from langchain.chains import RetrievalQA

from langchain.embeddings import OpenAIEmbeddings

from langchain.vectorstores import FAISS

from langchain.document\_loaders import PyPDFLoader, WebLoader

from langchain.llms import OpenAI

# Task 1: PDF Chat

def pdf\_chat\_pipeline(pdf\_path, query):

# Load and process PDF

loader = PyPDFLoader(pdf\_path)

documents = loader.load\_and\_split()

# Create vector embeddings

embeddings = OpenAIEmbeddings()

vectorstore = FAISS.from\_documents(documents, embeddings)

# Setup Retrieval-based QA

qa = RetrievalQA.from\_chain\_type(llm=OpenAI(), retriever=vectorstore.as\_retriever())

return qa.run(query)

# Task 2: Website Chat

def website\_chat\_pipeline(url, query):

# Load and process website

loader = WebLoader([url])

documents = loader.load\_and\_split()

# Create vector embeddings

embeddings = OpenAIEmbeddings()

vectorstore = FAISS.from\_documents(documents, embeddings)

# Setup Retrieval-based QA

qa = RetrievalQA.from\_chain\_type(llm=OpenAI(), retriever=vectorstore.as\_retriever())

return qa.run(query)

# Example Usage

pdf\_response = pdf\_chat\_pipeline("example.pdf", "What is the data on page 2?")

website\_response = website\_chat\_pipeline("https://www.stanford.edu", "What programs are offered?")

print("PDF Response:", pdf\_response)

print("Website Response:", website\_response)