Study Assistant RAG App

```
!pip install langchain_core
```

Show hidden output

```
!pip install langchain
```

Show hidden output

```
!pip install langchain langchain-community
```

Show hidden output

```
!pip install pypdf
```

Show hidden output

```
!pip install colab-xterm
```

Show hidden output

```
%load_ext colabxterm
```

Start coding or generate with AI.

```
%xterm
```

Show hidden output

```
%xterm
```

Show hidden output

```
!pip install langchain_ollama
```

Show hidden output

```
from langchain_ollama import OllamaLLM

model = OllamaLLM(model="llama3.2")
model.invoke("Come up with 10 names for a song about parrots")
```

```
"Here are 10 potential song title ideas about parrots:\n\n1. "Rainbow Wings"\n2. "Flock to Me"\n3. "Squawk of the Soul"\n4. "Parrot\'s
Lament"\n5. "Colorful Eyes"\n6. "Fly Away Free"\n7. "Tropical Temptation"\n8. "Birds of a Feather"\n9. "Sunset Serenade (For Polly)"\n1
0. "Winged Wisdom"\n\nThese titles aim to capture the vibrant colors, lively personalities, and whimsical nature of parrots that inspir
ed your song. Feel free to pick the one that resonates with you the most or use them as inspiration to come up with your own unique tit
```

```
from langchain_ollama import OllamaLLM

model = OllamaLLM(model="llama3.2")
model.invoke("Come up with 10 names for a song about parrots")
```

```
"Here are ten potential song title ideas about parrots:\n\n1. "Squawk of the Soul"\n2. "Parrot\'s Lament"\n3. "Rainbow Wings"\n4. "Feat
hered Friends Forever"\n5. "The Parrot\'s Eye View"\n6. "Colors of the Skies"\n7. "Polly Wants More"\n8. "Wild at Heart"\n9. "Tropical
Dreaming"\n10. "Skybound Spirits"\n\nThese titles aim to capture the vibrant colors, playful personalities, and exotic nature of parrot
```

```
model.invoke('hi there ')
```

```
"How can I assist you today?"
```

```
model.invoke('do you want to know my name')
```

```
"I'm curious! Yes, I'd love to know your name. It's always nice to learn the names of the people I interact with. Is it okay if I share it with you?"
```

```
model.invoke('but first you tell me your name')
```

```
"I don\'t have a personal name. I\'m an AI designed to assist and communicate with users, but I don\'t have a personal identity or emotions. I exist solely to provide information and help with questions and tasks.\n\nIf you\'d like, I can suggest some options for how we can refer to each other in our conversation. For example, I could be called "Assistant" or "AI Companion," or we could use a name that \'s not tied to my programming or functionality. Let me know if there\'s anything specific you\'d like to do!"
```

```
model.invoke('ok thats nice')
```

```
"Can I help you with something specific or would you like to chat?"
```

```
model.invoke('ok i tell you wait a sec')
```

```
"Take your time, I'll be here when you're ready to continue."
```

```
from langchain_core.output_parsers import StrOutputParser
parser = StrOutputParser()
chain = model | parser
chain.invoke("What is the capital of France?")
```

```
"The capital of France is Paris."
```

```
!pip install rapidocr-onnxruntime
```

