```
!pip install requests beautifulsoup4 langchain faiss-cpu transformers sentence-transformers langchain-community
```

```
Collecting dataclasses-json<0.7,>=0.5.7 (from langchain-community)
  Downloading dataclasses_json-0.6.7-py3-none-any.whl.metadata (25 kB)
Collecting httpx-sse<0.5.0,>=0.4.0 (from langchain-community)
  Downloading httpx_sse-0.4.0-py3-none-any.whl.metadata (9.0 kB)
Collecting pydantic-settings<3.0.0,>=2.4.0 (from langchain-community)
  Downloading pydantic_settings-2.7.1-py3-none-any.whl.metadata (3.5 kB)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchai
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (1.3
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (24.3.0
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (1.
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (6
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (0.2
Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (1.
Collecting marshmallow<4.0.0,>=3.18.0 (from dataclasses-json<0.7,>=0.5.7->langchain-community)
  Downloading marshmallow-3.25.1-py3-none-any.whl.metadata (7.3 kB)
Collecting typing-inspect<1,>=0.4.0 (from dataclasses-json<0.7,>=0.5.7->langchain-community)
  Downloading typing_inspect-0.9.0-py3-none-any.whl.metadata (1.5 kB)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.24.0->transfor
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.24.0
Requirement already satisfied: jsonpatch<2.0,>=1.33 in /usr/local/lib/python3.10/dist-packages (from langchain-core<0.4.0,>=0.3.29->lang
Requirement already satisfied: httpx<1,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from langsmith<0.3,>=0.1.17->langchain) (0.
Requirement already satisfied: orjson<4.0.0,>=3.9.14 in /usr/local/lib/python3.10/dist-packages (from langsmith<0.3,>=0.1.17->langchain
Requirement already satisfied: requests-toolbelt<2.0.0,>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from langsmith<0.3,>=0.1.17-
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.10/dist-packages (from pydantic<3.0.0,>=2.7.4->langchai
Requirement already satisfied: pydantic-core==2.27.2 in /usr/local/lib/python3.10/dist-packages (from pydantic<3.0.0,>=2.7.4->langchain
Collecting python-dotenv>=0.21.0 (from pydantic-settings<3.0.0,>=2.4.0->langchain-community)
  Downloading python_dotenv-1.0.1-py3-none-any.whl.metadata (23 kB)
Requirement already satisfied: greenlet!=0.4.17 in /usr/local/lib/python3.10/dist-packages (from SQLAlchemy<3,>=1.4->langchain) (3.1.1)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->sentence-transformers) (3.4.2)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->sentence-transformers) (3.1.5)
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.10/dist-packages (from torch>=1.11.0->sentence-transformers) (1.
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy==1.13.1->torch>=1.11.0->senten
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn->sentence-transformers) (1.4
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn->sentence-transformer
Requirement already satisfied: anyio in /usr/local/lib/python3.10/dist-packages (from httpx<1,>=0.23.0->langsmith<0.3,>=0.1.17->langcha
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.10/dist-packages (from httpx<1,>=0.23.0->langsmith<0.3,>=0.1.17-
Requirement already satisfied: h11<0.15,>=0.13 in /usr/local/lib/python3.10/dist-packages (from httpcore==1.*->httpx<1,>=0.23.0->langsm
Requirement already satisfied: jsonpointer>=1.9 in /usr/local/lib/python3.10/dist-packages (from jsonpatch<2.0,>=1.33->langchain-core<0
Collecting mypy-extensions>=0.3.0 (from typing-inspect<1,>=0.4.0->dataclasses-json<0.7,>=0.5.7->langchain-community)
  Downloading mypy_extensions-1.0.0-py3-none-any.whl.metadata (1.1 kB)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.11.0->sentence-transfo
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.10/dist-packages (from anyio->httpx<1,>=0.23.0->langsmith<0.3,>=0
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio->httpx<1,>=0.23.0->langsmith<0.3,>
