

```
!pip install -U langchain langchain-community huggingface_hub faiss-cpu sentence-transformers langchain-huggingface transformers
```

```
Requirement already satisfied: langchain in /usr/local/lib/python3.12/dist-packages (0.3.27)
Collecting langchain-community
  Downloading langchain_community-0.3.27-py3-none-any.whl.metadata (2.9 kB)
Requirement already satisfied: huggingface_hub in /usr/local/lib/python3.12/dist-packages (0.34.4)
Collecting faiss-cpu
  Downloading faiss_cpu-1.12.0-cp312-cp312-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl.metadata (5.1 kB)
Requirement already satisfied: sentence-transformers in /usr/local/lib/python3.12/dist-packages (5.1.0)
Collecting langchain-huggingface
  Downloading langchain_huggingface-0.3.1-py3-none-any.whl.metadata (996 bytes)
Requirement already satisfied: transformers in /usr/local/lib/python3.12/dist-packages (4.55.2)
Collecting transformers
  Downloading transformers-4.55.3-py3-none-any.whl.metadata (41 kB)
42.0/42.0 kB 1.7 MB/s eta 0:00:00
Requirement already satisfied: torch in /usr/local/lib/python3.12/dist-packages (2.8.0+cu126)
Requirement already satisfied: accelerate in /usr/local/lib/python3.12/dist-packages (1.10.0)
Collecting bitsandbytes
  Downloading bitsandbytes-0.47.0-py3-none-manylinux_2_24_x86_64.whl.metadata (11 kB)
Requirement already satisfied: langchain-core<1.0.0,>=0.3.72 in /usr/local/lib/python3.12/dist-packages (from langchain) (0.3.72)
Requirement already satisfied: langchain-text-splitters<1.0.0,>=0.3.9 in /usr/local/lib/python3.12/dist-packages (from langchain) (0.3.9)
Requirement already satisfied: langsmith<=0.1.17 in /usr/local/lib/python3.12/dist-packages (from langchain) (0.4.14)
Requirement already satisfied: pydantic<3.0.0,>=2.7.4 in /usr/local/lib/python3.12/dist-packages (from langchain) (2.11.7)
Requirement already satisfied: SQLAlchemy<3,>=1.4 in /usr/local/lib/python3.12/dist-packages (from langchain) (2.0.43)
Requirement already satisfied: requests<3,>=2 in /usr/local/lib/python3.12/dist-packages (from langchain) (2.32.4)
Requirement already satisfied: PyYAML<=5.3 in /usr/local/lib/python3.12/dist-packages (from langchain) (6.0.2)
Requirement already satisfied: aiohttp<4.0.0,>=3.8.3 in /usr/local/lib/python3.12/dist-packages (from langchain-community) (3.10.10)
Requirement already satisfied: tenacity<=8.4.0,<10,>=8.1.0 in /usr/local/lib/python3.12/dist-packages (from langchain-community) (8.4.0)
Collecting dataclasses-json<0.7,>=0.5.7 (from langchain-community)
  Downloading dataclasses_json-0.6.7-py3-none-any.whl.metadata (25 kB)
Requirement already satisfied: pydantic-settings<3.0.0,>=2.4.0 in /usr/local/lib/python3.12/dist-packages (from langchain-community) (2.6.0)
Requirement already satisfied: httpx-sse<1.0.0,>=0.4.0 in /usr/local/lib/python3.12/dist-packages (from langchain-community) (0.4.0)
Requirement already satisfied: numpy<=1.26.2 in /usr/local/lib/python3.12/dist-packages (from langchain-community) (2.0.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from huggingface_hub) (3.19.1)
Requirement already satisfied: fsspec<=2023.5.0 in /usr/local/lib/python3.12/dist-packages (from huggingface_hub) (2023.5.0)
Requirement already satisfied: packaging<=20.9 in /usr/local/lib/python3.12/dist-packages (from huggingface_hub) (25.0)
Requirement already satisfied: tqdm<=4.42.1 in /usr/local/lib/python3.12/dist-packages (from huggingface_hub) (4.67.1)
Requirement already satisfied: typing-extensions<=3.7.4.3 in /usr/local/lib/python3.12/dist-packages (from huggingface_hub) (4.12.2)
Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in /usr/local/lib/python3.12/dist-packages (from huggingface_hub) (1.1.3)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.12/dist-packages (from sentence-transformers) (1.6.1)
Requirement already satisfied: scipy in /usr/local/lib/python3.12/dist-packages (from sentence-transformers) (1.16.1)
Requirement already satisfied: Pillow in /usr/local/lib/python3.12/dist-packages (from sentence-transformers) (11.3.0)
Requirement already satisfied: tokenizers<=0.19.1 in /usr/local/lib/python3.12/dist-packages (from langchain-huggingface) (0.19.1)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2024.11.6)
Requirement already satisfied: safetensors<=0.4.3 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.6.2)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch) (75.2.0)
Requirement already satisfied: sympy<=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch) (1.13.3)
Requirement already satisfied: networkx in /usr/local/lib/python3.12/dist-packages (from torch) (3.5)
Requirement already satisfied: Jinja2 in /usr/local/lib/python3.12/dist-packages (from torch) (3.1.6)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.80)
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/python3.12/dist-packages (from torch) (9.10.2.21)
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.4.1)
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib/python3.12/dist-packages (from torch) (11.3.0.4)
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/lib/python3.12/dist-packages (from torch) (10.3.7.77)
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/lib/python3.12/dist-packages (from torch) (11.7.1.2)
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/lib/python3.12/dist-packages (from torch) (12.5.4.2)
Requirement already satisfied: nvidia-cusparselt-cu12==0.7.1 in /usr/local/lib/python3.12/dist-packages (from torch) (0.7.1)
Requirement already satisfied: nvidia-nccl-cu12==2.27.3 in /usr/local/lib/python3.12/dist-packages (from torch) (2.27.3)
```

```
import gc
import torch

# Clear any existing models from memory
gc.collect()
if torch.cuda.is_available():
    torch.cuda.empty_cache()
    print(f"GPU available: {torch.cuda.get_device_name()}")
    print(f"GPU memory: {torch.cuda.get_device_properties(0).total_memory / 1e9:.1f} GB")
else:
    print("Using CPU only")

# Check system RAM
import psutil
ram_gb = psutil.virtual_memory().total / (1024**3)
print(f"System RAM: {ram_gb:.1f} GB")
```

```

if ram_gb < 8:
    print("Warning: Less than 8GB RAM detected. Consider using smaller models.")
elif ram_gb < 16:
    print("Sufficient RAM for quantized models (8-16GB)")
else:
    print("Excellent RAM for larger models (16GB+)")

