```
!pip install faiss-cpu transformers datasets sentence-transformers
```

```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.3/56.3 MB 15.7 MB/s eta 0:00:00
Downloading nvidia_cusolver_cu12-11.6.1.9-py3-none-manylinux2014_x86_64.whl (127.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 127.9/127.9 MB 7.4 MB/s eta 0:00:00
Downloading nvidia_cusparse_cu12-12.3.1.170-py3-none-manylinux2014_x86_64.whl (207.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.5/207.5 MB 6.5 MB/s eta 0:00:00
Downloading nvidia_nvjitlink_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (21.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21.1/21.1 MB 55.7 MB/s eta 0:00:00
Downloading xxhash-3.5.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (194 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 194.8/194.8 kB 15.9 MB/s eta 0:00:00
Installing collected packages: xxhash, nvidia-nvjitlink-cu12, nvidia-curand-cu12, nvidia-cufft-cu12, nvidia-cuda-runtime-cu12, nv
  Attempting uninstall: nvidia-nvjitlink-cu12
    Found existing installation: nvidia-nvjitlink-cu12 12.5.82
    Uninstalling nvidia-nvjitlink-cu12-12.5.82:
      Successfully uninstalled nvidia-nvjitlink-cu12-12.5.82
  Attempting uninstall: nvidia-curand-cu12
    Found existing installation: nvidia-curand-cu12 10.3.6.82
    Uninstalling nvidia-curand-cu12-10.3.6.82:
      Successfully uninstalled nvidia-curand-cu12-10.3.6.82
  Attempting uninstall: nvidia-cufft-cu12
    Found existing installation: nvidia-cufft-cu12 11.2.3.61
    Uninstalling nvidia-cufft-cu12-11.2.3.61:
      Successfully uninstalled nvidia-cufft-cu12-11.2.3.61
  Attempting uninstall: nvidia-cuda-runtime-cu12
    Found existing installation: nvidia-cuda-runtime-cu12 12.5.82
    Uninstalling nvidia-cuda-runtime-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-runtime-cu12-12.5.82
  Attempting uninstall: nvidia-cuda-nvrtc-cu12
    Found existing installation: nvidia-cuda-nvrtc-cu12 12.5.82
    Uninstalling nvidia-cuda-nvrtc-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-nvrtc-cu12-12.5.82
  Attempting uninstall: nvidia-cuda-cupti-cu12
    Found existing installation: nvidia-cuda-cupti-cu12 12.5.82
    Uninstalling nvidia-cuda-cupti-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-cupti-cu12-12.5.82
  Attempting uninstall: nvidia-cublas-cu12
    Found existing installation: nvidia-cublas-cu12 12.5.3.2
    Uninstalling nvidia-cublas-cu12-12.5.3.2:
      Successfully uninstalled nvidia-cublas-cu12-12.5.3.2
  Attempting uninstall: fsspec
    Found existing installation: fsspec 2025.3.0
    Uninstalling fsspec-2025.3.0:
      Successfully uninstalled fsspec-2025.3.0
  Attempting uninstall: nvidia-cusparse-cu12
    Found existing installation: nvidia-cusparse-cu12 12.5.1.3
    Uninstalling nvidia-cusparse-cu12-12.5.1.3:
      Successfully uninstalled nvidia-cusparse-cu12-12.5.1.3
  Attempting uninstall: nvidia-cudnn-cu12
    Found existing installation: nvidia-cudnn-cu12 9.3.0.75
    Uninstalling nvidia-cudnn-cu12-9.3.0.75:
      Successfully uninstalled nvidia-cudnn-cu12-9.3.0.75
  Attempting uninstall: nvidia-cusolver-cu12
    Found existing installation: nvidia-cusolver-cu12 11.6.3.83
    Uninstalling nvidia-cusolver-cu12-11.6.3.83:
      Successfully uninstalled nvidia-cusolver-cu12-11.6.3.83
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the
gcsfs 2025.3.0 requires fsspec==2025.3.0, but you have fsspec 2024.12.0 which is incompatible.