Downloading faiss_cpu-1.9.0.post1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (27.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 27.5/27.5 MB 70.2 MB/s eta 0:00:00
Downloading langchain_community-0.3.14-py3-none-any.whl (2.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.5/2.5 MB 97.9 MB/s eta 0:00:00
Downloading dataclasses_json-0.6.7-py3-none-any.whl (28 kB)
Downloading httpx_sse-0.4.0-py3-none-any.whl (7.8 kB)
Downloading pydantic_settings-2.7.1-py3-none-any.whl (29 kB)
Downloading marshmallow-3.25.1-py3-none-any.whl (49 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 49.6/49.6 kB 5.0 MB/s eta 0:00:00
Downloading python_dotenv-1.0.1-py3-none-any.whl (19 kB)
Downloading typing_inspect-0.9.0-py3-none-any.whl (8.8 kB)
Downloading mypy_extensions-1.0.0-py3-none-any.whl (4.7 kB)
Installing collected packages: python-dotenv, mypy-extensions, marshmallow, httpx-sse, faiss-cpu, typing-inspect, pydantic-settings, dat
Successfully installed dataclasses-json-0.6.7 faiss-cpu-1.9.0.post1 httpx-sse-0.4.0 langchain-community-0.3.14 marshmallow-3.25.1 mypy-e
```

```
import requests
from bs4 import BeautifulSoup
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.vectorstores import FAISS
from langchain.embeddings import HuggingFaceEmbeddings
from langchain.chains import RetrievalQA
import torch
from transformers import AutoModelForCausalLM, AutoTokenizer
from huggingface_hub import login


class ModelInference:
    def __init__(self, model_name="shriasannuthi/gemma-2b-fargo", device="cuda"):
        self.model_name = model_name
        self.device = device if torch.cuda.is_available() else "cpu"
        self.model = self._load_model()
        self.tokenizer = self._load_tokenizer()

    def _load_model(self):
        """Load the pre-trained GPT model from Hugging Face."""
        print("Loading model...")
        model = AutoModelForCausalLM.from_pretrained(
            self.model_name,
            torch_dtype=torch.float16
        )
        return model.to(self.device)

    def _load_tokenizer(self):
```

```
        """Load the tokenizer associated with the model."""
        print("Loading tokenizer...")
        return AutoTokenizer.from_pretrained(self.model_name)

    def generate_response(self, prompt, max_new_tokens=100):
        """Generate a response from the model based on the prompt."""
        print("Generating response...")
        inputs = self.tokenizer(prompt, return_tensors="pt").to(self.device)
        outputs = self.model.generate(
            **inputs,
            max_new_tokens=max_new_tokens
        )
        return self.tokenizer.decode(outputs[0], skip_special_tokens=True)


def scrape_website(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')
    paragraphs = soup.find_all('p')
    text = "\n".join([para.get_text() for para in paragraphs])
    return text


def split_text_into_chunks(text, max_chunks=100):
    text_splitter = RecursiveCharacterTextSplitter(
        chunk_size=300,
        chunk_overlap=50,
        separators=['\n', ' ', '']
    )
    chunks = text_splitter.split_text(text)
    return chunks[:max_chunks]


embedding_model = "sentence-transformers/all-MiniLM-L6-v2"
embeddings = HuggingFaceEmbeddings(model_name=embedding_model)
```

```
<ipython-input-6-cbb286c71437>:2: LangChainDeprecationWarning: The class `HuggingFaceEmbeddings` was deprecated in LangChain 0.2.2 and wil
  embeddings = HuggingFaceEmbeddings(model_name=embedding_model)
/usr/local/lib/python3.10/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
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, 6.57kB/s]
config_sentence_transformers.json: 100%                             116/116 [00:00<00:00, 3.25kB/s]
README.md: 100%                                                     10.7k/10.7k [00:00<00:00, 303kB/s]
sentence_bert_config.json: 100%                                     53.0/53.0 [00:00<00:00, 948B/s]
config.json: 100%                                                   612/612 [00:00<00:00, 9.48kB/s]
model.safetensors: 100%                                             90.9M/90.9M [00:00<00:00, 118MB/s]