Show hidden output

```
from langchain_community.document_loaders import PyPDFLoader

loader = PyPDFLoader("/content/Chapter 02 - Central Processing Unit (CPU).pdf", extract_images=True)
pages = loader.load_and_split()
pages
```

```
[Document(metadata={'source': '/content/Chapter 02 - Central Processing Unit (CPU).pdf', 'page': 0}, page_content='Deborah Morley \nCharles S. Parker \n15th Edition \nUnderstanding Computers \nToday and Tomorrow \nComprehensive \n \nCopyright 2015 Cengage Learning \nChapter 2 \nThe Systems Unit:  \nProcessing and Memory'),
 Document(metadata={'source': '/content/Chapter 02 - Central Processing Unit (CPU).pdf', 'page': 1}, page_content='Learning Objectives \n1. Understand how data and programs are represented to a \ncomputer and be able to identify a few of the coding systems \nused to accomplish this.  \n2. Explain the functions of the hardware components \ncommonly found inside the system unit, such as the CPU, \nGPU, memory, buses, and expansion cards. \n3. Describe how peripheral devices or other hardware can be \nadded to a computer. \n4. Understand how a computer’s CPU and memory components \nprocess program instructions and data. \n \nUnderstanding Computers: Today and Tomorrow, 15th Edition 2 2'),
 Document(metadata={'source': '/content/Chapter 02 - Central Processing Unit (CPU).pdf', 'page': 2}, page_content='Learning Objectives \n5. Name and evaluate several strategies that can be used today \nfor speeding up the operations of a computer. \n6. List some processing technologies that may be used in future \ncomputers. \n \nUnderstanding Computers: Today and Tomorrow, 15th Edition 3'),
 Document(metadata={'source': '/content/Chapter 02 - Central Processing Unit (CPU).pdf', 'page': 3}, page_content='Overview \n• Explain how computers represent data and program \ninstructions. \n• Explain how the CPU and memory are arranged with other \ncomponents inside the system unit. \n• Explain how a CPU performs processing tasks. \n• Identify strategies that can be used today to create faster and \nbetter computers in the future. \n \nUnderstanding Computers: Today and Tomorrow, 15th Edition 4 4'),
 Document(metadata={'source': '/content/Chapter 02 - Central Processing Unit (CPU).pdf', 'page': 4}, page_content='Data and Program Representation \n• Digital Data Representation \n– Coding Systems \n• Used to represent data and programs in a manner \nunderstood by the computer \n– Digital Computers \n• Can only understand two states, off and on \n(0 and 1) \n– Digital Data Representation \n• The process of representing  \ndata in digital form so it can be  \nunderstood by a computer \n \nUnderstanding Computers: Today and Tomorrow, 15th Edition 5 \nOpen = 0 (off)\nCopyright @ 2015 Cengage Leaming\nNegative\nPositive\nClosed = 1 (on)\n=0 (off)\n= 1 (on)\nCIRCUIT\nMAGNETIZATIONFIGURE2-1\nWays ofrepresenting\n0and 1.Binary\ncomputersrecognize\nonlytwo states-off\nand on—usually\nrepresented by O and 1.'),
 Document(metadata={'source': '/content/Chapter 02 - Central Processing Unit (CPU).pdf', 'page': 5}, page_content='Digital Data Representation \n– Bit \n• The smallest unit of data that a  \nbinary computer can recognize \n(a single 1 or 0) \n– Byte = 8 bits \n• Byte terminology used to express  \nthe size of documents and other  \nfiles, programs, etc. \n– Prefixes are often used to express larger quantities of \nbytes: kilobyte (KB), megabyte (MB), gigabyte (GB), \nterabyte (TB), petabyte (PB), exabyte (EB), zettabyte (ZB), \nyottabyte (YB). \nUnderstanding Computers: Today and Tomorrow, 15th Edition 6 \nBit\n0110000\nByte\nApproximate\nAbbreviation\nSize\nKB\n1thousandbytes\nCopyright @ 2015 Cengage Leaming\nMB\n1million bytes\nGB\n1billionbytes\nTB\n1 trillion bytes\nPB\n1,000terabytes\nEB\n1,000petabtyes\nZB\n1,000exabytes\nYB\n1,000zettabytesEIGURE\n2-2\nBitsandbytes.\nDocument size,\nstorage capacity,and\nmemory\ncapacity\nare\nallmeasured inbytes.'),
 Document(metadata={'source': '/content/Chapter 02 - Central Processing Unit (CPU).pdf', 'page': 6}, page_content='Representing Numerical Data \n• The Binary Numbering System \n– Numbering system \n• A way of representing numbers \n– Decimal numbering system \n• Uses 10 symbols (0-9) \n– Binary numbering system \n• Uses only two symbols (1 and 0) to represent all \npossible numbers \n– In both systems, the position of the digits determines the \npower to which the base number (such as 10 or 2) is raised \n \nUnderstanding Computers: Today and Tomorrow, 15th Edition 7'),
 Document(metadata={'source': '/content/Chapter 02 - Central Processing Unit (CPU).pdf', 'page': 7}, page_content='Representing Numerical Data \nUnderstanding Computers: Today and Tomorrow, 15th Edition 8 \nThe decimal\nnumber\n103\n102\n101\n10\n10 raised to\n7,216\n(100)\ndifferent \n(1,000)\n(10)\n(1)\npowers\nDECIMAL NUMBERING\nSYSTEM\nEach place value in\na decimal number\n2\n6\nrepresents 10 raised to\nthe appropriate power.\nmeans 6 x1\n6\n→means1x10\n10\nmeans2x100=200\nmeans 7 x1,000
