```
!pip install -q pandas faiss-cpu sentence-transformers google-generativeai gradio
```

```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 31.3/31.3 MB 29.6 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 363.4/363.4 MB 4.1 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.8/13.8 MB 26.0 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 24.6/24.6 MB 34.7 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 883.7/883.7 kB 50.8 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 664.8/664.8 MB 1.3 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 211.5/211.5 MB 5.7 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.3/56.3 MB 9.3 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 127.9/127.9 MB 9.9 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.5/207.5 MB 5.2 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21.1/21.1 MB 37.0 MB/s eta 0:00:00
```

```
from google.colab import files
uploaded = files.upload()
```

Choose Files Training Dataset.csv

- **Training Dataset.csv**(text/csv) - 38011 bytes, last modified: 7/28/2025 - 100% done

```
Saving Training Dataset.csv to Training Dataset.csv
```

```
import pandas as pd

df = pd.read_csv("Training Dataset.csv")
print("Dataset Shape:", df.shape)
df.head()
```

```
Dataset Shape: (614, 13)
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	360.0	1.0	Urban	Y
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	1.0	Rural	N
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	1.0	Urban	Y
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0	Urban	Y

Next steps: Generate code with df   View recommended plots   New interactive sheet

```
documents = df.astype(str).apply(lambda row: ' | '.jo
```

What can I help you build?

```
from sentence_transformers import SentenceTransformer
import numpy as np

model = SentenceTransformer('all-MiniLM-L6-v2')  #
doc_embeddings = model.encode(documents, convert_to_tensor=False)
```

```
/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 secret in your Google Colab and restart your
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, 22.2kB/s]
config_sentence_transformers.json: 100%                                      116/116 [00:00<00:00, 8.22kB/s]
README.md:      10.5k/? [00:00<00:00, 502kB/s]
sentence_bert_config.json: 100%                                         53.0/53.0 [00:00<00:00, 3.56kB/s]
config.json: 100%                                               612/612 [00:00<00:00, 55.4kB/s]
model.safetensors: 100%                                           90.9M/90.9M [00:03<00:00, 32.4MB/s]
tokenizer_config.json: 100%                                         350/350 [00:00<00:00, 32.0kB/s]
vocab.txt:      232k/? [00:00<00:00, 1.74MB/s]
tokenizer.json:      466k/? [00:00<00:00, 5.32MB/s]
special_tokens_map.json: 100%                                         112/112 [00:00<00:00, 10.4kB/s]
config.json: 100%                                               190/190 [00:00<00:00, 10.6kB/s]
```

```
import faiss

dimension = doc_embeddings[0].shape[0]
index = faiss.IndexFlatL2(dimension)
index.add(np.array(doc_embeddings))

def retrieve_top_k(query, k=20):
    query_embedding = model.encode([query])[0]
    D, I = index.search(np.array([query_embedding]), k)
    return [documents[i] for i in I[0]]
```

```
import google.generativeai as genai

genai.configure(api_key="AIzaSyDPg3h1Y6qSF0iOu4e8NSNSDaGj5l9RjV8")
```

```
def generate_response(query, context_docs):
    prompt = f"""You are a helpful assistant for loan approval analysis.

Context:
{chr(10).join(context_docs)}

Question: {query}
Answer:"""

    model = genai.GenerativeModel("gemini-1.5-flash")
    response = model.generate_content(prompt)
    return response.text



def rag_chatbot(query):
    context = retrieve_top_k(query)
    return generate_response(query, context)



print(rag_chatbot("How does credit history affect loan approval?"))
```

```
Based on the provided data, a definitive conclusion on how credit history affects loan approval is difficult to draw due to the small sample size and missing data. However, we

* **Mixed Results:**  The data shows examples of both approved (Y) and not approved (N) loans for individuals with and without credit history.  There's no clear pattern showin

* **Other Factors:** Loan approval seems to depend on multiple factors beyond credit history, such as applicant income (`ApplicantIncome`), co-applicant income (`CoapplicantIn

