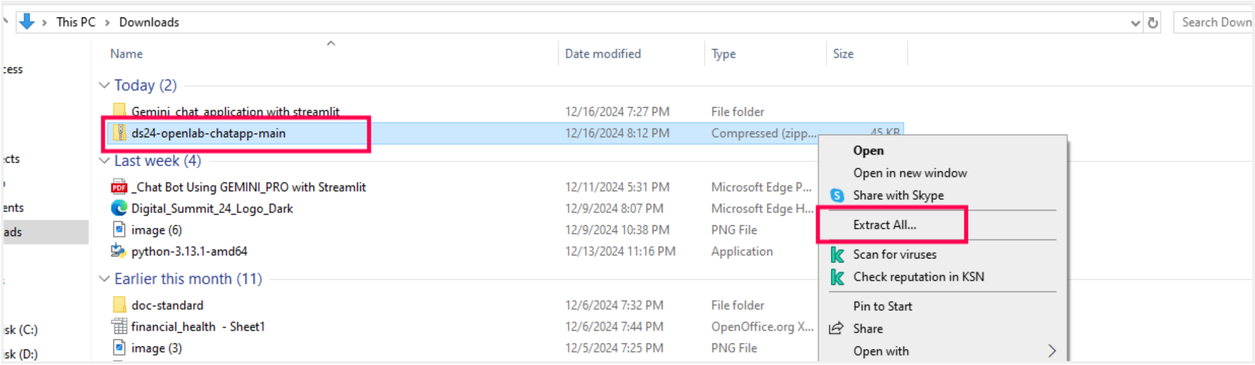
DS24 Open Lab | Integrating Ge
and Building a Chatbot App with Streamlit

This repository contains the lab for demonstrating Hands-On with Integrating Gemini APIs with Python and Building a Chatbot App with Streamlit. It is one of the labs brought to you by Miracle's Innovation Labs.

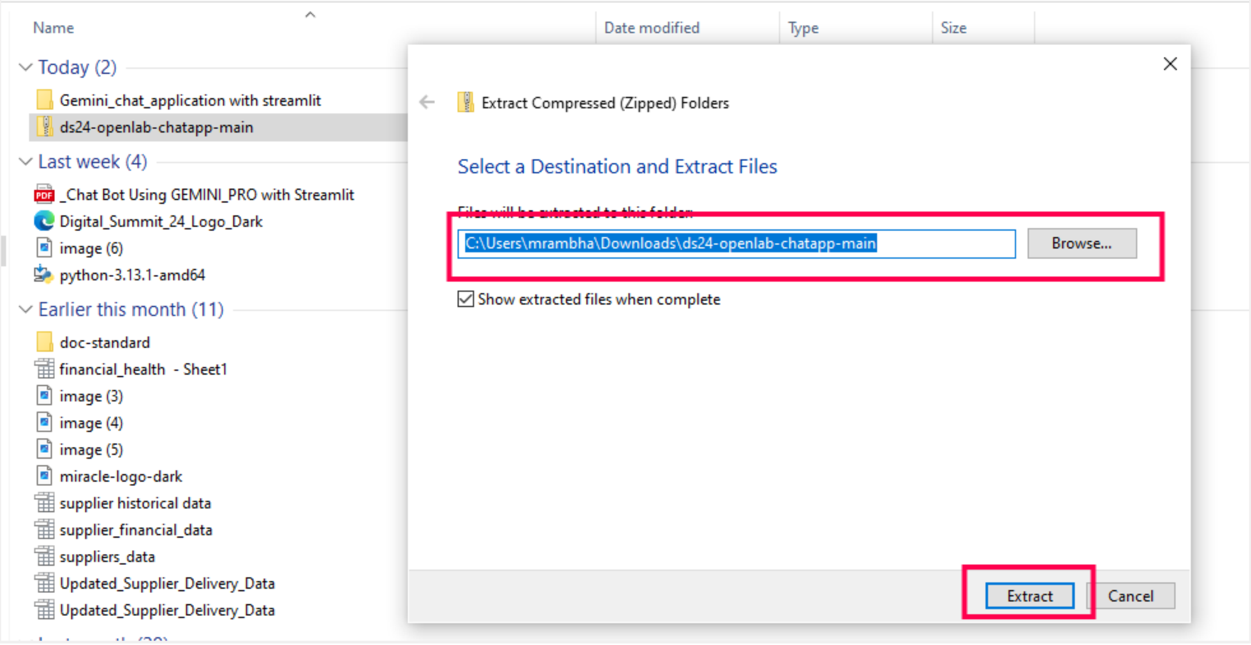
Technologies Overview

The following are the technologies that are used in this lab. Any prior knowledge of these technologies will accelerate learning.

After downloading, unzip it



Name	Date modified	Type	Size
Gemini_chat_application with streamlit	12/16/2024 7:27 PM	File folder	
ds24-openlab-chatapp-main	12/16/2024 8:12 PM	Compressed (zipp...	45 KB
_Chat Bot Using GEMINI_PRO with Streamlit	12/11/2024 5:31 PM	Microsoft Edge P...	
Digital_Summit_24_Logo_Dark	12/9/2024 8:07 PM	Microsoft Edge H...	
image (6)	12/9/2024 10:38 PM	PNG File	
python-3.13.1-amd64	12/13/2024 11:16 PM	Application	
doc-standard	12/6/2024 7:32 PM	File folder	
financial_health - Sheet1	12/6/2024 7:44 PM	OpenOffice.org X...	
image (3)	12/5/2024 7:25 PM	PNG File	



Today (2)

- Gemini_chat_application with streamlit
- ds24-openlab-chatapp-main

Last week (4)

- _Chat Bot Using GEMINI_PRO with Streamlit
- Digital_Summit_24_Logo_Dark
- image (6)
- python-3.13.1-amd64

Earlier this month (11)

- doc-standard
- financial_health - Sheet1
- image (3)
- image (4)
- image (5)
- miracle-logo-dark
- supplier historical data
- supplier_financial_data
- suppliers_data
- Updated_Supplier_Delivery_Data
- Updated_Supplier_Delivery_Data

Extract Compressed (Zipped) Folders

Select a Destination and Extract Files

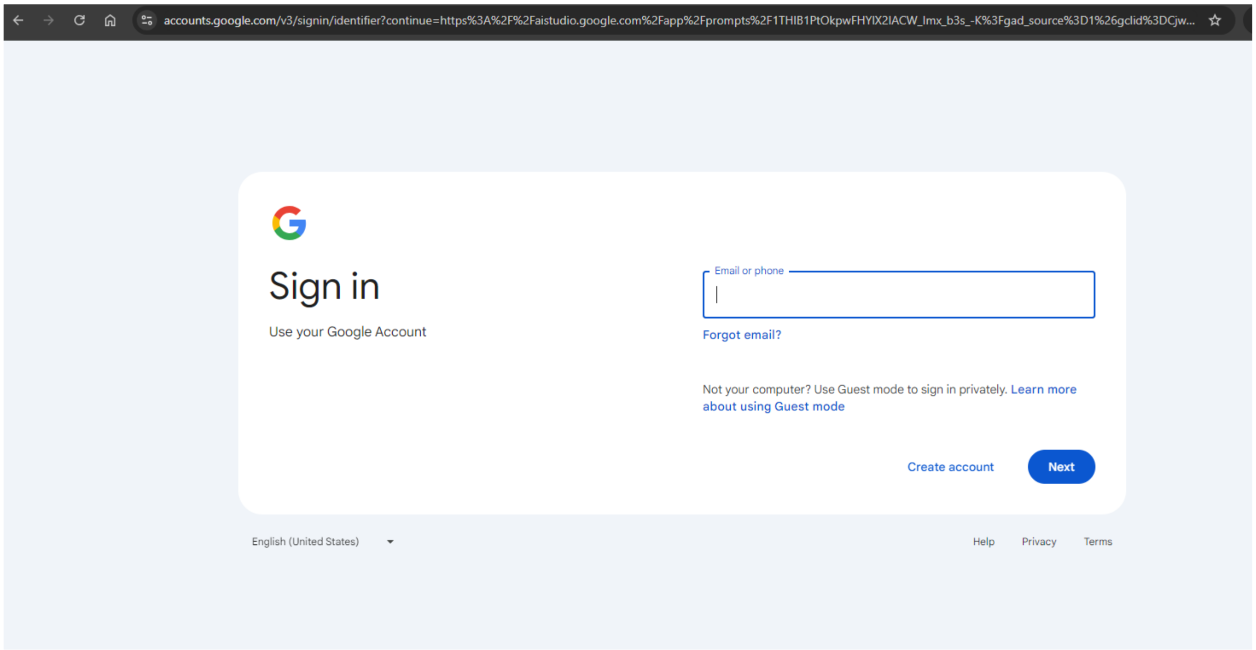
Files will be extracted to this folder:

C:\Users\mrambha\Downloads\ds24-openlab-chatapp-main

☑ Show extracted files when complete

Step #2 | Access Google AI Studio

Click on **Google AI Studio.**



accounts.google.com/v3/signin/identifier?continue=https%3A%2F%2Faistudio.google.com%2Fapp%2Fprompts%2F1THIB1PtOkpwFHYIX2IACW_Imx_b3s_-K%3Fgad_source%3D1%26gclid%3DCjw...

Sign in

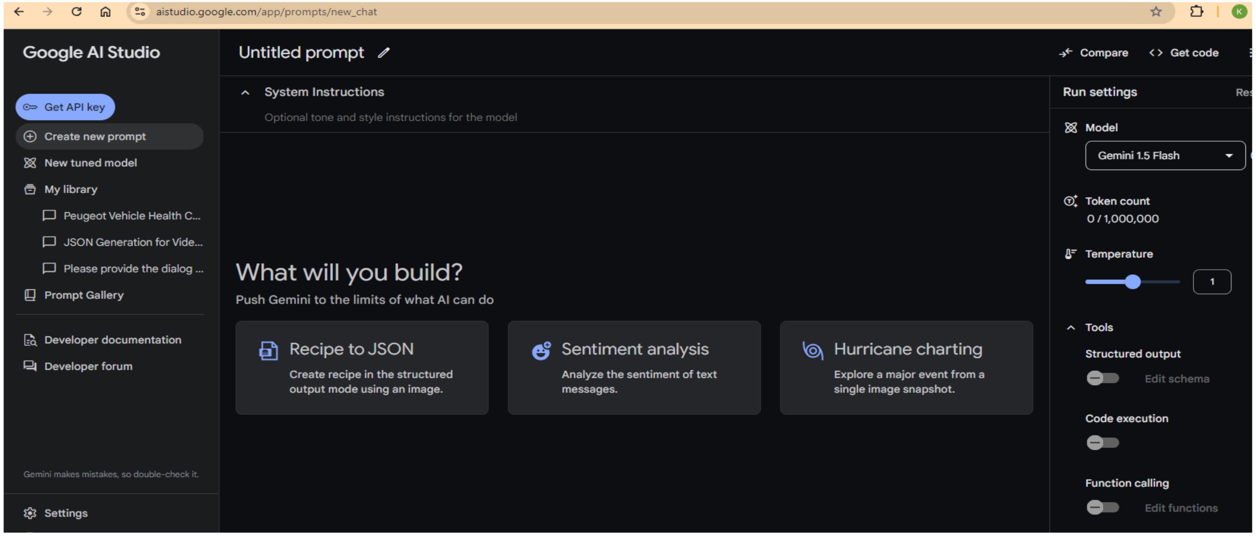
Use your Google Account

Email or phone

Forgot email?

Not your computer? Use Guest mode to sign in privately. Learn more about using Guest mode

After logging into your Google account, you'll be redirected to the AI Studio dashboard, where you need to accept the Terms of Service and click "Continue" to proceed.

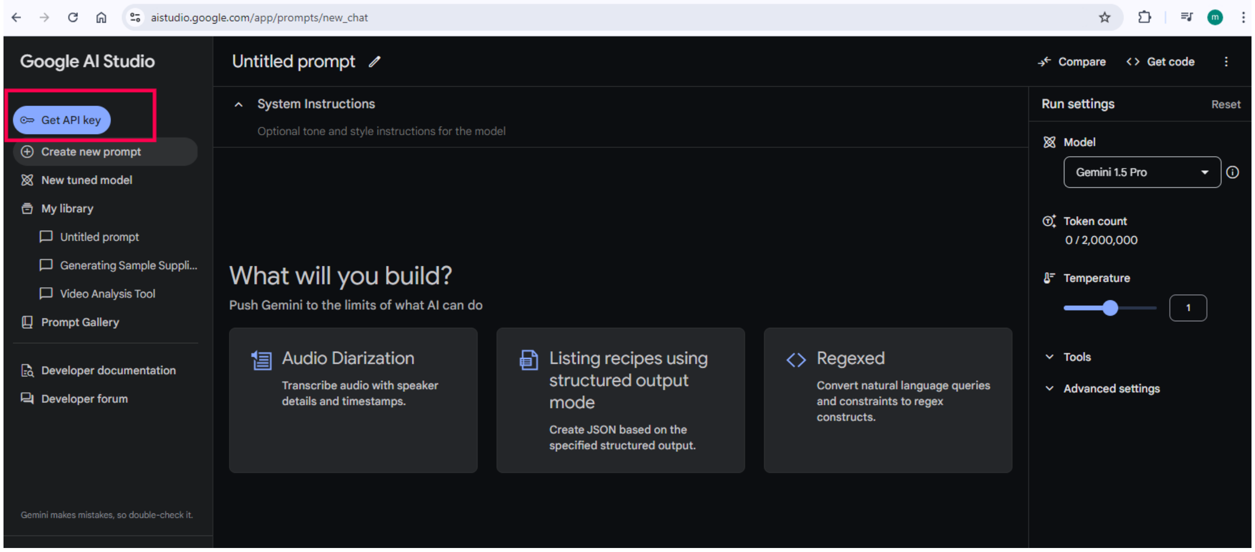

aistudio.google.com/app/prompts/new_chat
Google AI Studio
Get API key
Create new prompt
New tuned model
My library
Peugeot Vehicle Health C...
JSON Generation for Vide...
Please provide the dialog ...
Prompt Gallery
Developer documentation
Developer forum
Gemini makes mistakes, so double-check it.
Settings
Untitled prompt
Compare
Get code
System Instructions
Optional tone and style instructions for the model
What will you build?
Push Gemini to the limits of what AI can do
Recipe to JSON
Create recipe in the structured output mode using an image.
Sentiment analysis
Analyze the sentiment of text messages.
Hurricane charting
Explore a major event from a single image snapshot.
Run settings
Tools
Structured output
Edit schema
Code execution
Function calling
Edit functions

Model: Gemini 1.5 Flash

Token count: 0 / 1,000,000

Temperature: 1

Click on '**Get API Key**' in the top left corner to proceed to the screen shown below.