```

```
=7,000\n7,216\nThe binary \nnumber\n28\nm乙\n2®\n2\n2 raised to\n1001\n(1)\ndifferent\nBINARYNUMBERING\npowers\nSYSTEM\nEach place
value in \n a binary number \n0\n0\nrepresents 2 raised to\nthe appropriate power.\n→means1x1 = 1\nCopyright @ 2015 Cengage Leaming?
\nmeans0x2= 0\nmeans 0x4 = 0\nmeans1x8 = 8\n9\nDecimal\nequivalentFIGURE2-3\nExamplesofusing\nthedecimaland\nbinary
numbering\nsystems.'),
 Document(metadata={'source': '/content/Chapter 02 - Central Processing Unit (CPU).pdf', 'page': 8}, page_content='Coding Systems for
Text-Based Data \n• ASCII (American Standard Code for \nInformation Interchange) \n– Coding system traditionally used \nwith personal
computers \n• EBCDIC (Extended Binary-Coded \nDecimal Interchange Code) \n– Developed by IBM, primarily for \nmainframes
\nUnderstanding Computers: Today and Tomorrow, 15th Edition 9
\nCHARACTER\nASCII\n0\n00110000\n1\n00110001\n2\n00110010\n3\n00110011\n4\n00110100\n5\n00110101\nA\n01000001\nB\n01000010\nC\n0100001
2015 0\n+\n00101011\n00100001\n#\n00100011FIGURE2-4\nSome\nextended\nASCI\ncode\nexamples.'),
 Document(metadata={'source': '/content/Chapter 02 - Central Processing Unit (CPU).pdf', 'page': 9}, page_content='Coding Systems for
Text-Based Data \n• Unicode \n– Newer code (32 bits per character is common) \n– Universal coding standard designed to represent
text-\nbased data written in any ancient or modern language \n– Replacing ASCII as the primary text-coding system \n \nUnderstanding
Computers: Today and Tomorrow, 15th Edition 10 \n铜\nR\nCopyright2015CengageLearning°
\nCHINESE\nGREEK\nHEBREW\nAMHARIC\nTIBETAN\nRUSSIANFIGURE2-5\nUnicode.Many\ncharacters,such\nas these, can be\nrepresented
```

```
from langchain.document_loaders import PyPDFLoader
from langchain.text_splitter import CharacterTextSplitter

pdf_loader = PyPDFLoader("/content/(ICT) Chp No1 short answers.pdf")
documents = pdf_loader.load()

text_splitter = CharacterTextSplitter
chunk_size=1000,
chunk_overlap=100
splitted_docs = text_splitter.split_documents(documents)

for i, chunk in enumerate(splitted_docs):
    print(f"Chunk {i + 1}:\n{chunk.page_content}\n{'-'*40}")
```

```
from langchain.prompts import PromptTemplate

template = """
You must Answer the question based on the context below. If you can't
answer the question, reply "I don't know".

Context: {context}

Question: {question}
"""

prompt = PromptTemplate.from_template(template)
print(prompt.format(context="Here is some context", question="Here is a question"))
```

```
You must Answer the question based on the context below. If you can't
answer the question, reply "I don't know".

Context: Here is some context

Question: Here is a question
```

```
chain = prompt | model | parser

chain.invoke({"context": "My parents named me Sajid Ali", "question": "What's your name'?"})
```

```
'My name is Sajid Ali.'
```

```
!pip install docarray
```

```
Requirement already satisfied: docarray in /usr/local/lib/python3.10/dist-packages (0.40.0)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from docarray) (1.26.4)
Requirement already satisfied: orjson>=3.8.2 in /usr/local/lib/python3.10/dist-packages (from docarray) (3.10.11)
Requirement already satisfied: pydantic>=1.10.8 in /usr/local/lib/python3.10/dist-packages (from docarray) (2.10.1)
Requirement already satisfied: rich>=13.1.0 in /usr/local/lib/python3.10/dist-packages (from docarray) (13.9.4)
Requirement already satisfied: types-requests>=2.28.11.6 in /usr/local/lib/python3.10/dist-packages (from docarray) (2.32.0.20241016)
Requirement already satisfied: typing-inspect>=0.8.0 in /usr/local/lib/python3.10/dist-packages (from docarray) (0.9.0)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.10/dist-packages (from pydantic>=1.10.8->docarray) (0.7.
Requirement already satisfied: pydantic-core==2.27.1 in /usr/local/lib/python3.10/dist-packages (from pydantic>=1.10.8->docarray) (2.27.
Requirement already satisfied: typing-extensions>=4.12.2 in /usr/local/lib/python3.10/dist-packages (from pydantic>=1.10.8->docarray) (4
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from rich>=13.1.0->docarray) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from rich>=13.1.0->docarray) (2.18.0)
Requirement already satisfied: urllib3>=2 in /usr/local/lib/python3.10/dist-packages (from types-requests>=2.28.11.6->docarray) (2.2.3)
Requirement already satisfied: mypy-extensions>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from typing-inspect>=0.8.0->docarray)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.0->rich>=13.1.0->docarray
```

```
