

## rag.py

```
1 import streamlit as st
2 from langchain_community.vectorstores import Chroma
3 from langchain_core.messages import HumanMessage, AIMessage, SystemMessage
4 from langchain_core.prompts import ChatPromptTemplate, SystemMessagePromptTemplate,
  HumanMessagePromptTemplate
5 from langchain_core.runnables import RunnablePassthrough
6 from langchain_core.output_parsers import StrOutputParser
7 from langchain_community.document_loaders import WebBaseLoader
8 from langchain_text_splitters import NLTKTextSplitter
9 from langchain_google_genai import GoogleGenerativeAIEmbeddings, ChatGoogleGenerativeAI
10 from langchain_core.runnables import RunnablePassthrough
11 from IPython.display import Markdown as md
12
13 db_connection = Chroma(persist_directory="./chroma_db_", embedding_function=
  embedding_model)
14
15
16 def format_docs(docs):
17     return "\n\n".join(doc.page_content for doc in docs)
18
19 # Converting CHROMA db_connection to Retriever Object
20 retriever = db_connection.as_retriever(search_kwargs={"k": 5})
21
22 # Define chat prompt template
23 chat_template = ChatPromptTemplate.from_messages([
24     SystemMessage(content="""You are a Helpful AI Bot.
25     Your task is to provide assistance based on the context given by the user.
26     Make sure your answers are relevant and helpful."""),
27     HumanMessagePromptTemplate.from_template("Answer the question based on the given
  context.\nContext:\n{context}\nQuestion:\n{question}\nAnswer:")
28 ])
29
30 # Initialize chat model
31 chat_model = ChatGoogleGenerativeAI(google_api_key="
  AIzaSyC2Bztff9XtDCDrCJfMJ8py9JaT8VkwS1Y",
32                                     model="gemini-1.5-pro-latest")
33
34 # Initialize output parser
35 output_parser = StrOutputParser()
36
37 # Define RAG chain
38 rag_chain = (
39     {"context": retriever | format_docs, "question": RunnablePassthrough()}
40     | chat_template
41     | chat_model
42     | output_parser
43 )
44
45 # Streamlit UI
46 st.title("Rag System On Leave")
47 question = st.text_input("Ask your Qurey")
48
49 if st.button("Generate Answer"):
50     if question:
51         response = rag_chain.invoke(query)
52         st.write("answer")
53         st.write(response)
54     else:
```

```
55 | st.warning("enter a query.")
```