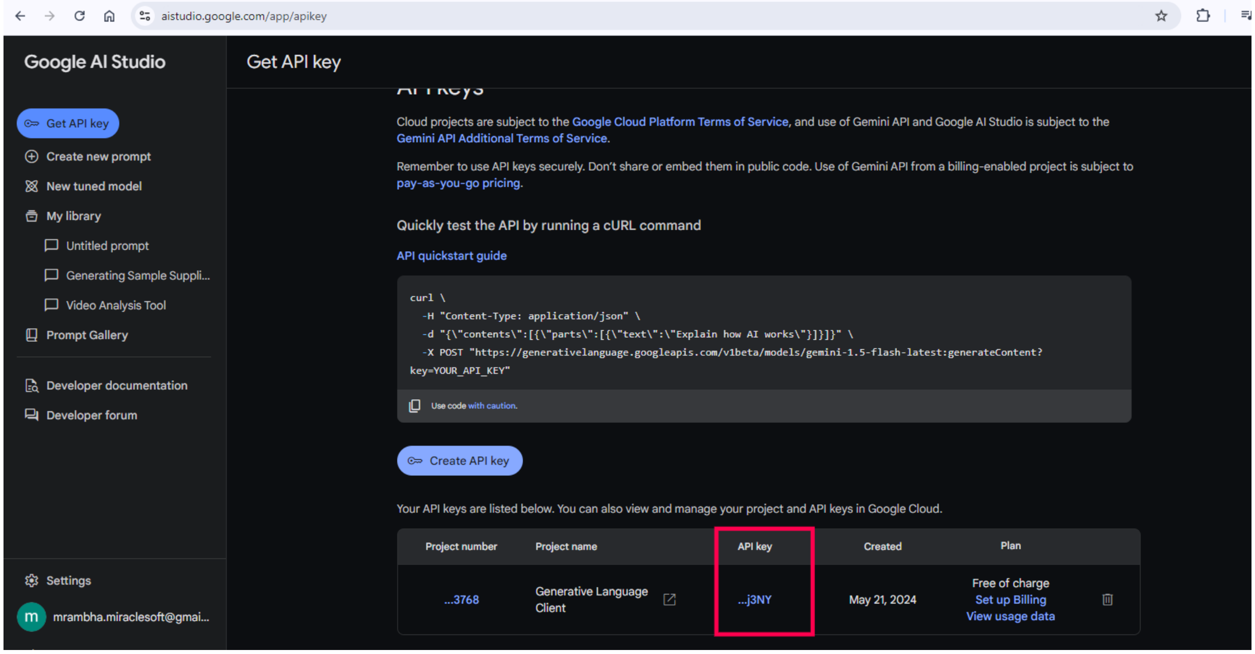

aistudio.google.com/app/prompts/new_chat
Google AI Studio
Get API key
Create new prompt
New tuned model
My library
Untitled prompt
Generating Sample Suppli...
Video Analysis Tool
Prompt Gallery
Developer documentation
Developer forum
Gemini makes mistakes, so double-check it.
Untitled prompt
Compare
Get code
System Instructions
Optional tone and style instructions for the model
What will you build?
Push Gemini to the limits of what AI can do
Audio Diarization
Transcribe audio with speaker details and timestamps.
Listing recipes using structured output mode
Create JSON based on the specified structured output.
Regexed
Convert natural language queries and constraints to regex constructs.
Run settings
Reset
Tools
Advanced settings

Model: Gemini 1.5 Pro

Token count: 0 / 2,000,000

Temperature: 1

Click on '**Create API Key**' as shown below.



Get API key

API keys

Cloud projects are subject to the Google Cloud Platform Terms of Service, and use of Gemini API and Google AI Studio is subject to the Gemini API Additional Terms of Service.

Remember to use API keys securely. Don't share or embed them in public code. Use of Gemini API from a billing-enabled project is subject to pay-as-you-go pricing.

Quickly test the API by running a cURL command

API quickstart guide

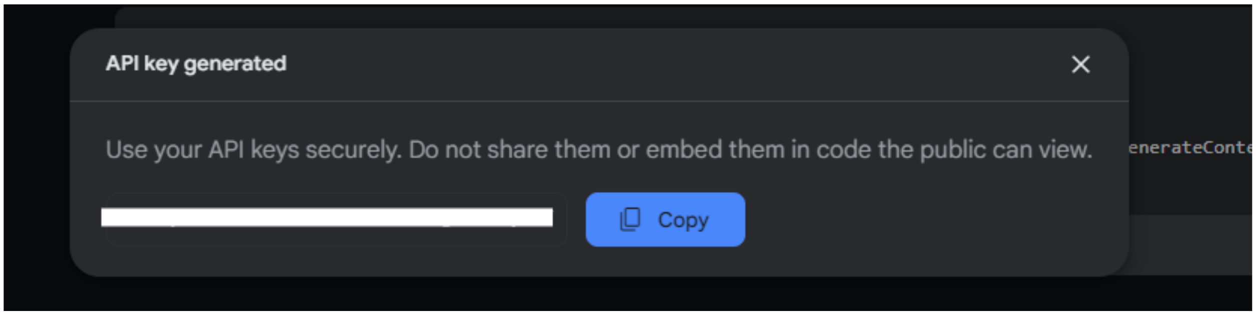
```
curl \
  -H "Content-Type: application/json" \
  -d "{\"contents\":[{\"parts\":[{\"text\":\"Explain how AI works\"}]}]}" \
  -X POST "https://generativelanguage.googleapis.com/v1beta/models/gemini-1.5-flash-latest:generateContent?
key=YOUR_API_KEY"
```

Use code with caution.

Your API keys are listed below. You can also view and manage your project and API keys in Google Cloud.

Project number	Project name	API key	Created	Plan
...3768	Generative Language Client	...j3NY	May 21, 2024	Free of charge Set up Billing View usage data

Now, copy the **API key** and use it in the code.



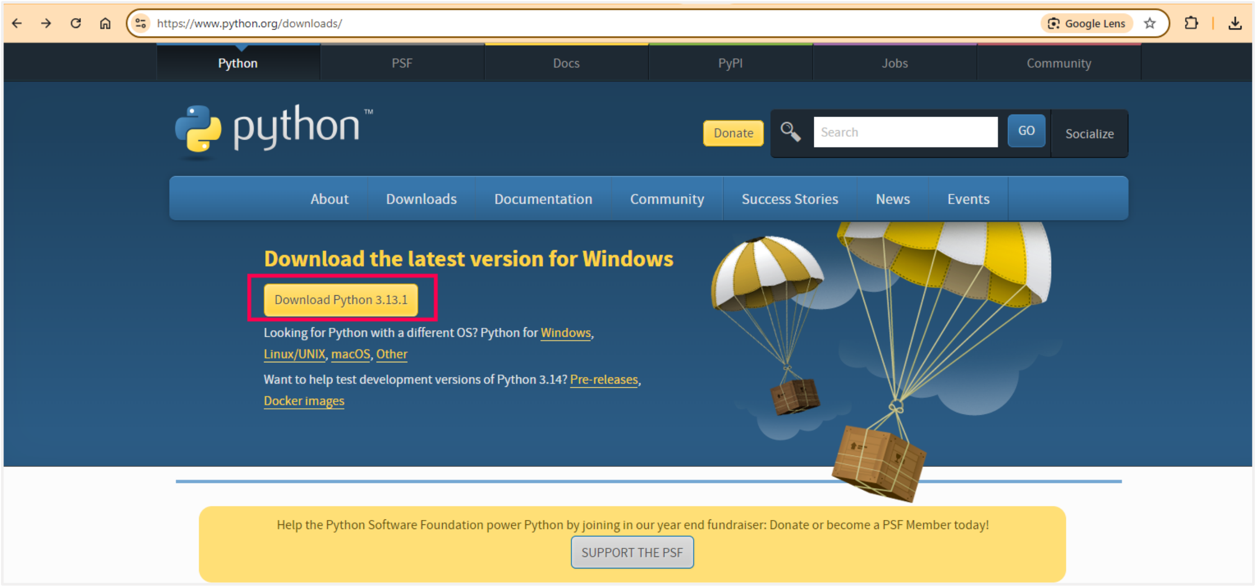
API key generated

Use your API keys securely. Do not share them or embed them in code the public can view.