tokenizer_config.json: 100%                                         350/350 [00:00<00:00, 7.93kB/s]
vocab.txt: 100%                                                     232k/232k [00:00<00:00, 3.44MB/s]
tokenizer.json: 100%                                                466k/466k [00:00<00:00, 3.41MB/s]
special_tokens_map.json: 100%                                       112/112 [00:00<00:00, 5.56kB/s]
1_Pooling/config.json: 100%                                         190/190 [00:00<00:00, 3.99kB/s]
```

```
def create_faiss_index(chunks):
    return FAISS.from_texts(chunks, embeddings)


from langchain.llms.base import LLM
from langchain.prompts import PromptTemplate
from langchain.chains import LLMChain
from langchain.chains.combine_documents.stuff import StuffDocumentsChain

def setup_rag_system(index, model_inference):
    retriever = index.as_retriever()

    class CustomLLM(LLM):
        inference_engine: object

        def __init__(self, inference_engine):
            super().__init__(inference_engine=inference_engine)
            self.inference_engine = inference_engine

        def _call(self, prompt: str, stop: list = None) -> str:
            return self.inference_engine.generate_response(prompt)
```

```
    @property
    def _identifying_params(self):
        return {"model_name": self.inference_engine.model_name}

    @property
    def _llm_type(self):
        return "custom_llm"

custom_llm = CustomLLM(inference_engine=model_inference)

prompt = PromptTemplate(
    template="{context}\n\nQuestion: {question}\nAnswer:",
    input_variables=["context", "question"]
)
llm_chain = LLMChain(llm=custom_llm, prompt=prompt)

combine_documents_chain = StuffDocumentsChain(
    llm_chain=llm_chain,
    document_variable_name="context"
)
rag_system = RetrievalQA(
    retriever=retriever,
    combine_documents_chain=combine_documents_chain
)

return rag_system
```

```
from IPython.display import display
import ipywidgets as widgets

hf_token_input = widgets.Password(description='HF Token:', placeholder='Enter your Hugging Face token')
token_submit_button = widgets.Button(description='Login')
token_output_area = widgets.Output()

display(hf_token_input, token_submit_button, token_output_area)

def on_token_submit_clicked(b):
    with token_output_area:
        token_output_area.clear_output()
        hf_token = hf_token_input.value
        if not hf_token:
            print("Please provide a valid Hugging Face token.")
            return
        try:
            login(token=hf_token)
            print("Logged in to Hugging Face successfully!")
        except Exception as e:
            print(f"Error logging in to Hugging Face: {e}")
            return

token_submit_button.on_click(on_token_submit_clicked)
url_input = widgets.Text(description='URL:', placeholder='Enter website URL')
question_input = widgets.Text(description='Question:', placeholder='Enter your question')
submit_button = widgets.Button(description='Submit')
output_area = widgets.Output()

display(url_input, question_input, submit_button, output_area)
```

```
def on_submit_button_clicked(b):
    with output_area:
        output_area.clear_output()
        url = url_input.value
        question = question_input.value

        if not url or not question:
            print("Please provide both a URL and a question.")
            return

        print("Scraping website...")
        scraped_text = scrape_website(url)

        print("Splitting text into chunks...")
        chunks = split_text_into_chunks(scraped_text)

        print("Creating FAISS index...")
        faiss_index = create_faiss_index(chunks)

        print("Setting up RAG system...")
        model_inference = ModelInference()
        rag_system = setup_rag_system(faiss_index, model_inference)

        print("Answering your question...")
        try:
            answer = rag_system.run({"query": question})
            print(f"Context: {answer}")
```

```
        except Exception as e:
            print(f"Error during RAG processing: {e}")

submit_button.on_click(on_submit_button_clicked)
```