Successfully installed datasets-3.5.0 dill-0.3.8 faiss-cpu-1.10.0 fsspec-2024.12.0 multiprocess-0.70.16 nvidia-cublas-cu12-12.4.5
```

```
import faiss
import numpy as np
from sentence_transformers import SentenceTransformer


# Load a sentence embedding model
embed_model = SentenceTransformer('all-MiniLM-L6-v2')
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as s
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
modules.json: 100%                                                  349/349 [00:00<00:00, 8.02kB/s]
config_sentence_transformers.json: 100%                             116/116 [00:00<00:00, 1.86kB/s]
README.md: 100%                                                     10.5k/10.5k [00:00<00:00, 774kB/s]
sentence_bert_config.json: 100%                                     53.0/53.0 [00:00<00:00, 3.95kB/s]
config.json: 100%                                                   612/612 [00:00<00:00, 62.7kB/s]
model.safetensors: 100%                                             90.9M/90.9M [00:00<00:00, 222MB/s]
tokenizer_config.json: 100%                                         350/350 [00:00<00:00, 34.7kB/s]
vocab.txt: 100%                                                     232k/232k [00:00<00:00, 1.07MB/s]
tokenizer.json: 100%                                                466k/466k [00:00<00:00, 1.07MB/s]
special_tokens_map.json: 100%                                       112/112 [00:00<00:00, 9.93kB/s]
config.json: 100%                                                   190/190 [00:00<00:00, 17.3kB/s]
```

```
# Example startup-related documents
documents = [
    "Startup India provides funding and tax benefits for new startups in India.",
    "Angel investors are individuals who invest in early-stage startups in exchange for equity.",
    "A pitch deck is a presentation that startups use to attract investors.",
    "The government offers startup grants through various schemes.",
    "Networking events connect entrepreneurs with investors and mentors."
]


# Convert documents to vectors
doc_vectors = embed_model.encode(documents, convert_to_numpy=True)


# Initialize FAISS index
dimension = doc_vectors.shape[1]
index = faiss.IndexFlatL2(dimension)
index.add(doc_vectors)


def retrieve_docs(query, top_k=2):
    query_vector = embed_model.encode([query], convert_to_numpy=True)
    distances, indices = index.search(query_vector, top_k)
    return [documents[i] for i in indices[0]]

# Test retrieval
print(retrieve_docs("How can I get funding?"))
```

```
['The government offers startup grants through various schemes.', 'Networking events connect entrepreneurs with investors and mentor
```

```
from transformers import AutoModelForSeq2SeqLM, AutoTokenizer
import torch

# Load FLAN-T5 Small
model_name = "google/flan-t5-small"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSeq2SeqLM.from_pretrained(model_name)

# Generate response
def generate_response(query):
    retrieved_docs = retrieve_docs(query)
    context = " ".join(retrieved_docs)

    prompt = f"Context: {context}\nQuestion: {query}\nAnswer:"
    inputs = tokenizer(prompt, return_tensors="pt", padding=True, truncation=True)
    outputs = model.generate(**inputs, max_new_tokens=100)
    return tokenizer.decode(outputs[0], skip_special_tokens=True)

# Test chatbot
print(generate_response("Tell me about startup funding."))
```

```
tokenizer_config.json: 100%  2.54k/2.54k [00:00<00:00, 224kB/s]
spiece.model: 100%  792k/792k [00:00<00:00, 21.8MB/s]
tokenizer.json: 100%  2.42M/2.42M [00:00<00:00, 2.91MB/s]
special_tokens_map.json: 100%  2.20k/2.20k [00:00<00:00, 148kB/s]
config.json: 100%  1.40k/1.40k [00:00<00:00, 144kB/s]
model.safetensors: 100%  308M/308M [00:01<00:00, 259MB/s]
generation_config.json: 100%  147/147 [00:00<00:00, 14.8kB/s]
Startup India provides funding and tax benefits for new startups in India.
```

```
!git clone https://huggingface.co/spaces/YOUR_USERNAME/startup-chatbot
%cd startup-chatbot
```

```
Cloning into 'startup-chatbot'...
fatal: could not read Username for 'https://huggingface.co': No such device or address
[Errno 2] No such file or directory: 'startup-chatbot'
/content
```

```
%%writefile app.py
from fastapi import FastAPI
from transformers import AutoModelForSeq2SeqLM, AutoTokenizer
import faiss
import torch
from sentence_transformers import SentenceTransformer

app = FastAPI()

# Load models
embed_model = SentenceTransformer('all-MiniLM-L6-v2')
tokenizer = AutoTokenizer.from_pretrained("google/flan-t5-small")
model = AutoModelForSeq2SeqLM.from_pretrained("google/flan-t5-small")

# Sample documents
documents = [
    "Startup India provides funding and tax benefits for new startups in India.",
    "Angel investors are individuals who invest in early-stage startups in exchange for equity.",
    "A pitch deck is a presentation that startups use to attract investors.",
    "The government offers startup grants through various schemes.",
    "Networking events connect entrepreneurs with investors and mentors."
