

```
!python -m pip install langchain-community langchainhub langchain-  
chroma langchain langchain-experimental --quiet  
!python -m pip install faiss-cpu --quiet  
!python -m pip install pypdf --quiet
```

```
import os  
import pandas as pd  
from typing import List, Tuple  
from langchain_openai import AzureOpenAIEmbeddings  
from typing_extensions import TypedDict  
  
from langchain_openai import AzureChatOpenAI  
from dotenv import load_dotenv  
  
load_dotenv(override=True)  
  
embedding_model_name = "text-embedding-3-small"  
embeddings = AzureOpenAIEmbeddings(  
    deployment=embedding_model_name,  
    model="text-embedding-3-small"  
)  
llm = AzureChatOpenAI(model="gpt4o")  
model_name = "gpt4o"  
df = pd.read_csv("./assignment-2.csv")  
  
texts = df["title"] + " " + df["description"]  
metadatas = df.apply(lambda row: {"course_id": row["course_id"],  
    "title": row["title"]}, axis=1).tolist()  
ids = df["course_id"].tolist()  
  
from langchain_chroma import Chroma  
vectorstore = Chroma.from_texts(  
    texts.tolist(),  
    embeddings,  
    metadatas=metadatas,  
    ids=ids,  
    collection_name="courses",  
    persist_directory="chroma_db" # persist to disk  
)  
  
class RecState(TypedDict):  
    profile: str  
    completed_ids: List[str]  
    results: List[Tuple[str, float, str]]  
    answer: str # store LLM response  
  
def recommend_node(state: RecState) -> RecState:  
    profile = state["profile"]  
    completed_ids = state.get("completed_ids", [])  
  
    # Similarity search from vectorstore
```

```

results = vectorstore.similarity_search_with_score(profile, k=5)
recs = []

for doc, score in results:
    cid = doc.metadata.get("course_id")
    title = doc.metadata.get("title", "Untitled")
    if cid and cid not in completed_ids:
        recs.append((cid, score, title))

state["results"] = recs
return state

from langchain_core.prompts import ChatPromptTemplate

prompt_template = ChatPromptTemplate.from_messages([
    ("system", "You are a helpful course recommendation assistant."),
    ("human", "Profile: {profile}\nCompleted: {completed_ids}\n\nTop\nMatches:\n{recs}\n\nGive top 5 course recommendations.")
])

def llm_node(state: RecState) -> RecState:
    recs = state["results"]
    rec_text = "\n".join(
        f"{cid} - {title} (similarity: {score:.4f})"
        for cid, score, title in recs
    )

    prompt = prompt_template.format(
        profile=state["profile"],
        completed_ids=state.get("completed_ids", []),
        recs=rec_text
    )

    response = llm.invoke(prompt)

    return {**state, "answer": response.content}

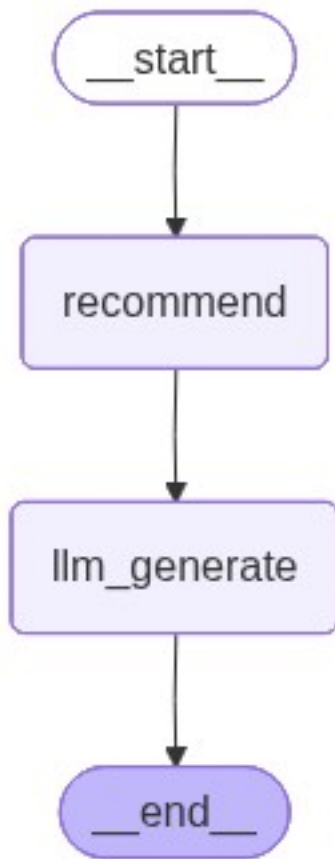
from langgraph.graph import StateGraph, START, END
builder = StateGraph(RecState)

builder.add_node("recommend", recommend_node)
builder.add_node("llm_generate", llm_node)

builder.add_edge(START, "recommend")
builder.add_edge("recommend", "llm_generate")
builder.add_edge("llm_generate", END)

rec_graph = builder.compile()
rec_graph

```



```
initial_state = {
    "profile": "I want to learn machine learning and AI",
    "completed_ids": []
}

final_state = rec_graph.invoke(initial_state)
print("LLM Answer:\n", final_state["answer"])

initial_state_2 = {
    "profile": "I know Azure basics and want to manage containers and
build CI/CD pipelines.",
    "completed_ids": []
}

final_state = rec_graph.invoke(initial_state_2)
print("LLM Answer:\n", final_state["answer"])

LLM Answer:
Based on your profile and your interest in managing containers and
building CI/CD pipelines in Azure, here are the top 5 course
recommendations:
```

1. ****C008 - DevOps Practices and CI/CD**** (similarity: 0.9916)
 - This course will help you understand the core principles of DevOps and how to implement CI/CD pipelines effectively.
2. ****C009 - Containerization with Docker and Kubernetes**** (similarity: 1.0585)
 - Learn about containerization technologies and how to manage containers using Docker and orchestrate them with Kubernetes.
3. ****C007 - Cloud Computing with Azure**** (similarity: 0.9313)
 - A foundational course that reinforces your Azure knowledge, which can be beneficial as you start managing containers and pipelines in the Azure ecosystem.
4. ****C006 - Data Engineering on AWS**** (similarity: 1.2221)
 - While this course focuses on AWS, understanding data engineering concepts can be important for creating robust pipelines, although it may have less direct relevance to Azure.
5. ****C025 - MLOps: Productionizing Machine Learning**** (similarity: 1.2723)
 - If you're interested in integrating machine learning into your CI/CD processes, this course can provide useful insights, though it may not be directly related to container management.

