```
!pip install langchain_community langchainhub chromadb langchain langchain-openai
```

Show hidden output

```
from google.colab import userdata
import os
os.environ['OPENAI_API_KEY'] = userdata.get('openAIYtKey')
```

```
from langchain_community.document_loaders import WebBaseLoader

loader = WebBaseLoader(web_paths=["https://www.educosys.com/course/genai"])

docs = loader.load()
print(docs)
```

```
WARNING:langchain_community.utils.user_agent:USER_AGENT environment variable not set, consider setting it to identify your requests.
[Document(metadata={'source': 'https://www.educosys.com/course/genai', 'title': 'Hands-on Generative AI Course', 'description': 'Hands-on Generative AI Course', 'language
```

```
from langchain.text_splitter import RecursiveCharacterTextSplitter

text_splitter = RecursiveCharacterTextSplitter(chunk_size = 1000, chunk_overlap = 200)
splits = text_splitter.split_documents(docs)
```

```
print(splits[0])
print(splits[1])
print(splits[2])
```

```
page_content='Hands-on Generative AI CourseCoursesBundle CoursesMentorFree ContentTestimonialsFAQLogin Signup Hands-on Generative AI CourseLearn, Build, Deploy and Apply
page_content='(VAEs) Probabilistic Data Generation Using VAEs Four Mini Projects using TensorFlow Metrics Visualization using TensorBoard Mini Project - Implement a GAN t
page_content='Nodes, State, StateGraph, Workflows AI Agents Mini Project - Simple Q&A Application Using LangChain5Week 5Vector Databases, RAG Vector Databases ChromaDB Ap
```

```
print(len(splits))
```

```
11
```

```
from langchain.embeddings.openai import OpenAIEmbeddings
from langchain.vectorstores import Chroma

vectorstore = Chroma.from_documents(documents=splits, embedding=OpenAIEmbeddings())
```

```
<ipython-input-7-f24fa4352e95>:4: LangChainDeprecationWarning: The class `OpenAIEmbeddings` was deprecated in LangChain 0.0.9 and will be removed in 1.0. An updated versi
  vectorstore = Chroma.from_documents(documents=splits, embedding=OpenAIEmbeddings())
```

```
print(vectorstore._collection.count())
```

```
11
```

```
print(vectorstore._collection.get())
```

```
{'ids': ['3a550eec-2fab-40a1-8b74-ea1a26fb34c2', '905a995d-6fd2-46f2-ae97-2cc010f00cde', 'f5109d13-0b7b-421e-885d-f558c24411f4', '0cfc1adc-df88-49bf-97fb-f7ba67446ed5', '
```

```
print("\nCollection 1 - ", vectorstore._collection.get(ids=['28651d9a-ab51-41f8-ab83-e68285623c4e'], include=["embeddings", "documents"]))
print("\nCollection 2 - ", vectorstore._collection.get(ids=['054dee19-19ed-4574-bc51-511060fd707a'], include=["embeddings", "documents"]))
print("\nCollection 3 - ", vectorstore._collection.get(ids=['2fd71cb4-835a-43c5-b920-b7e1be51c450'], include=["embeddings", "documents"]))
```

```
Collection 1 -  {'ids': [], 'embeddings': array([], dtype=float64), 'documents': [], 'uris': None, 'included': ['embeddings', 'documents'], 'data': None, 'metadatas': Nor

Collection 2 -  {'ids': [], 'embeddings': array([], dtype=float64), 'documents': [], 'uris': None, 'included': ['embeddings', 'documents'], 'data': None, 'metadatas': Nor

Collection 3 -  {'ids': [], 'embeddings': array([], dtype=float64), 'documents': [], 'uris': None, 'included': ['embeddings', 'documents'], 'data': None, 'metadatas': Nor
```

```
retriever = vectorstore.as_retriever()
```

```
from langchain import hub
prompt = hub.pull("rlm/rag-prompt")
```

```
/usr/local/lib/python3.11/dist-packages/langsmith/client.py:278: LangSmithMissingAPIKeyWarning: API key must be provided when using hosted LangSmith API
  warnings.warn(
```

```
from langchain_openai import ChatOpenAI
llm = ChatOpenAI()
```

```
from langchain_core.runnables import RunnablePassthrough
from langchain_core.output_parsers import StrOutputParser
```

```
def format_docs(docs):
  return "\n".join(doc.page_content for doc in docs)
```

```
rag_chain = ({"context" : retriever | format_docs, "question" : RunnablePassthrough()}
             | prompt
             | llm
             | StrOutputParser())
```

```
rag_chain.invoke("Are the recordings of the course available? For how long?")
```

```
'Yes, the recordings of the course are available for lifetime access.'
```

```
rag_chain.invoke("Are the testimonials for the course available? Name the studenst who have shared testimonials")
```

```
'Yes, testimonials for the course are available on the website. Students who have shared testimonials are Sahitya Raj, Manika Kaushik, Ashish Upreti, and Sathish Krishn
a.'
```

```
rag_chain.invoke("Are the certificates for the course provided?")
```

```
'Yes, certificates are provided for the course.'
```

```
rag_chain.invoke("What all projects are covered in the course?")
```

```
'The projects covered in the course include implementing GANs to generate handwritten digits, training a VAE to generate faces using the CelebA dataset, building a trans
former from scratch, and performing sentiment analysis using BERT. The course also covers topics such as attention mechanisms, transformers, fine-tuning techniques, and
vector databases. It offers hands-on learning experiences and allows participants to work on mini and major projects to enhance their skills in generative AI.'
```

```
from langchain_core.runnables import RunnableLambda
```

```
def print_prompt(prompt_text):
  print("Prompt - ", prompt_text)
  return prompt_text
```

```
rag_chain_with_print = ({"context" : retriever | format_docs, "question" : RunnablePassthrough()}
             | prompt
             | RunnableLambda(print_prompt)
             | llm
             | StrOutputParser())
```

```
rag_chain_with_print.invoke("What all projects are covered in the course?")
```

```
Prompt -  messages=[HumanMessage(content="You are an assistant for question-answering tasks. Use the following pieces of retrieved context to answer the question. If you
'The projects covered in the course include implementing a GAN to generate handwritten digits, training a VAE to generate faces using the CelebA dataset, code Transforme
r from scratch, and sentiment analysis using BERT. There are also mini-projects related to TensorFlow, TensorBoard, and various deep generative models. The course covers
a range of topics related to generative AI and provides hands-on learning opportunities.'
```

```
Start coding or generate with AI.
```