]

# Convert documents to embeddings and store in FAISS
doc_vectors = embed_model.encode(documents, convert_to_numpy=True)
dimension = doc_vectors.shape[1]
index = faiss.IndexFlatL2(dimension)
index.add(doc_vectors)

def retrieve_docs(query, top_k=2):
    query_vector = embed_model.encode([query], convert_to_numpy=True)
    distances, indices = index.search(query_vector, top_k)
    return [documents[i] for i in indices[0]]

def generate_response(query):
    retrieved_docs = retrieve_docs(query)
    context = " ".join(retrieved_docs)
    prompt = f"Context: {context}\nQuestion: {query}\nAnswer:"
    inputs = tokenizer(prompt, return_tensors="pt", padding=True, truncation=True)
    outputs = model.generate(**inputs, max_new_tokens=100)
    return tokenizer.decode(outputs[0], skip_special_tokens=True)

@app.get("/chat")
def chat(query: str):
    return {"response": generate_response(query)}
```

```
Writing app.py
```

```
%%writefile requirements.txt
faiss-cpu
transformers
sentence-transformers
fastapi
uvicorn
torch
```

```
Writing requirements.txt
```

```
%%writefile Dockerfile
# Use official Python image
FROM python:3.10

# Set working directory
WORKDIR /app

# Copy files
COPY . /app

# Install dependencies
RUN pip install --no-cache-dir -r requirements.txt

# Expose API port
EXPOSE 7860

# Start FastAPI server
CMD ["uvicorn", "app:app", "--host", "0.0.0.0", "--port", "7860"]
```

```
Writing Dockerfile
```

```
!git add .
!git commit -m "Deploy startup chatbot API"
!git push
```

```
fatal: not a git repository (or any of the parent directories): .git
fatal: not a git repository (or any of the parent directories): .git
fatal: not a git repository (or any of the parent directories): .git
```

```
!pip install huggingface_hub
from huggingface_hub import notebook_login

notebook_login()
```

```
Requirement already satisfied: huggingface_hub in /usr/local/lib/python3.11/dist-packages (0.29.3)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from huggingface_hub) (3.18.0)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.11/dist-packages (from huggingface_hub) (2024.12.0)
Requirement already satisfied: packaging>=20.9 in /usr/local/lib/python3.11/dist-packages (from huggingface_hub) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages (from huggingface_hub) (6.0.2)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from huggingface_hub) (2.32.3)
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.11/dist-packages (from huggingface_hub) (4.67.1)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.11/dist-packages (from huggingface_hub) (4.12.2
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->huggingface_hub)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->huggingface_hub) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->huggingface_hub) (2.3.0
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->huggingface_hub) (2025
```

```
!git clone https://huggingface.co/spaces/key-life/startup-chatbot
```

```
Cloning into 'startup-chatbot'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 13 (delta 2), reused 0 (delta 0), pack-reused 4 (from 1)
Unpacking objects: 100% (13/13), 5.06 KiB | 1.01 MiB/s, done.
```

```
!git clone https://huggingface.co/spaces/key-life/VentureAI
```

```
Cloning into 'VentureAI'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 9 (delta 1), reused 1 (delta 0), pack-reused 4 (from 1)
Unpacking objects: 100% (9/9), 3.21 KiB | 1.07 MiB/s, done.
```

```
import shutil
import os
```

```
# Define source and destination
source_files = ["/content/main.py", "/content/app.py", "/content/requirements.txt"]
destination_folder = "/content/VentureAI/"

# Move files if they exist, handling existing files
for file in source_files:
    destination_path = os.path.join(destination_folder, os.path.basename(file))
    if os.path.exists(file) and not os.path.exists(destination_path): # Check if dest file exists
        shutil.move(file, destination_folder)
    elif os.path.exists(destination_path):
        print(f"Warning: {destination_path} already exists! Skipping.") # Notify if file exists
    else:
        print(f"Warning: {file} not found!")
```

```
Warning: /content/main.py not found!
Warning: /content/VentureAI/app.py already exists! Skipping.
Warning: /content/VentureAI/requirements.txt already exists! Skipping.
```

```
!ls -R /content
```

```
/content:
app.py  Dockerfile  requirements.txt  sample_data  startup-chatbot  VentureAI

/content/sample_data:
anscombe.json                california_housing_train.csv  mnist_train_small.csv
california_housing_test.csv  mnist_test.csv                README.md

/content/startup-chatbot:
app.py  Dockerfile  README.md  requirements.txt

/content/VentureAI:
app.py  README.md  requirements.txt
```

```
!find /content -name "main.py"
```

```
with open("/content/VentureAI/main.py", "w") as f:
    f.write("# This is the main.py file for VentureAI chatbot\n")
print("main.py has been created!")
```

```
main.py has been created!
```

```
cd /content/VentureAI
```

```
/content/VentureAI
```

```
!git config --global user.email "sejalqp@gmail.com"
!git config --global user.name "key-life"
```

```
!git add .
!git commit -m "Added main.py and updated chatbot files"
!git push
```

```
[main 8e4d88c] Added main.py and updated chatbot files
 1 file changed, 1 insertion(+)
 create mode 100644 main.py
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 326 bytes | 326.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
```