```

↳ Using CPU only
System RAM: 12.7 GB
Sufficient RAM for quantized models (8-16GB)

```

import os
import warnings
warnings.filterwarnings('ignore')

# Set your HuggingFace API token
os.environ["HUGGINGFACEHUB_API_TOKEN"] = "hf_ySVsPvHbstetUUBhBMwmdgmPagRmCSEBk"

# Basic imports
from langchain.docstore.document import Document
from langchain.text_splitter import RecursiveCharacterTextSplitter

print(" Environment setup complete!")

```

↳ Environment setup complete!

```

# Sample legal documents with GDPR content
docs = [
    """This contract is subject to GDPR compliance.
    All personal data must be processed lawfully, fairly, and transparently.
    Data subjects have the right to access, rectify, and erase their personal data.
    """,
    """The agreement includes clauses for termination and arbitration.
    GDPR requires data deletion upon request by the user.
    Data controllers must implement appropriate technical and organizational measures.
    """,
    """This is a confidentiality clause. Information shared must not be disclosed
    without prior written consent, except as required by GDPR or applicable law.
    Personal data breaches must be reported within 72 hours under GDPR.
    """,
]

```

```

# Document objects
documents = [Document(page_content=doc.strip()) for doc in docs]
print(f" Created {len(documents)} documents")

```

```

# Display the documents
for i, doc in enumerate(documents, 1):
    print(f"\nDocument {i}:")
    print(doc.page_content)

```

↳ Created 3 documents

```

Document 1:
This contract is subject to GDPR compliance.
    All personal data must be processed lawfully, fairly, and transparently.
    Data subjects have the right to access, rectify, and erase their personal data.

Document 2:
The agreement includes clauses for termination and arbitration.
    GDPR requires data deletion upon request by the user.
    Data controllers must implement appropriate technical and organizational measures.

Document 3:
This is a confidentiality clause. Information shared must not be disclosed
    without prior written consent, except as required by GDPR or applicable law.
    Personal data breaches must be reported within 72 hours under GDPR.