* **Missing Data:** The presence of `nan` values for some fields (e.g., `LoanAmount`, `Gender`) further complicates the analysis and limits the ability to draw robust conclusi

**To better understand the impact of credit history, a larger and more complete dataset is needed.** A statistical analysis, potentially including regression modeling, could t
```

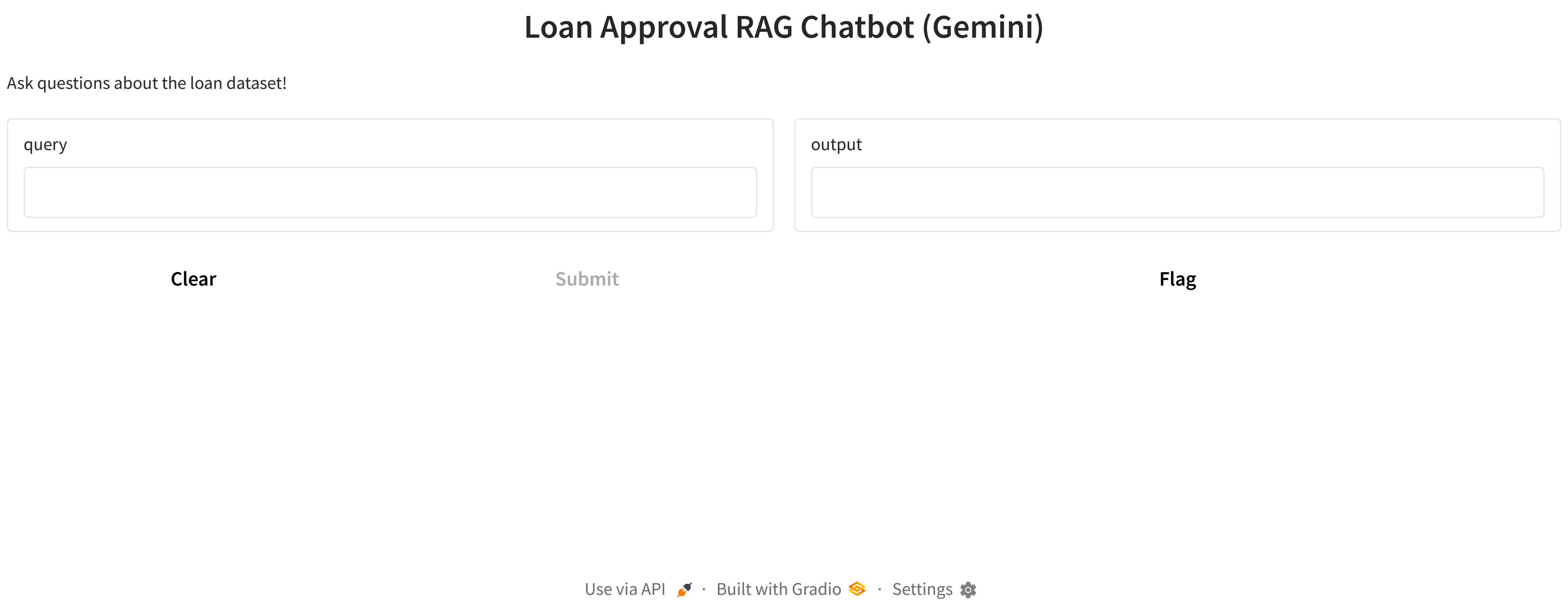
```
import gradio as gr

gr.Interface(fn=rag_chatbot,
             inputs="text",
             outputs="text",
             title="Loan Approval RAG Chatbot (Gemini)",
             description="Ask questions about the loan dataset!"
            ).launch()
```

```
It looks like you are running Gradio on a hosted Jupyter notebook, which requires `share=True`. Automatically setting `share=True` (you can turn this off by setting `share=Fal

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://ecd9c619427dd48c3d.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working directory to deploy to Hugging Face Spaces
```



Loan Approval RAG Chatbot (Gemini)

Ask questions about the loan dataset!