!pip uninstall pydantic==1.10.8
```

```
Found existing installation: pydantic 2.10.1
Uninstalling pydantic-2.10.1:
  Would remove:
    /usr/local/lib/python3.10/dist-packages/pydantic-2.10.1.dist-info/*
    /usr/local/lib/python3.10/dist-packages/pydantic/*
Proceed (Y/n)? n
```

```
!pip install pydantic==2.10.1
```

```
Requirement already satisfied: pydantic==2.10.1 in /usr/local/lib/python3.10/dist-packages (2.10.1)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.10/dist-packages (from pydantic==2.10.1) (0.7.0)
Requirement already satisfied: pydantic-core==2.27.1 in /usr/local/lib/python3.10/dist-packages (from pydantic==2.10.1) (2.27.1)
Requirement already satisfied: typing-extensions>=4.12.2 in /usr/local/lib/python3.10/dist-packages (from pydantic==2.10.1) (4.12.2)
```

```
from langchain_community.embeddings import OllamaEmbeddings
embeddings = OllamaEmbeddings(model="llama3.2")
```

```
from langchain_community.vectorstores import DocArrayInMemorySearch
vectorstores= DocArrayInMemorySearch.from_documents(pages, embedding = embeddings)
```

```
retriever= vectorstores.as_retriever()
retriever.invoke("Learning Objectives")
```

```
[Document(metadata={'source': '/content/Chapter 02 - Central Processing Unit (CPU).pdf', 'page': 67}, page_content='Summary \n• Data and Program Representation \n• Inside the System Unit \n• How the CPU Works \n• Making Computers Faster and Better Now and in the Future \nUnderstanding Computers: Today and Tomorrow, 15th Edition 68'),
 Document(metadata={'source': '/content/Chapter 02 - Central Processing Unit (CPU).pdf', 'page': 56}, page_content='Making Computers Faster and Better \nNow and in the Future \n• Error check and defrag the hard drive periodically \n• Scan for viruses and spyware continually \n• Clean out dust once or twice a year \n– Buy a larger or second hard drive \n– Upgrade your Internet connection \n– Upgrade your video graphics card \n \n \nUnderstanding Computers: Today and Tomorrow, 15th Edition 57'),
 Document(metadata={'source': '/content/Chapter 02 - Central Processing Unit (CPU).pdf', 'page': 1}, page_content='Learning Objectives \n1. Understand how data and programs are represented to a \ncomputer and be able to identify a few of the coding systems \nused to accomplish this.  \n2. Explain the functions of the hardware components \ncommonly found inside the system unit, such as the CPU, \nGPU, memory, buses, and expansion cards. \n3. Describe how peripheral devices or other hardware can be \nadded to a computer. \n4. Understand how a computer’s CPU and memory components \nprocess program instructions and data. \n \nUnderstanding Computers: Today and Tomorrow, 15th Edition 2 2'),
 Document(metadata={'source': '/content/Chapter 02 - Central Processing Unit (CPU).pdf', 'page': 28}, page_content='Memory \n• Memory \n– Refers to chip-based storage located inside the system unit \n– Storage refers to the amount of long-term storage \navailable to a computer \n– Random Access Memory (RAM) \n• Computer’s main memory \n• Consists of chips arranged on a circuit board called a \nmemory module which are plugged into the \nmotherboard \n• Stores essential parts of operating system, programs, \nand data the computer is currently using  \n \n \nUnderstanding Computers: Today and Tomorrow, 15th Edition 29')]
```

```
from operator import itemgetter

chain= (
    {
        "context" : itemgetter('question') | retriever,
        "question": itemgetter('question')
    }
    | prompt
    | model
    | parser

)
chain.invoke({"question": "tell me the Learning Objectives"})
```

```
"The Learning Objectives are:\n\n1. Understand how data and programs are represented to a computer and be able to identify a few of the coding systems used to accomplish this.\n2. Explain the functions of the hardware components commonly found inside the system unit, such as the CPU, GPU, memory, buses, and expansion cards.\n3. Describe how peripheral devices or other hardware can be added to a computer.\n4. Understand how a computer’s CPU and memory components process program instructions and data."
```

```
chain.invoke({'question': 'what is Machine Language'})
```

```
"Machine Language is the lowest-level programming language used by computers. It consists of binary code (0s and 1s) that directly corresponds to specific machine-specific instructions. In other words, it is a set of symbols or codes that are used by the computer\'s processor (CPU) to understand and execute specific tasks.\n\nAccording to the context provided, this can be inferred from page 3 of the document where it mentions "Explain how computers represent data and program instructions" and also "Identify a few of the coding systems used to accomplish this." "
```

```
chain.invoke({'question': "can you tell me the "})
```