```

```

# Initialize text splitter
splitter = RecursiveCharacterTextSplitter(
    chunk_size=300,
    chunk_overlap=50,
    separators=["\n\n", "\n", ".", "!", "?", ",", " ", " "]
)

```

```
# Split documents
chunks = splitter.split_documents(documents)
print(f"Created {len(chunks)} chunks")
```

```
# Display chunks
for i, chunk in enumerate(chunks, 1):
    print(f"\nChunk {i}:")
    print(f"Content: {chunk.page_content}")
    print(f"Length: {len(chunk.page_content)} characters")
```

➦ Created 3 chunks

```
Chunk 1:
Content: This contract is subject to GDPR compliance.
        All personal data must be processed lawfully, fairly, and transparently.
        Data subjects have the right to access, rectify, and erase their personal data.
Length: 207 characters

Chunk 2:
Content: The agreement includes clauses for termination and arbitration.
        GDPR requires data deletion upon request by the user.
        Data controllers must implement appropriate technical and organizational measures.
Length: 210 characters

Chunk 3:
Content: This is a confidentiality clause. Information shared must not be disclosed
        without prior written consent, except as required by GDPR or applicable law.
        Personal data breaches must be reported within 72 hours under GDPR.
Length: 229 characters
```

```
from langchain_community.embeddings import HuggingFaceEmbeddings
from langchain_community.vectorstores import FAISS
```

```
print("Creating embeddings...")
```

```
try:
    # Initialize embeddings
    embedding = HuggingFaceEmbeddings(
        model_name="sentence-transformers/all-mpnet-base-v2",
        model_kwargs={'device': 'cpu'}
    )

    print("✅ Embeddings initialized successfully!")

    # Test embedding
    test_text = "GDPR compliance test"
    test_embedding = embedding.embed_query(test_text)
    print(f"✅ Test embedding created, dimension: {len(test_embedding)}")

except Exception as e:
    print(f" Error with embeddings: {e}")
    raise
```

```

↳ Creating embeddings...
/tmp/ipython-input-3507374516.py:8: LangChainDeprecationWarning: The class `HuggingFaceEmbeddings` was deprecated in LangCha
embedding = HuggingFaceEmbeddings(
modules.json: 100% 349/349 [00:00<00:00, 30.4kB/s]

config_sentence_transformers.json: 100% 116/116 [00:00<00:00, 10.5kB/s]

README.md: 11.6k/? [00:00<00:00, 618kB/s]

sentence_bert_config.json: 100% 53.0/53.0 [00:00<00:00, 4.93kB/s]

config.json: 100% 571/571 [00:00<00:00, 37.7kB/s]

model.safetensors: 100% 438M/438M [00:05<00:00, 76.0MB/s]

tokenizer_config.json: 100% 363/363 [00:00<00:00, 39.1kB/s]

vocab.txt: 232k/? [00:00<00:00, 4.52MB/s]

tokenizer.json: 466k/? [00:00<00:00, 19.2MB/s]

special_tokens_map.json: 100% 239/239 [00:00<00:00, 18.4kB/s]

config.json: 100% 190/190 [00:00<00:00, 14.3kB/s]
✅ Embeddings initialized successfully!
✅ Test embedding created, dimension: 768

```

```

try:
    print("Building vector store...")

    # Create FAISS vector store
    vectorstore = FAISS.from_documents(chunks, embedding)

    print("Vector store created successfully!")

    # Create retriever
    retriever = vectorstore.as_retriever(
        search_type="similarity",
        search_kwargs={"k": 3}
    )

    print("Retriever configured!")

    # Test retrieval
    test_query = "GDPR"
    test_docs = retriever.get_relevant_documents(test_query)
    print(f"Test retrieval successful: found {len(test_docs)} relevant documents")

except Exception as e:
    print(f"Error creating vector store: {e}")
    raise