[redacted]

Step #3 |Steps to Install and Set Up Python on Your Local

You must have the Python interpreter installed on your local machine. Click the link below to install Python (version). https://www.python.org/downloads/



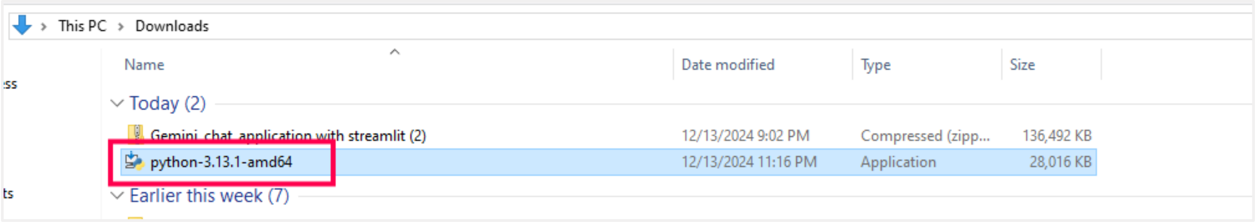
Download the latest version for Windows

Looking for Python with a different OS? Python for Windows, Linux/UNIX, macOS, Other

Want to help test development versions of Python 3.14? Pre-releases, Docker images

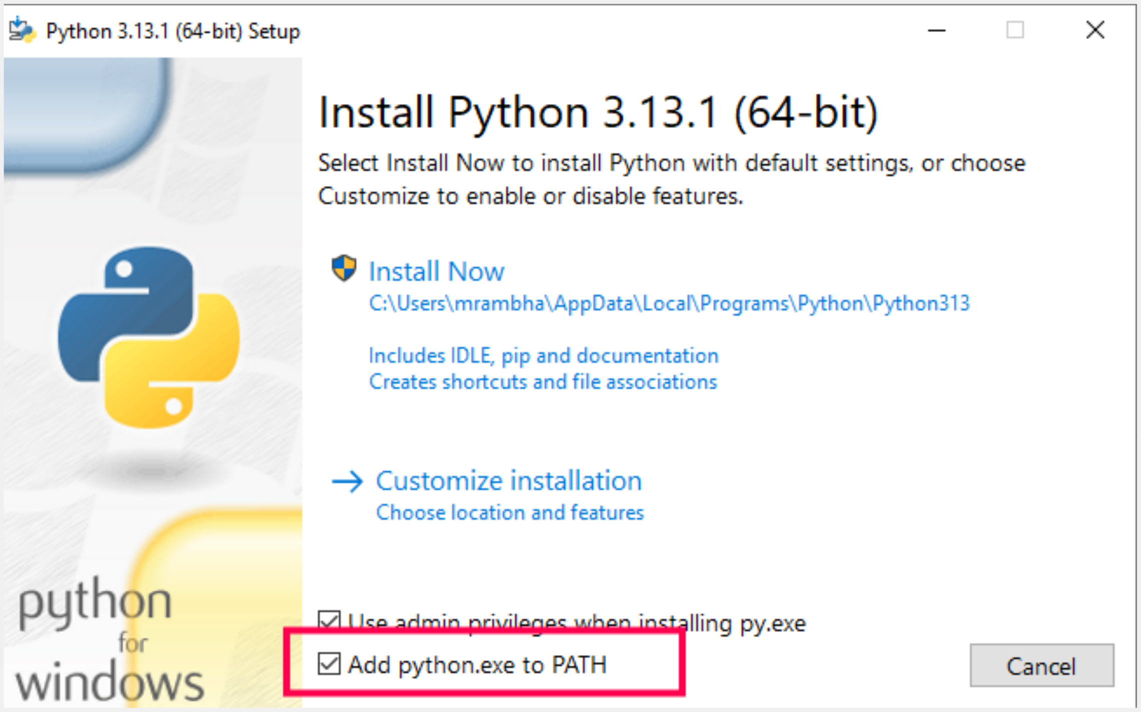
Help the Python Software Foundation power Python by joining in our year end fundraiser: Donate or become a PSF Member today!

Open the downloaded installer file, as shown in the image below.



Name	Date modified	Type	Size
Today (2)			
Gemini_chat_application with streamlit (2)	12/13/2024 9:02 PM	Compressed (zipp...	136,492 KB
python-3.13.1-amd64	12/13/2024 11:16 PM	Application	28,016 KB
Earlier this week (7)			

Click on "**Add Python to PATH**" checkbox.



Python 3.13.1 (64-bit) Setup

Install Python 3.13.1 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

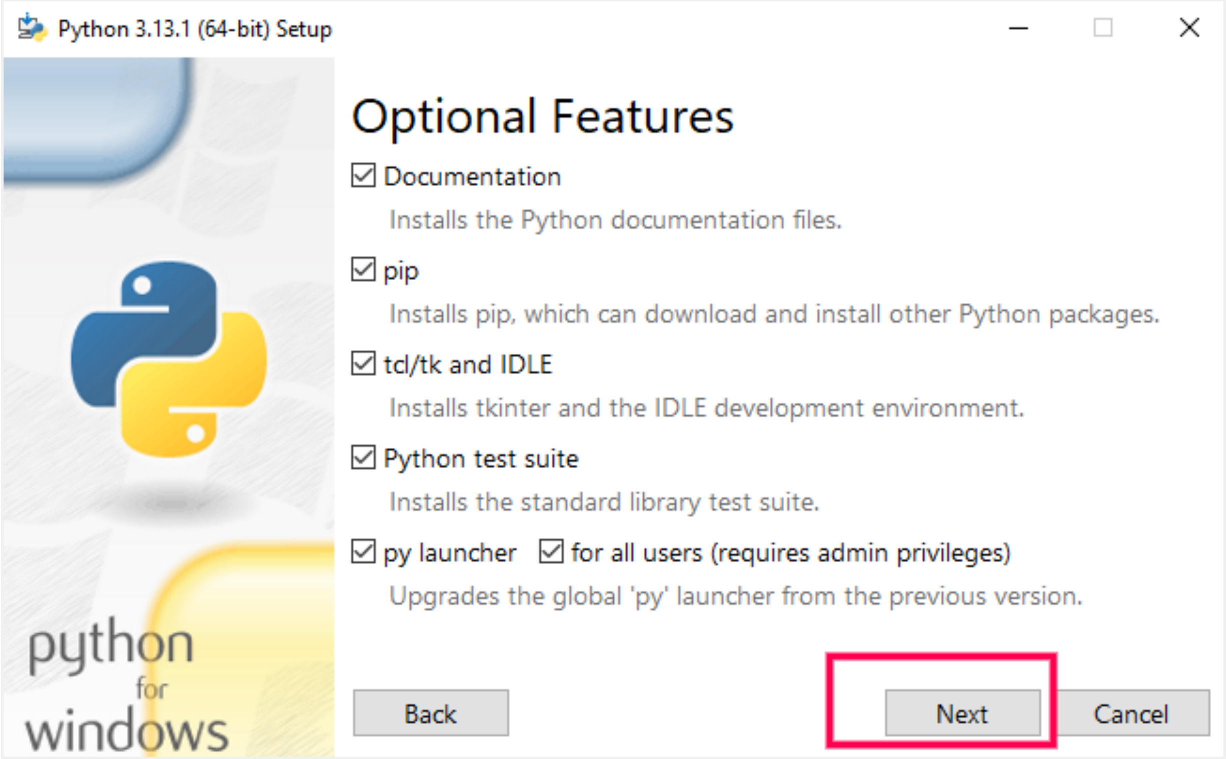
C:\Users\mrambha\AppData\Local\Programs\Python\Python313

Includes IDLE, pip and documentation
Creates shortcuts and file associations

☑ Use admin privileges when installing py.exe

☑ Add python.exe to PATH

Click on "**Next**" as shown in the image below.



Python 3.13.1 (64-bit) Setup

Optional Features

☑ Documentation
Installs the Python documentation files.

