```
# Course Recommendation Engine
# Assignment 2: Personalized Course Recommendation Engine

# Step 1: Install required packages
!pip install openai pandas numpy scikit-learn chromadb --quiet
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 67.3/67.3 kB 3.0 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 19.3/19.3 MB 97.8 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 94.9/94.9 kB 7.3 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 284.2/284.2 kB 20.4 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.9/1.9 MB 69.2 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 101.6/101.6 kB 7.7 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 16.4/16.4 MB 71.0 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 65.8/65.8 kB 5.6 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 55.7/55.7 kB 4.1 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 196.2/196.2 kB 14.5 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 118.5/118.5 kB 8.8 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 102.5/102.5 kB 7.6 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 459.8/459.8 kB 25.8 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 71.5/71.5 kB 4.8 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.0/4.0 MB 66.5 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 454.8/454.8 kB 26.4 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 46.0/46.0 kB 3.3 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 86.8/86.8 kB 6.6 MB/s eta 0:00:00
  Building wheel for pypika (pyproject.toml) ... done
```

```
# Step 2: Import libraries
import pandas as pd
import numpy as np
from typing import List, Tuple
import openai
from openai import AzureOpenAI
import chromadb
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.metrics.pairwise import cosine_similarity


# Step 3: Setup Azure OpenAI
# Replace with your actual Azure OpenAI details
client = AzureOpenAI(
    azure_endpoint="https://user4-mbqm2gg0-westeurope.cognitiveservices.azure.com/",
    api_key="BNDYOKM2ha3JgNR8OOBmdpTq9ePMI7vWQpAHL6XA1sx4HNXlrdYpJQQJ99BFAC5RqLJXJ3w3AAAAACOGNrWw",
    api_version="2024-02-01"
)


# Step 4: Load the dataset
url = "https://raw.githubusercontent.com/Bluedata-Consulting/GAAPB01-training-code-base/refs/heads/main/Assignments/as
df = pd.read_csv(url)
print(f"Dataset loaded: {df.shape[0]} courses")
print(df.head())
```

```
Dataset loaded: 25 courses
  course_id                                     title  \
0      C001           Foundations of Machine Learning   
1      C002   Deep Learning with TensorFlow and Keras   
2      C003  Natural Language Processing Fundamentals   
3      C004      Computer Vision and Image Processing   
4      C005             Reinforcement Learning Basics   

                                         description  
0  Understand foundational machine learning algor...  
1  Explore neural network architectures using Ten...  
2  Dive into NLP techniques for processing and un...  
3  Learn the principles of computer vision and im...  
4  Get introduced to reinforcement learning parad...  
```

```
# Step 5: Create embeddings function
def get_embedding(text):
    """Get embedding for a text using Azure OpenAI"""
    response = client.embeddings.create(
        input=text,
        model="text-embedding-ada-002",  # Your deployment name
        model="text-embedding-ada-002"  # Your deployment name
    )
    return response.data[0].embedding


# Step 6: Create embeddings for all courses
print("Creating embeddings for courses...")
course_embeddings = []
for index, row in df.iterrows():
    # Combine title and description
    course_text = f"{row['title']}. {row['description']}"
    embedding = get_embedding(course_text)
    course_embeddings.append(embedding)
    print(f"Processed course {index + 1}/{len(df)}")
```

```
Creating embeddings for courses...
Processed course 1/25
Processed course 2/25
Processed course 3/25
Processed course 4/25
Processed course 5/25
Processed course 6/25
Processed course 7/25
Processed course 8/25
Processed course 9/25
Processed course 10/25
Processed course 11/25
Processed course 12/25
Processed course 13/25
Processed course 14/25
Processed course 15/25
Processed course 16/25
Processed course 17/25
Processed course 18/25
Processed course 19/25
Processed course 20/25
Processed course 21/25
Processed course 22/25
Processed course 23/25
Processed course 23/25
Processed course 24/25
Processed course 25/25
```

```
# Convert to numpy array
course_embeddings = np.array(course_embeddings)
print(f"Created embeddings for {len(course_embeddings)} courses")
```

```
Created embeddings for 25 courses
```

```
# Step 7: Main recommendation function
def recommend_courses(profile: str, completed_ids: List[str]) -> List[Tuple[str, float]]:
    """
    Returns a list of (course_id, similarity_score) for the top-5 recommendations.