```

```

↳ Building vector store...
Vector store created successfully!
Retriever configured!
Test retrieval successful: found 3 relevant documents

```

```

def test_search(query):
    print(f"\n🔍 Testing search for: '{query}'")
    try:
        docs = retriever.get_relevant_documents(query)
        print(f"Found {len(docs)} relevant documents:")

        for i, doc in enumerate(docs, 1):
            print(f"\n--- Result {i} ---")
            print(doc.page_content.strip())

        return docs
    except Exception as e:
        print(f"Search error: {e}")
        return []

# Test searches
test_search("GDPR compliance")
test_search("data deletion")

```

```
test_search("personal data")
```

```
--- Result 1 ---
➤ The agreement includes clauses for termination and arbitration.
  GDPR requires data deletion upon request by the user.
  Data controllers must implement appropriate technical and organizational measures.
```

```
--- Result 2 ---
This contract is subject to GDPR compliance.
  All personal data must be processed lawfully, fairly, and transparently.
  Data subjects have the right to access, rectify, and erase their personal data.
```

```
--- Result 3 ---
This is a confidentiality clause. Information shared must not be disclosed
  without prior written consent, except as required by GDPR or applicable law.
  Personal data breaches must be reported within 72 hours under GDPR.
```

```
🔍 Testing search for: 'data deletion'
Found 3 relevant documents:
```

```
--- Result 1 ---
The agreement includes clauses for termination and arbitration.
  GDPR requires data deletion upon request by the user.
  Data controllers must implement appropriate technical and organizational measures.
```

```
--- Result 2 ---
This contract is subject to GDPR compliance.
  All personal data must be processed lawfully, fairly, and transparently.
  Data subjects have the right to access, rectify, and erase their personal data.
```

```
--- Result 3 ---
This is a confidentiality clause. Information shared must not be disclosed
  without prior written consent, except as required by GDPR or applicable law.
  Personal data breaches must be reported within 72 hours under GDPR.
```

```
🔍 Testing search for: 'personal data'
Found 3 relevant documents:
```

```
--- Result 1 ---
This contract is subject to GDPR compliance.
  All personal data must be processed lawfully, fairly, and transparently.
  Data subjects have the right to access, rectify, and erase their personal data.
```

```
--- Result 2 ---
This is a confidentiality clause. Information shared must not be disclosed
  without prior written consent, except as required by GDPR or applicable law.
  Personal data breaches must be reported within 72 hours under GDPR.
```

```
--- Result 3 ---
The agreement includes clauses for termination and arbitration.
  GDPR requires data deletion upon request by the user.
  Data controllers must implement appropriate technical and organizational measures.
[Document(id='ba4148b2-d6eb-442c-9b63-77ef48f8af58', metadata={}, page_content='This contract is subject to GDPR
compliance.\n    All personal data must be processed lawfully, fairly, and transparently.\n    Data subjects have the
right to access, rectify, and erase their personal data.'),
 Document(id='a2e4481d-7889-42d7-8fa0-ece9f107f3a9', metadata={}, page_content='This is a confidentiality clause.
Information shared must not be disclosed\n    without prior written consent, except as required by GDPR or applicable
law.\n    Personal data breaches must be reported within 72 hours under GDPR.'),
 Document(id='e19fc5b1-a3d5-492f-9507-98b288a3ddcb', metadata={}, page_content='The agreement includes clauses for
termination and arbitration.\n    GDPR requires data deletion upon request by the user.\n    Data controllers must
implement appropriate technical and organizational measures.')]