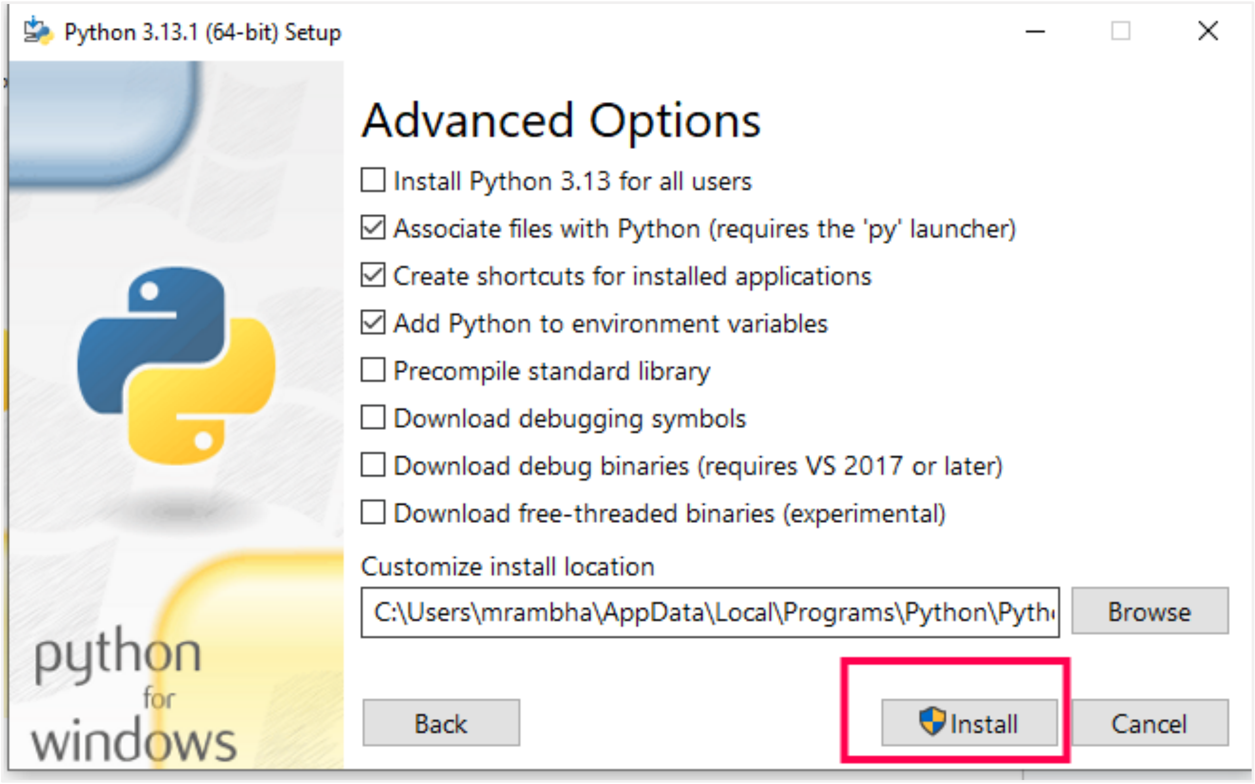
☑ pip
Installs pip, which can download and install other Python packages.

☑ tcl/tk and IDLE
Installs tkinter and the IDLE development environment.

☑ Python test suite
Installs the standard library test suite.

☑ py launcher ☑ for all users (requires admin privileges)
Upgrades the global 'py' launcher from the previous version.

Click on "**Install Now**" as shown in the image below.



Python 3.13.1 (64-bit) Setup

Advanced Options

- ☐ Install Python 3.13 for all users
- ☑ Associate files with Python (requires the 'py' launcher)
- ☑ Create shortcuts for installed applications
- ☑ Add Python to environment variables
- ☐ Precompile standard library
- ☐ Download debugging symbols
- ☐ Download debug binaries (requires VS 2017 or later)
- ☐ Download free-threaded binaries (experimental)

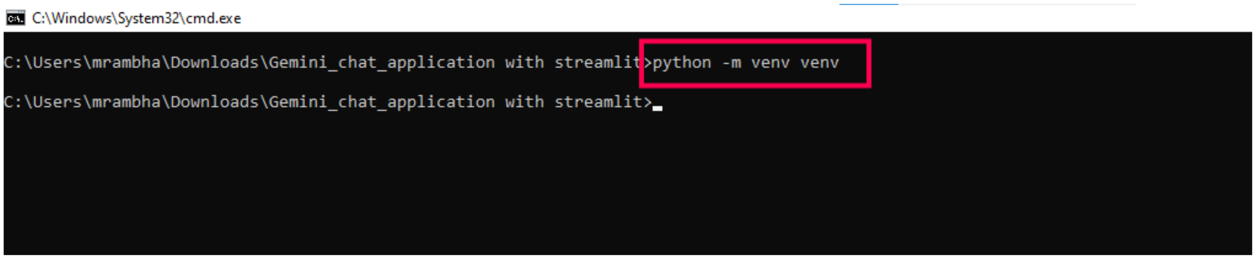
Customize install location

C:\Users\mrambha\AppData\Local\Programs\Python\Pyth

Step #4 |Steps to Set Up the Python Project

1. Project with Virtual Environment

Open a terminal or command prompt in the directory where you want to create the virtual environment. Then, use the following command to install the required libraries - **python -m venv venv.**



C:\Windows\System32\cmd.exe

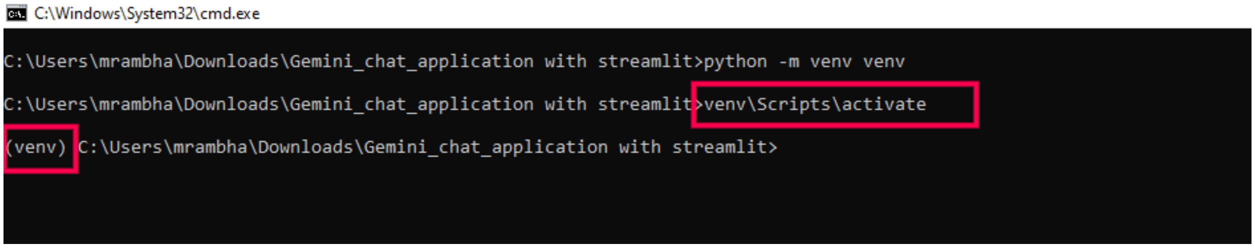
```
C:\Users\mrambha\Downloads\Gemini_chat_application with streamlit>python -m venv venv

C:\Users\mrambha\Downloads\Gemini_chat_application with streamlit>_
```

Once the environment is created, you need to activate it:

For Windows - **venv\Scripts\activate**

For Linux/Mac - **source venv/bin/activate**

After activation, your terminal prompt will look like the image below.



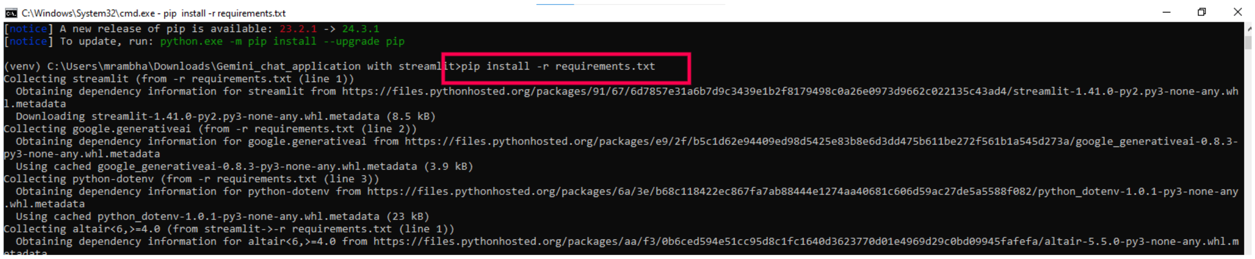
C:\Windows\System32\cmd.exe

```
C:\Users\mrambha\Downloads\Gemini_chat_application with streamlit>python -m venv venv

C:\Users\mrambha\Downloads\Gemini_chat_application with streamlit>venv\Scripts\activate

(venv) C:\Users\mrambha\Downloads\Gemini_chat_application with streamlit>
```

