```
# 📌 Install dependencies
!pip install -q langchain==1.0.5 langchain-community==0.4.1 langchain-groq==1.0.0 langchain-google-genai==3.0.1 chromadb

```

```
!pip show langchain langchain-community langchain-groq langchain-google-genai chromadb pypdf
```

```
Requires: langchain-core, langgraph, pydantic
Required-by: 
---
Name: langchain-community
Version: 0.4.1
Summary: Community contributed LangChain integrations.
Home-page: 
Author: 
Author-email: 
License: MIT
Location: /usr/local/lib/python3.12/dist-packages
Requires: aiohttp, dataclasses-json, httpx-sse, langchain-classic, langchain-core, langsmith, numpy, pydantic-settings, Py
Required-by: 
---
Name: langchain-groq
Version: 1.0.0
Summary: An integration package connecting Groq and LangChain
Home-page: https://docs.langchain.com/oss/python/integrations/providers/groq
Author: 
Author-email: 
License: MIT
Location: /usr/local/lib/python3.12/dist-packages
Requires: groq, langchain-core
Required-by: 
---
Name: langchain-google-genai
Version: 3.0.1
Summary: An integration package connecting Google's genai package and LangChain
Home-page: 
Author: 
Author-email: 
License: MIT
Location: /usr/local/lib/python3.12/dist-packages
Requires: filetype, google-ai-generativelanguage, langchain-core, pydantic
Required-by: 
---
Name: chromadb
Version: 1.3.4
Summary: Chroma.
Home-page: https://github.com/chroma-core/chroma
Author: 
Author-email: Jeff Huber <jeff@trychroma.com>, Anton Troynikov <anton@trychroma.com>
License: 
Location: /usr/local/lib/python3.12/dist-packages
Requires: bcrypt, build, grpcio, httpx, importlib-resources, jsonschema, kubernetes, mmh3, numpy, onnxruntime, opentelemet
Required-by: 
---
Name: pypdf
Version: 6.6.2
Summary: A pure-python PDF library capable of splitting, merging, cropping, and transforming PDF files
Home-page: 
Author: 
Author-email: Mathieu Fenniak <biziqe@mathieu.fenniak.net>
License: 
Location: /usr/local/lib/python3.12/dist-packages
Requires: 
Required-by: 
```

```
from google.colab import userdata
GEMINI_API_KEY = userdata.get('GEMINI_API_KEY')
GROQ_API_KEY = userdata.get('GROQ_API_KEY')
```

```
from langchain_community.document_loaders import PyPDFLoader

file_path = "/content/ai_syllabus.pdf"
loader = PyPDFLoader(file_path)
doc = loader.load()
```

```
from langchain_text_splitters import RecursiveCharacterTextSplitter

splitter = RecursiveCharacterTextSplitter(
    chunk_size=500,
    chunk_overlap=200
)

splits = splitter.split_documents(doc)
```

```
print(f"✅ Split into {len(splits)} chunks.")

# Preview first chunk
print("\n--- First Chunk ---\n")
print(splits[0].page_content[:500])
```

```
✅ Split into 14 chunks.

--- First Chunk ---

GUJARAT TECHNOLOGICAL UNIVERSITY
Program Name: Engineering
Level: Degree
Branch: Course / Subject Code: BE02000041
Course / Subject Name:  Fundamental of AI


w.e.f. 2024-25                                                    http://syllabus.gtu.ac.in/



Prerequisite
:
� Basic of computing and fundamental knowledge of problem solving techniques.
� Understanding of key concepts related to algorithms.
Rationale:
```

```
from langchain_google_genai import GoogleGenerativeAIEmbeddings
from langchain_classic.vectorstores import Chroma

embeddings = GoogleGenerativeAIEmbeddings(model="models/gemini-embedding-001", google_api_key=GEMINI_API_KEY)

# Create Chroma vector store
vector_store = Chroma(
    embedding_function=embeddings,
    persist_directory="rag_chroma_db",
    collection_name="ipl_docs"
)

# Add documents
vector_store.add_documents(splits)
```

```
/tmp/ipython-input-1672483023.py:7: LangChainDeprecationWarning: The class `Chroma` was deprecated in LangChain 0.2.9 and wi
  vector_store = Chroma(
['7cb5aabf-27b7-43d9-843a-574b36b4d211',
 '7ddde5d1-06a0-42f8-8a52-3e3d91c91e98',
 '91e87ecb-768b-416d-a3f5-5fabf41bc85c',
 'e42b218d-53e6-47db-af84-37dfe575ef2d',
 '2e8f5779-8c35-4fa5-a675-fcfcc08032d1',
 '0668c522-c519-4d35-8a21-6f0f1932fc67',
 '506b03aa-2589-44c6-86b4-3b773cac291d',
 '68e82b9c-8ed2-40f9-9d36-e113aa52e5b1',
 'b857db30-5112-4087-a374-898fe308068d',
 '22d77e24-23b0-4beb-8a7b-a41676b94351',
 'f2fffaea-8a08-4dea-b9ff-741d8b8663b3',
 '22155772-447f-4300-a68c-4f7822c3a302',
 'd9883e8d-1507-4cac-8304-7d1ce6af3af4',
 'f45ca1c9-0be3-466b-a60e-6d4e1e31cde7']
```

```
retriever = vector_store.as_retriever(search_kwargs={"k": 2})
```

```
from langchain_groq import ChatGroq
from google.colab import userdata
from langchain_groq import ChatGroq

# Load API key
GROQ_API_KEY = userdata.get('GROQ_API_KEY')
llm = ChatGroq(
    model="openai/gpt-oss-20b",
    api_key=GROQ_API_KEY,
    temperature=0.3,
    max_tokens=5000
)
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
from langchain_classic.chains import RetrievalQA

```

```
qa_chain = RetrievalQA.from_chain_type(
    llm=llm,
    retriever=retriever,
    return_source_documents=True
)
```

```
# Query 1
query = "which chapters are there?"
response = qa_chain.invoke({"query": query})

print("Query:", query)
print("Answer:", response["result"])

# Query 2
query2 = "What are their individual weightage?"
response2 = qa_chain.invoke({"query": query2})

print("\nQuery:", query2)
print("Answer:", response2["result"])

```

```
ncepts of Production, Agents and Environments<br>• Characteristics of Intelligent Agents<br>• Concept of Rationality<br>• Na
```

```
ighted in this particular specification.
```