```

```
llm_method = None
```

```
if llm_method is None:
```

```
    print("Trying alternative LLM method...")
```

```
try:
```

```
    from transformers import pipeline
    from langchain.llms import HuggingFacePipeline
```

```
    # Create local pipeline
    hf_pipeline = pipeline(
        "text2text-generation",
        model="google/flan-t5-small",
        max_new_tokens=256,
        temperature=0.1
    )
```

```
    # Wrap in LangChain
    llm = HuggingFacePipeline(pipeline=hf_pipeline)
```

```
    # Test the LLM
```

```
test_response = llm.invoke("What is GDPR?")
print("LLM Method 2 (Local Pipeline) successful!")
print(f"Test response: {test_response}")
```

```
llm_method = "local"
```

```
except Exception as e:
    print(f"Method 2 also failed: {e}")
    llm_method = None
```

➦ Trying alternative LLM method...

```
config.json:      1.40k/? [00:00<00:00, 25.1kB/s]
```

```
model.safetensors: 100%
```

```
308M/308M [00:14<00:00, 26.0MB/s]
```

```
generation_config.json: 100%
```

```
147/147 [00:00<00:00, 2.06kB/s]
```

```
tokenizer_config.json:   2.54k/? [00:00<00:00, 28.7kB/s]
```

```
spiece.model: 100%
```

```
792k/792k [00:00<00:00, 240kB/s]
```

```
tokenizer.json:      2.42M/? [00:00<00:00, 52.0MB/s]
```

```
special_tokens_map.json:  2.20k/? [00:00<00:00, 129kB/s]
```

```
Device set to use cpu
```

```
The following generation flags are not valid and may be ignored: ['temperature']. Set `TRANSFORMERS_VERBOSITY=info` for more
/tmp/ipython-input-1690579610.py:18: LangChainDeprecationWarning: The class `HuggingFacePipeline` was deprecated in LangChain
```

```
llm = HuggingFacePipeline(pipeline=hf_pipeline)
The following generation flags are not valid and may be ignored: ['temperature']. Set `TRANSFORMERS_VERBOSITY=info` for more
LLM Method 2 (Local Pipeline) successful!
Test response: GDPR
```

```
from langchain.chains import RetrievalQA
```

```
try:
```

```
    print("Creating QA Chain...")
```

```
    if llm_method in ["endpoint", "local"]:
        qa_chain = RetrievalQA.from_chain_type(
            llm=llm,
            chain_type="stuff",
            retriever=retriever,
            return_source_documents=True
        )
        print("QA Chain created successfully!")
```

```
    else:
```

```
        # Custom QA for fallback method
```

```
        class SimpleQAChain:
```

```
            def __init__(self, retriever, llm):
                self.retriever = retriever
                self.llm = llm
```

```
            def invoke(self, query_dict):
                query = query_dict["query"]
```

```
                # Get relevant documents
```

```
                source_docs = self.retriever.get_relevant_documents(query)
```

```
                # Create context from documents
```

```
                context = "\n\n".join([doc.page_content for doc in source_docs])
```

```
                # Generate response
```

```
                prompt = f"Based on this context: {context}\n\nQuestion: {query}\nAnswer:"
                result = self.llm.invoke(prompt)
```

```
                return {
                    "result": result,
                    "source_documents": source_docs
                }
```

```
        qa_chain = SimpleQAChain(retriever, llm)
```

```
        print("Custom QA Chain created!")
```

```
except Exception as e:
```

```
    print(f" Error creating QA chain: {e}")
```

```
qa_chain = None
```

→ Creating QA Chain...
QA Chain created successfully!

```
def query_legal_documents(query):
    """Query the legal documents with comprehensive error handling"""

    print(f"\n{'='*60}")
    print(f"🔍 Query: {query}")
    print(f"\n{'='*60}")

    try:
        if qa_chain is not None:
            # Use QA chain
            response = qa_chain.invoke({"query": query})

            print("\n📄 ANSWER:")
            print("-" * 40)
            print(response["result"])

            print("\n📄 SOURCE DOCUMENTS:")
            print("-" * 40)
            for i, doc in enumerate(response["source_documents"], 1):
                print(f"\nSource {i}:")
                print(doc.page_content.strip())

            return response

        else:
            # Fallback to direct search
            print("Using fallback search method...")
            docs = retriever.get_relevant_documents(query)

            print(f"\n📄 RELEVANT DOCUMENTS ({len(docs)} found):")
            print("-" * 40)

            for i, doc in enumerate(docs, 1):
                print(f"\nDocument {i}:")
                print(doc.page_content.strip())

            return {"source_documents": docs, "result": "Search completed using fallback method."}

    except Exception as e:
        print(f"Query failed: {e}")
        return None

test_queries = [
    "What are the data processing requirements under GDPR?",
    "What happens when a user requests data deletion?",
    "What are the confidentiality requirements?",
    "What technical measures are required for GDPR compliance?",
    "What is the timeframe for reporting data breaches?"
]

for query in test_queries:
    query_legal_documents(query)
```

→ The following generation flags are not valid and may be ignored: ['temperature']. Set `TRANSFORMERS_VERBOSITY=info` for more details.