2. Install Required Packages

Install all the dependencies required for this usecase by using the below command,

pip install -r requirements.txt

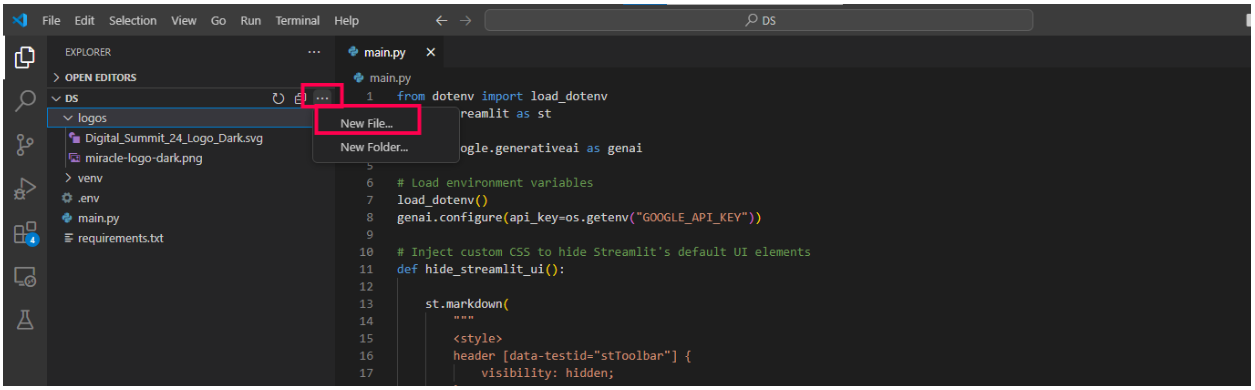


C:\Windows\System32\cmd.exe - pip install -r requirements.txt

```
[notice] A new release of pip is available: 23.2.1 -> 24.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip

(venv) C:\Users\mrambha\Downloads\Gemini_chat_application with streamlit>pip install -r requirements.txt
Collecting streamlit (from -r requirements.txt (line 1))
  Obtaining dependency information for streamlit from https://files.pythonhosted.org/packages/91/67/6d7857e31a6b7d9c3439e1b2f8179498c0a26e0973d9662c022135c43ad4/streamlit-1.41.0-py2.py3-none-any.wh
l.metadata
  Downloading streamlit-1.41.0-py2.py3-none-any.whl.metadata (8.5 kB)
Collecting google.generativeai (from -r requirements.txt (line 2))
  Obtaining dependency information for google.generativeai from https://files.pythonhosted.org/packages/e9/2f/b5c1d62e94409ed98d5425e83b8e6d3dd475b611be272f561b1a545d273a/google_generativeai-0.8.3-
py3-none-any.whl.metadata
  Using cached google_generativeai-0.8.3-py3-none-any.whl.metadata (3.9 kB)
Collecting python-dotenv (from -r requirements.txt (line 3))
  Obtaining dependency information for python-dotenv from https://files.pythonhosted.org/packages/6a/3e/b68c118422ec867fa7ab88444e1274aa40681c606d59ac27de5a5588f082/python_dotenv-1.0.1-py3-none-any
.whl.metadata
  Using cached python_dotenv-1.0.1-py3-none-any.whl.metadata (23 kB)
Collecting altair<6,>=4.0 (from streamlit->-r requirements.txt (line 1))
  Obtaining dependency information for altair<6,>=4.0 from https://files.pythonhosted.org/packages/aa/f3/0b6ced594e51cc95d8c1fc1640d3623770d01e4969d29c0bd09945fafefa/altair-5.5.0-py3-none-any.whl.m
etadata
```

3. Environment Variables (.env)

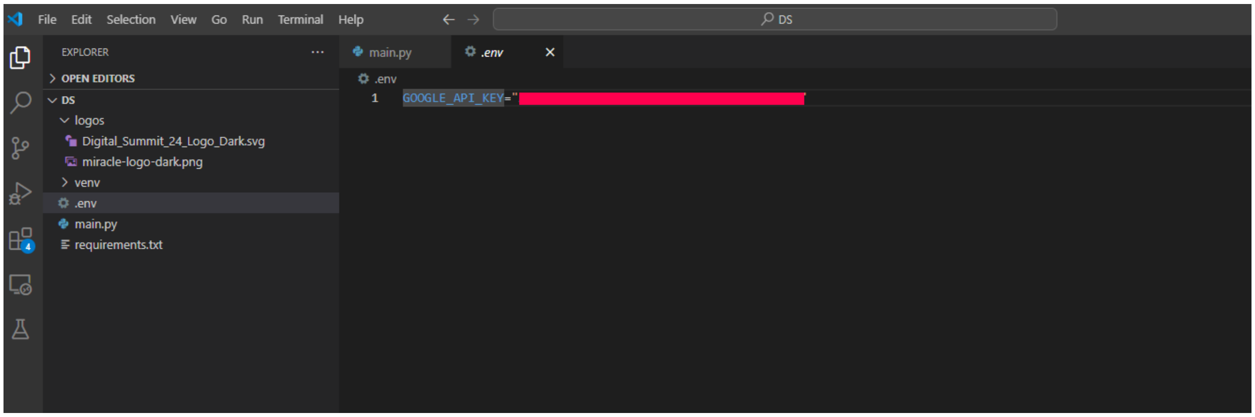
Create a new file, name it as .env in the root of your project directory.



```
from dotenv import load_dotenv
reamlit as st

ogle.generativeai as genai

# Load environment variables
load_dotenv()
genai.configure(api_key=os.getenv("GOOGLE_API_KEY"))

# Inject custom CSS to hide Streamlit's default UI elements
def hide_streamlit_ui():

    st.markdown(
        """
        <style>
        header [data-testid="stToolbar"] {
            visibility: hidden;
```

Inside the .env file, add your variables in the KEY=VALUE format, one per line **GOOGLE_API_KEY=<Your-Google-API-Key>**

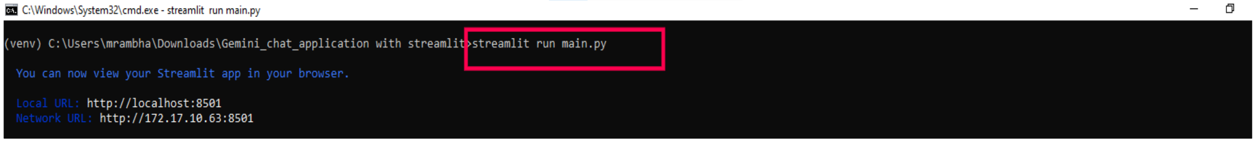
Replace **<Your-Google-API-Key>** with your actual API key that we got in the **Step-2**



```
GOOGLE_API_KEY="[redacted]
```

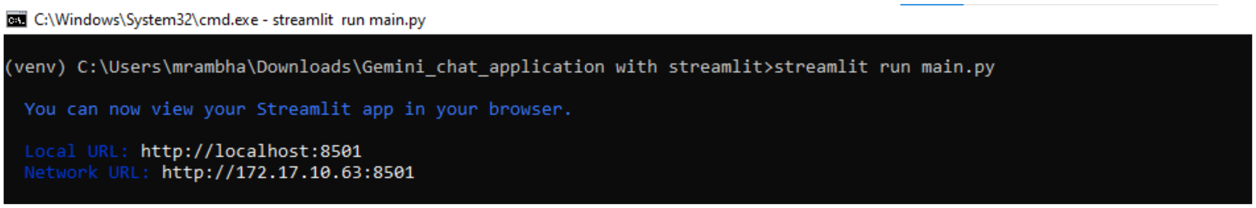
Step #5 | Run the Chatbot Application

Use the below command to start the Streamlit app, **streamlit run main.py**



```
(venv) C:\Users\mrambha\Downloads\Gemini_chat_application with streamlit>streamlit run main.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://172.17.10.63:8501
```

