**GOOGLE COLLAB WEBSITE**

**COPY CODE AND PASTE IT ON NOTEBOOK**

from IPython import get\_ipython

from IPython.display import display

!pip install faiss-gpu

!pip install faiss-cpu

# %%

!pip install pypdf

from IPython import get\_ipython

from IPython.display import display

# %%

!pip install langchain-huggingface # Install the missing package 'langchain-huggingface'

!pip install langchain\_community

from google.colab import files

import requests

import os

from langchain\_community.document\_loaders import PyPDFLoader

from langchain.text\_splitter import CharacterTextSplitter

from langchain\_huggingface import HuggingFaceEmbeddings # Now this import should work

from langchain\_community.vectorstores import FAISS

import pypdf # Import the pypdf library

# Step 1: Set HuggingFace Token

from google.colab import userdata

os.environ["HUGGINGFACEHUB\_API\_TOKEN"] = userdata.get('HUGGINGFACEHUB\_API\_TOKEN')

# Step 2: Upload Document

print("Please upload your document (PDF):")

uploaded = files.upload()

pdf\_path = next(iter(uploaded.keys())) # Get the uploaded file name

# Step 3: Load and Process Document

loader = PyPDFLoader(pdf\_path)

# Use strict=False to disable strict parsing rules

# This might help with minor inconsistencies in the PDF file.

pages = loader.load\_and\_split(text\_splitter=CharacterTextSplitter(chunk\_size=1000, chunk\_overlap=200))

# Step 4: Create Embeddings and FAISS Vector Store

embeddings = HuggingFaceEmbeddings(

model\_name="sentence-transformers/all-MiniLM-L6-v2"

)

vector\_db = FAISS.from\_documents(pages, embeddings)

# Step 5: Set Up Groq API

GROQ\_API\_KEY = userdata.get('GROQ\_API\_KEY') # Access the secret

def groq\_llm(query, context):

"""Send query and context to Groq API"""

headers = {

"Authorization": f"Bearer {GROQ\_API\_KEY}",

"Content-Type": "application/json"

}

prompt = f"Context: {context}\n\nQuestion: {query}\nAnswer:"

data = {

"model": "llama3-70b-8192",

"messages": [{

"role": "user",

"content": prompt

}],

"temperature": 0.7

}

try:

response = requests.post(

"https://api.groq.com/openai/v1/chat/completions",

headers=headers,

json=data

)

response.raise\_for\_status() # Raise an error for bad status codes

response\_data = response.json()

# Debug: Print the full API response

print("API Response:", response\_data)

return response\_data['choices'][0]['message']['content']

except requests.exceptions.RequestException as e:

print(f"API Request Failed: {e}")

return "Sorry, I couldn't process your request. Please try again."

except KeyError as e:

print(f"Invalid API Response Format: {e}")

return "Sorry, there was an issue with the API response."

# Step 6: RAG Query Function

def rag\_query(query):

# Retrieve relevant context

docs = vector\_db.similarity\_search(query, k=3)

context = "\n".join([d.page\_content for d in docs])

# Get LLM response

return groq\_llm(query, context)

# Step 7: Interactive Prompt Loop

print("\nDocument processing complete! You can now ask questions about the document.")

while True:

user\_prompt = input("\nEnter your question (or type 'exit' to quit): ")

if user\_prompt.lower() == "exit":

print("Goodbye!")

break

response = rag\_query(user\_prompt)

print(f"\nAnswer: {response}")

**Step 1: Set Up Google Colab**

1. **Open Google Colab**:
   * Go to [Google Colab](https://colab.research.google.com/).
   * Click on New Notebook to create a new Python notebook.
2. **Copy the Code**:
   * Copy the entire code provided above.
   * Paste it into the first cell of your Google Colab notebook.
3. **Set Runtime to GPU**:
   * Click on Runtime > Change runtime type.
   * Under Hardware accelerator, select **GPU** and save.

**Step 2: Get Hugging Face API Key**

1. **Create a Hugging Face Account**:
   * Go to [Hugging Face](https://huggingface.co/).
   * Sign up for a free account if you don’t have one.
2. **Generate an API Token**:
   * After logging in, click on your profile picture in the top-right corner.
   * Go to Settings > Access Tokens.
   * Click on New Token to generate a new API token.
   * Copy the token (you’ll need it in the next step).
3. **Add the Token to Google Colab**:
   * In Google Colab, click on the **key icon** (🔑) on the left sidebar (this is the "Secrets" section).
   * Click on Add new secret.
   * Enter the name as HUGGINGFACEHUB\_API\_TOKEN and paste the token you copied earlier.
   * Click Save.

**Step 3: Get Groq API Key**

1. **Create a Groq Cloud Account**:
   * Go to [Groq Cloud](https://console.groq.com/" \t "_blank).
   * Sign up for a free account.
2. **Generate an API Key**:
   * After logging in, go to the API Keys section.
   * Click on Create API Key.
   * Copy the API key.
3. **Add the Key to Google Colab**:
   * In Google Colab, click on the **key icon** (🔑) again.
   * Click on Add new secret.
   * Enter the name as GROQ\_API\_KEY and paste the Groq API key.
   * Click Save.

**Step 4: Run the Code**

1. **Run the First Cell**:
   * Click the ▶️ button to run the first cell.
   * This will install all the required libraries (faiss, pypdf, langchain, etc.).
2. **Upload a PDF Document**:
   * After running the code, you’ll see a prompt: Please upload your document (PDF):.
   * Click on Choose File and upload a PDF file from your computer.
3. **Wait for Processing**:
   * The code will process the PDF, split it into chunks, and create embeddings using Hugging Face’s all-MiniLM-L6-v2 model.
4. **Ask Questions**:
   * Once the processing is complete, you’ll see the message: Document processing complete! You can now ask questions about the document.
   * Type your question (e.g., "What is the main topic of this document?") and press Enter.
   * The code will use the **Groq API** (with Llama3-70b) to generate an answer based on the PDF content.
5. **Exit the Program**:
   * To stop the program, type exit and press Enter.

**Step 5: Debugging and Troubleshooting**

1. **Check API Responses**:
   * If the Groq API fails, the code will print the error message. Check if the API key is correct and if you have sufficient credits.
2. **PDF Issues**:
   * If the PDF is scanned or contains images, the code may not work properly. Use OCR tools to extract text from such PDFs.
3. **Library Errors**:
   * If any library fails to install, manually install it using !pip install <library\_name>.

**Example Workflow**

1. **Upload a PDF**: For example, upload a research paper or a book chapter.
2. **Ask Questions**:
   * Question: "What is the main topic of this document?"
   * Answer: The AI will analyze the PDF and provide a summary or answer.
3. **Exit**: Type exit to stop the program.