🔍 Query: What are the data processing requirements under GDPR?

The following generation flags are not valid and may be ignored: ['temperature']. Set `TRANSFORMERS_VERBOSITY=info` for more details.

📄 ANSWER:

Data subjects have the right to access, rectify, and erase their personal data

📄 SOURCE DOCUMENTS:

Source 1:
The agreement includes clauses for termination and arbitration.

GDPR requires data deletion upon request by the user.
Data controllers must implement appropriate technical and organizational measures.

Source 2:

This contract is subject to GDPR compliance.
All personal data must be processed lawfully, fairly, and transparently.
Data subjects have the right to access, rectify, and erase their personal data.

Source 3:

This is a confidentiality clause. Information shared must not be disclosed
without prior written consent, except as required by GDPR or applicable law.
Personal data breaches must be reported within 72 hours under GDPR.

Query: What happens when a user requests data deletion?

The following generation flags are not valid and may be ignored: ['temperature']. Set `TRANSFORMERS_VERBOSITY=info` for more

ANSWER:

Data controllers must implement appropriate technical and organizational measures

SOURCE DOCUMENTS:

Source 1:

The agreement includes clauses for termination and arbitration.
GDPR requires data deletion upon request by the user.
Data controllers must implement appropriate technical and organizational measures.

Source 2:

This contract is subject to GDPR compliance.
All personal data must be processed lawfully, fairly, and transparently.
Data subjects have the right to access, rectify, and erase their personal data.

Source 3:

This is a confidentiality clause. Information shared must not be disclosed
without prior written consent, except as required by GDPR or applicable law.
Personal data breaches must be reported within 72 hours under GDPR.

```
print("\n" + "="*60)
print(" SYSTEM STATUS SUMMARY")
print("="*60)

print(f" Documents loaded: {len(documents)}")
print(f" Chunks created: {len(chunks)}")
print(f" Embeddings: {' Working' if 'embedding' in locals() else ' Failed'}")
print(f" Vector Store: {' Working' if 'vectorstore' in locals() else ' Failed'}")
print(f" Retriever: {' Working' if 'retriever' in locals() else ' Failed'}")
print(f" LLM Method: {llm_method}")
print(f" QA Chain: {' Working' if qa_chain is not None else ' Failed'}")

print(f"\n System is ready for legal document queries!")
```



```
=====
SYSTEM STATUS SUMMARY
=====
Documents loaded: 3
Chunks created: 3
Embeddings: Working
Vector Store: Working
Retriever: Working
LLM Method: local
QA Chain: Working

System is ready for legal document queries!
```

```
while True:
    user_query = input("\n Enter your legal question (or 'quit' to exit): ")
    if user_query.lower() in ['quit', 'exit', 'q']:
        break
    query_legal_documents(user_query)
```



Enter your legal question (or 'quit' to exit): what should be processed lawfully and fairly
The following generation flags are not valid and may be ignored: ['temperature']. Set `TRANSFORMERS_VERBOSITY=info` for more


```
=====
🔍 Query: what should be processed lawfully and fairly
=====
```

```
📄 ANSWER:
```

```
-----
personal data
```

```
📄 SOURCE DOCUMENTS:
```

```
-----
Source 1:
```

```
This contract is subject to GDPR compliance.
```

```
    All personal data must be processed lawfully, fairly, and transparently.
```

```
    Data subjects have the right to access, rectify, and erase their personal data.
```

```
Source 2:
```

```
The agreement includes clauses for termination and arbitration.
```

```
    GDPR requires data deletion upon request by the user.
```

```
    Data controllers must implement appropriate technical and organizational measures.
```

```
Source 3:
```