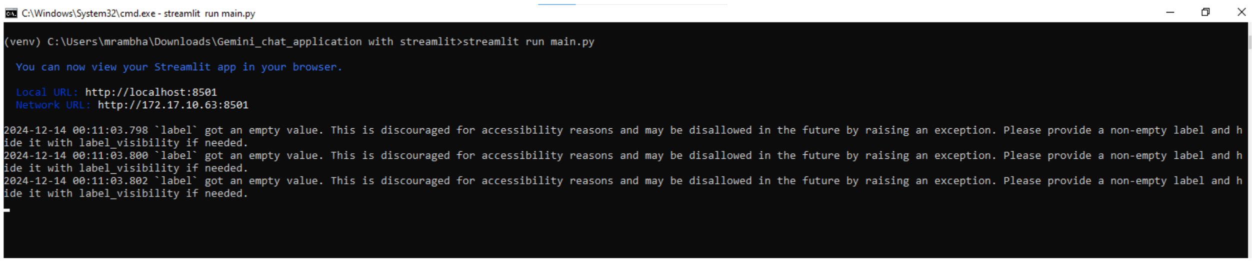
After running the command, your terminal will look like below



```
(venv) C:\Users\mrambha\Downloads\Gemini_chat_application with streamlit>streamlit run main.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://172.17.10.63:8501
```

Step #6 |Test the Chatbot Application

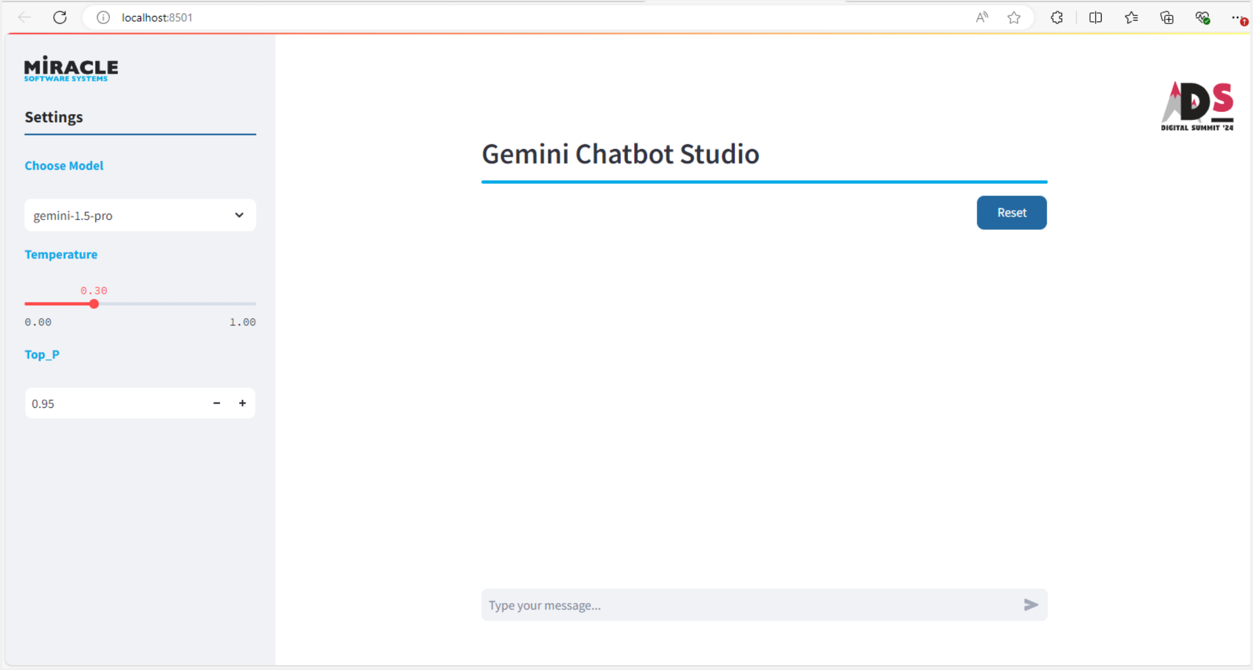
After running the command, Streamlit will provide a URL (**http://localhost:8501**) in the terminal



C:\Windows\System32\cmd.exe - streamlit run main.py

```
(venv) C:\Users\mrambha\Downloads\Gemini_chat_application with streamlit>streamlit run main.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://172.17.10.63:8501

2024-12-14 00:11:03.798 `label` got an empty value. This is discouraged for accessibility reasons and may be disallowed in the future by raising an exception. Please provide a non-empty label and h
ide it with label_visibility if needed.
2024-12-14 00:11:03.800 `label` got an empty value. This is discouraged for accessibility reasons and may be disallowed in the future by raising an exception. Please provide a non-empty label and h
ide it with label_visibility if needed.
2024-12-14 00:11:03.802 `label` got an empty value. This is discouraged for accessibility reasons and may be disallowed in the future by raising an exception. Please provide a non-empty label and h
ide it with label_visibility if needed.
```

Open this URL in your browser to interact with the chatbot application, which will redirect you to the UI.



localhost:8501

Settings

Choose Model: gemini-1.5-pro

Temperature: 0.30 (0.00 – 1.00)

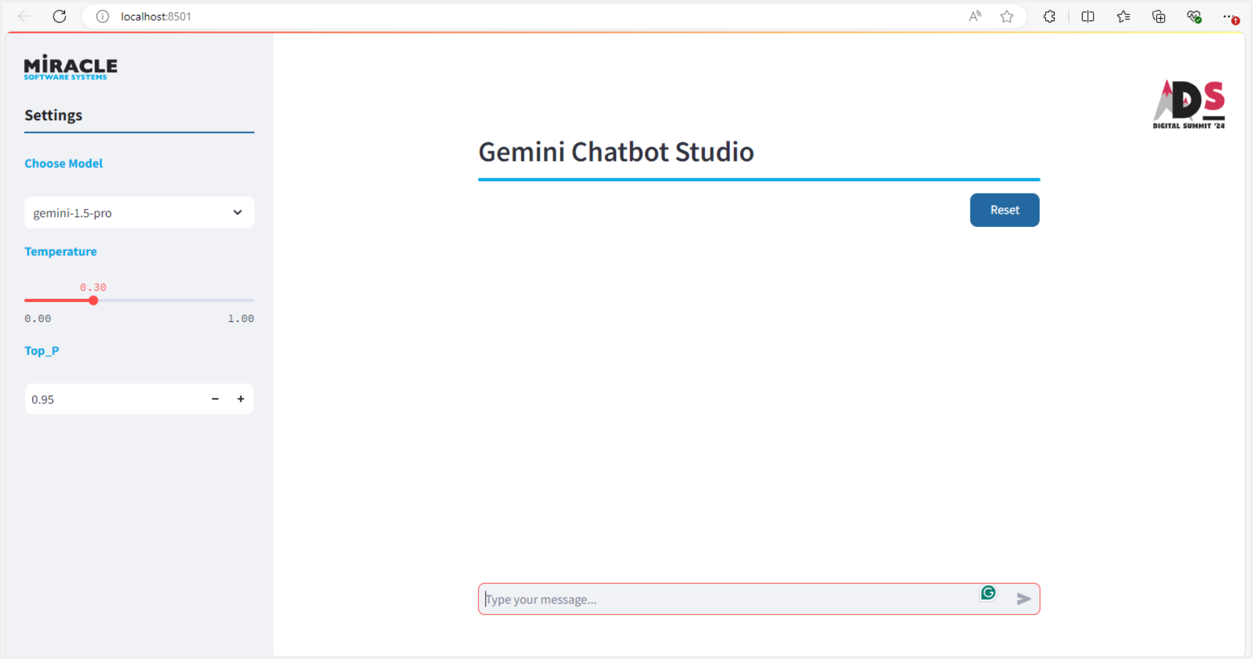
Top_P: 0.95

Gemini Chatbot Studio

Enter a question or prompt in the input box and submit it. Ask a variety of questions from different domains, powered by Google's Gemini Pro.