    """
    # Get embedding for user profile
    profile_embedding = get_embedding(profile)
    profile_embedding = np.array(profile_embedding).reshape(1, -1)

    # Calculate cosine similarity
    similarities = cosine_similarity(profile_embedding, course_embeddings)[0]

    # Create list of course_id and similarity pairs
    recommendations = []
    for i, similarity in enumerate(similarities):
        course_id = df.iloc[i]['course_id']
        if course_id not in completed_ids:  # Skip completed courses
            recommendations.append((course_id, similarity))

    # Sort by similarity and return top 5
    recommendations.sort(key=lambda x: x[1], reverse=True)
    return recommendations[:5]
```

```
# Step 8: Test with sample queries
test_profiles = [
    {
        "query": "I've completed the 'Python Programming for Data Science' course and enjoy data visualization. What s
        "completed": ["python-data-science"]
    },
    {
        "query": "I know Azure basics and want to manage containers and build CI/CD pipelines. Recommend courses.",
        "completed": []
    },
    {
        "query": "My background is in ML fundamentals; I'd like to specialize in neural networks and production workfl
        "completed": []
    },
    {
        "query": "I want to learn to build and deploy microservices with Kubernetes—what courses fit best?",
        "completed": []
    },
    {
        "query": "I'm interested in blockchain and smart contracts but have no prior experience. Which courses do you
        "completed": []
    }
]


# Step 9: Run tests and show results
print("\n" + "="*80)
print("COURSE RECOMMENDATION RESULTS")
print("="*80)

for i, test in enumerate(test_profiles, 1):
    print(f"\nTest {i}:")
    print(f"Query: {test['query']}")
    print(f"Completed courses: {test['completed']}")
    print("\nRecommendations:")

    # Get recommendations
    recommendations = recommend_courses(test['query'], test['completed'])

    # Display results
```

```
for j, (course_id, score) in enumerate(recommendations, 1):
    # Find course details
    course_row = df[df['course_id'] == course_id].iloc[0]
    print(f"{j}. Course ID: {course_id}")
    print(f"   Title: {course_row['title']}")
    print(f"   Description: {course_row['description'][:100]}...")
    print(f"   Similarity Score: {score:.4f}")
    print("-" * 60)

print("\n" + "="*80)
```

```
================================================================================
COURSE RECOMMENDATION RESULTS
================================================================================

Test 1:
Query: I've completed the 'Python Programming for Data Science' course and enjoy data visualization. What should I
Completed courses: ['python-data-science']

Recommendations:
1. Course ID: C016
   Title: Python Programming for Data Science
   Description: Learn Python fundamentals for data science: variables, control flow, functions, and object-oriente
   Similarity Score: 0.8383
------------------------------------------------------------
2. Course ID: C011
   Title: Big Data Analytics with Spark
   Description: Process and analyze large datasets using Apache Spark and PySpark. The course covers RDDs, DataFra
   Similarity Score: 0.8111
------------------------------------------------------------
3. Course ID: C014
   Title: Data Visualization with Tableau
   Description: Transform raw data into compelling visual stories using Tableau. Learn to connect to diverse data
   Similarity Score: 0.8079
------------------------------------------------------------
4. Course ID: C017
   Title: R Programming and Statistical Analysis
   Description: Get introduced to R for statistical computing and graphics. Topics include data structures, contro
   Similarity Score: 0.7963
------------------------------------------------------------
5. Course ID: C004
   Title: Computer Vision and Image Processing
   Description: Learn the principles of computer vision and image processing. Topics include filtering, edge detec
   Similarity Score: 0.7899
------------------------------------------------------------

================================================================================

Test 2:
Query: I know Azure basics and want to manage containers and build CI/CD pipelines. Recommend courses.
Completed courses: []

Recommendations:
1. Course ID: C007
   Title: Cloud Computing with Azure
   Description: Master Microsoft Azure’s core services: virtual machines, Azure Functions, Azure SQL Database, Cos
   Similarity Score: 0.8476
------------------------------------------------------------
2. Course ID: C009
   Title: Containerization with Docker and Kubernetes
   Description: Learn container fundamentals with Docker: images, containers, and Compose. Then advance to Kuberne
   Similarity Score: 0.8329
------------------------------------------------------------
3. Course ID: C008
   Title: DevOps Practices and CI/CD
   Description: Adopt DevOps methodologies to accelerate software delivery. Explore version control with Git, cont
   Similarity Score: 0.8253
------------------------------------------------------------
```

Start coding or generate with AI.