These courses should provide you with a comprehensive foundation and the advanced skills necessary to manage containers and build CI/CD pipelines effectively.

```
initial_state_3 = {  
    "profile": "My background is in ML fundamentals; I'd like to  
specialize in neural networks and production workflows.",  
    "completed_ids": []  
}  
final_state = rec_graph.invoke(initial_state_3)  
  
print("LLM Answer:\n", final_state["answer"])
```

LLM Answer:

Based on your background in machine learning fundamentals and your interest in specializing in neural networks and production workflows, here are the top 5 course recommendations for you:

1. ****C002 - Deep Learning with TensorFlow and Keras**** (similarity: 1.2304)
 - This course will provide you with hands-on experience in building and training deep neural networks using popular frameworks.
2. ****C025 - MLOps: Productionizing Machine Learning**** (similarity: 1.0598)

- This course focuses on the best practices for deploying and maintaining machine learning models in production environments.

3. ****C001 - Foundations of Machine Learning**** (similarity: 1.1405)

- A solid review of machine learning principles, which will help strengthen your understanding as you move into more specialized areas.

4. ****C003 - Natural Language Processing Fundamentals**** (similarity: 1.2559)

- This course explores applications of neural networks in the realm of NLP, deepening your understanding of neural network applications.

5. ****C005 - Reinforcement Learning Basics**** (similarity: 1.2807)

- An introductory course on reinforcement learning, which utilizes neural networks in various ways; a good expansion to your neural network knowledge.

These courses will help you deepen your expertise in neural networks while also covering production workflows effectively.

```
initial_state_4 = {  
    "profile": "I want to learn to build and deploy microservices with  
Kubernetes—what courses fit best?",  
    "completed_ids": []  
}  
final_state = rec_graph.invoke(initial_state_4)  
  
print("LLM Answer:\n", final_state["answer"])
```

LLM Answer:

Based on your interest in learning to build and deploy microservices with Kubernetes, here are the top 5 course recommendations:

1. ****C010 - APIs and Microservices Architecture**** (similarity: 1.0504)

- This course focuses on the principles of microservices architecture, which is crucial for building scalable applications.

2. ****C009 - Containerization with Docker and Kubernetes**** (similarity: 0.7764)

- This course will teach you about containerization using Docker and how to orchestrate those containers with Kubernetes, aligning perfectly with your goals.

3. ****C008 - DevOps Practices and CI/CD**** (similarity: 1.1419)

- Understanding DevOps practices and Continuous Integration/Continuous Deployment (CI/CD) is essential for effectively managing microservices, making this course highly relevant.

4. ****C007 - Cloud Computing with Azure**** (similarity: 1.1238)

- This course provides insights into deploying and managing applications in the cloud, which often includes working with

Kubernetes for orchestration.

5. ****C006 - Data Engineering on AWS**** (similarity: 1.1915)
 - While not directly focused on microservices, this course offers important knowledge about cloud infrastructure that can complement your microservices learning.

These courses will provide you with a solid foundation in building and deploying microservices with Kubernetes and related technologies.

```
initial_state_5 = {  
    "profile": "I'm interested in blockchain and smart contracts but  
have no prior experience. Which courses do you suggest?",  
    "completed_ids": []  
}  
final_state = rec_graph.invoke(initial_state_5)  
  
print("LLM Answer:\n", final_state["answer"])
```

LLM Answer:

Based on your interest in blockchain and smart contracts, here are the top 5 course recommendations tailored to your profile:

1. ****Blockchain Technology and Smart Contracts****
 - Overview: This course provides a comprehensive introduction to blockchain technology, focusing on its principles, architecture, and applications. You'll also learn about smart contracts and how they function on blockchain platforms.
2. ****Cybersecurity Fundamentals****
 - Overview: Understanding cybersecurity is essential in the blockchain space. This course covers fundamental concepts of cybersecurity that will help you comprehend the security implications of blockchain applications.
3. ****NoSQL Databases and MongoDB****
 - Overview: While not directly related to blockchain, this course focuses on NoSQL databases, which are often used in conjunction with blockchain technology. Understanding database management can give you an edge in smart contract development.
4. ****Internet of Things (IoT) Development****
 - Overview: IoT and blockchain often intersect. This course will provide you with insights into how IoT devices can leverage blockchain technology, especially for data security and smart contract applications.
5. ****Augmented and Virtual Reality Development****
 - Overview: Although more tangential, this course explores immersive technologies. Understanding how these technologies can

interact with blockchain may provide creative ideas for applications in various fields, including gaming and simulations.

These courses will give you a solid foundation in the key areas related to blockchain and smart contracts while enhancing your understanding of interconnected technologies.