Example Prompts/ Questions

- What does the temperature setting do in generative AI models?
- Can you help me debug this error: KeyError: 'value'?
- Explain the theory of relativity?
- Can you suggest a workout plan for beginners?



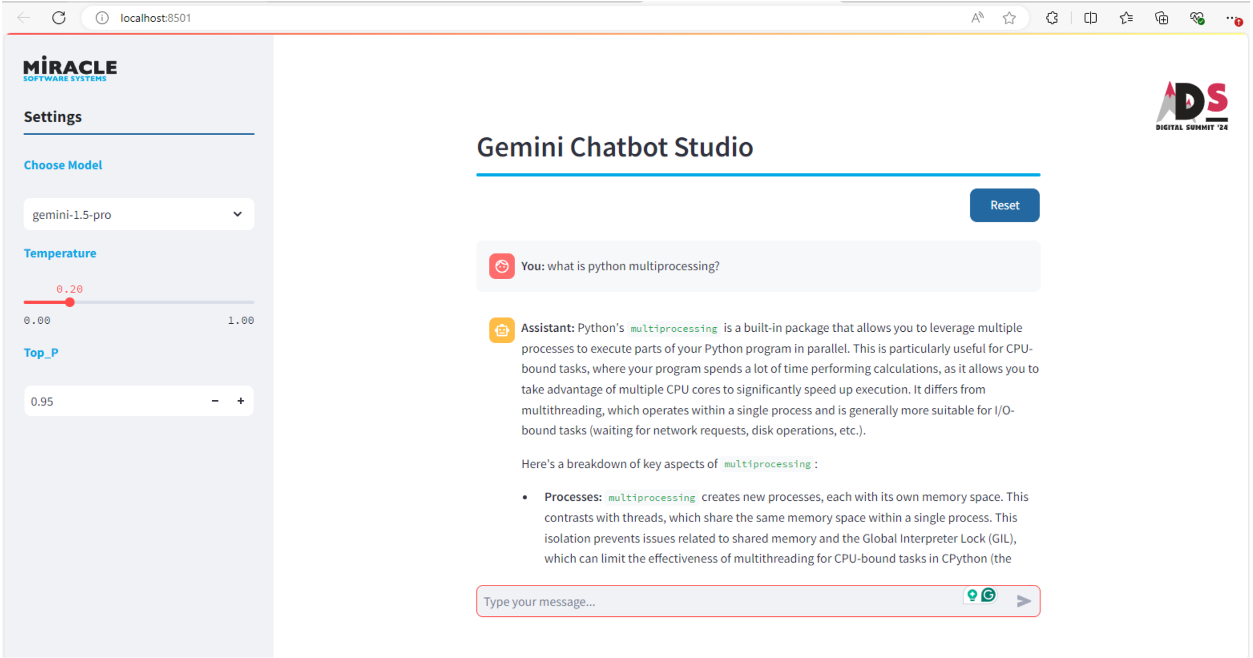
Settings

Choose Model: gemini-1.5-pro

Temperature: 0.30 (0.00 – 1.00)

Top_P: 0.95

Gemini Chatbot Studio



Settings

Choose Model: gemini-1.5-pro

Temperature: 0.20 (0.00 – 1.00)

Top_P: 0.95

Gemini Chatbot Studio

You: what is python multiprocessing?

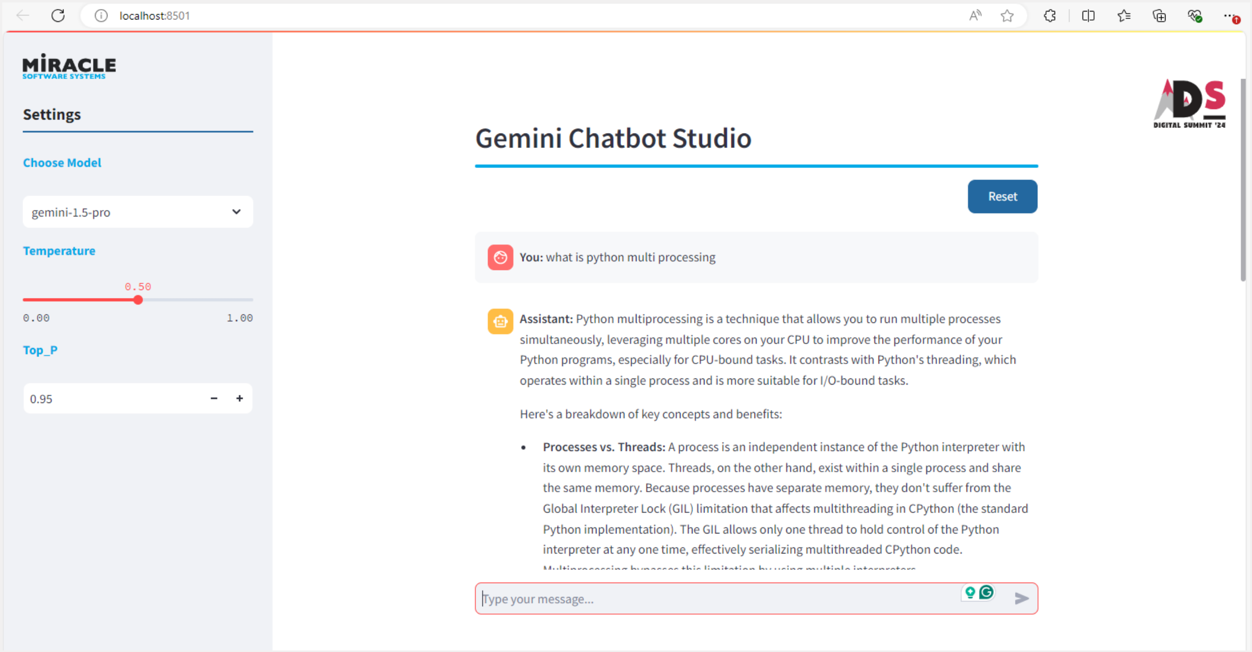
Assistant: Python's `multiprocessing` is a built-in package that allows you to leverage multiple processes to execute parts of your Python program in parallel. This is particularly useful for CPU-bound tasks, where your program spends a lot of time performing calculations, as it allows you to take advantage of multiple CPU cores to significantly speed up execution. It differs from multithreading, which operates within a single process and is generally more suitable for I/O-bound tasks (waiting for network requests, disk operations, etc.).

Here's a breakdown of key aspects of `multiprocessing`:

- **Processes:** `multiprocessing` creates new processes, each with its own memory space. This contrasts with threads, which share the same memory space within a single process. This isolation prevents issues related to shared memory and the Global Interpreter Lock (GIL), which can limit the effectiveness of multithreading for CPU-bound tasks in CPython (the

Temperature - Controls randomness in responses. Lower values (e.g., 0.2) lead to more focused answers, while higher values (e.g., 1.0) encourage more creative responses.

Top_P - Limits choices to the top tokens with a cumulative probability above P. Higher values (e.g., 0.95) increase diversity in responses.



Choose Model

gemini-1.5-pro

Temperature

0.50

0.00 1.00

Top_P

0.95

Gemini Chatbot Studio

You: what is python multi processing

Assistant: Python multiprocessing is a technique that allows you to run multiple processes simultaneously, leveraging multiple cores on your CPU to improve the performance of your Python programs, especially for CPU-bound tasks. It contrasts with Python's threading, which operates within a single process and is more suitable for I/O-bound tasks.

Here's a breakdown of key concepts and benefits:

- **Processes vs. Threads:** A process is an independent instance of the Python interpreter with its own memory space. Threads, on the other hand, exist within a single process and share the same memory. Because processes have separate memory, they don't suffer from the Global Interpreter Lock (GIL) limitation that affects multithreading in CPython (the standard Python implementation). The GIL allows only one thread to hold control of the Python interpreter at any one time, effectively serializing multithreaded CPython code. Multiprocessing bypasses this limitation by using multiple interpreters