HF Token: Enter your Hugging Face token

Login

```
Logged in to Hugging Face successfully!
```

URL: https://creditcards.wellsfargo.com/

Question: What is the Intro Offer on Active C

Submit

```
Scraping website...
Splitting text into chunks...
Creating FAISS index...
Setting up RAG system...
Loading model...
```

config.json: 100% 691/691 [00:00<00:00, 37.2kB/s]

model.safetensors.index.json: 100% 13.5k/13.5k [00:00<00:00, 752kB/s]

Downloading shards: 100% 2/2 [02:00<00:00, 50.02s/it]

model-00001-of-00002.safetensors: 100% 4.95G/4.95G [01:58<00:00, 45.0MB/s]

model-00002-of-00002.safetensors: 100% 67.1M/67.1M [00:01<00:00, 42.3MB/s]

```
`config.hidden_act` is ignored, you should use `config.hidden_activation` instead.
Gemma's activation function will be set to `gelu_pytorch_tanh`. Please, use
`config.hidden_activation` if you want to override this behaviour.
See https://github.com/huggingface/transformers/pull/29402 for more details.
```

Loading checkpoint shards: 100% 2/2 [00:00<00:00, 2.10it/s]

generation_config.json: 100% 132/132 [00:00<00:00, 8.63kB/s]

```
Loading tokenizer...
```

tokenizer_config.json: 100% 40.6k/40.6k [00:00<00:00, 1.75MB/s]

tokenizer.model: 100% 4.24M/4.24M [00:00<00:00, 35.9MB/s]

tokenizer.json: 100% 34.4M/34.4M [00:00<00:00, 42.7MB/s]

special_tokens_map.json: 100% 522/522 [00:00<00:00, 34.6kB/s]

```
<ipython-input-8-abfb4607f052>:33: LangChainDeprecationWarning: The class `LLMChain` was deprecated in LangChain 0.1.17 and will be
removed in 1.0. Use :meth:`~RunnableSequence, e.g., `prompt | llm`` instead.
  llm_chain = LLMChain(llm=custom_llm, prompt=prompt)
<ipython-input-8-abfb4607f052>:35: LangChainDeprecationWarning: This class is deprecated. Use the `create_stuff_documents_chain`
constructor instead. See migration guide here: https://python.langchain.com/docs/versions/migrating_chains/stuff_docs_chain/
  combine_documents_chain = StuffDocumentsChain(
<ipython-input-8-abfb4607f052>:39: LangChainDeprecationWarning: This class is deprecated. Use the `create_retrieval_chain` constructor
instead. See migration guide here: https://python.langchain.com/docs/versions/migrating_chains/retrieval_qa/
  rag_system = RetrievalQA(
<ipython-input-9-dc7aecdc987f>:57: LangChainDeprecationWarning: The method `Chain.run` was deprecated in langchain 0.1.0 and will be
removed in 1.0. Use :meth:`~invoke` instead.
  answer = rag_system.run({"query": question})
Answering your question...
Generating response...
Context: Meet the Active Cash® Card
Earn a $200 cash rewards bonus when you spend $500 in purchases in the first 3 months2
Plus, earn unlimited 2% cash rewards on purchases1
Learn more

    Deliberately simple.

    available for this offer. Refer to the Summary of the Wells Fargo Rewards® Program Terms and Conditions and the Wells Fargo Active Cash
    Visa® Card Addendum for more information about the rewards program. ←back to content

    within 1 – 2 billing periods after they are earned. Cash advances and balance transfers do not apply for purposes of this offer and may
    affect the credit line available for this offer. ATM charges, cash advances, traveler’s checks, money orders, pre-paid gift cards, balance
    transfers, SUPERCHECKS™,

    within 1 – 2 billing periods after they are earned. “Purchases” that do not apply to this offer and do not earn cash rewards include: cash
    advances and equivalents of any kind (ATM transactions, cash advances, traveler’s checks, money orders, pre-paid gift cards, peer-to-peer
    payments, and wire

    Question: What is the Intro Offer on Active Cash Card?
    Answer: The Intro Offer is a $200 cash rewards bonus when you spend $500 in purchases in the first 3 months. Plus, earn unlimited 2% cash
    rewards on purchases.
```

```
!jupyter nbconvert --to pdf /content/drive/MyDrive/Colab_Notebooks/Fargo_RAG.ipynb
```

```
[NbConvertApp] Converting notebook /content/drive/MyDrive/Colab_Notebooks/Fargo_RAG.ipynb to pdf
[NbConvertApp] Writing 62480 bytes to notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', 'notebook.tex', '-quiet']
[NbConvertApp] Running bibtex 1 time: ['bibtex', 'notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 54638 bytes to /content/drive/MyDrive/Colab_Notebooks/Fargo_RAG.pdf
```