KỊCH BẢN DEMO: MINI QA AGENT – CHỊ PHỤNG

Đây là kịch bản demo hoàn chỉnh cho Chị Phụng trình bày Mini QA Agent trong Workshop: AI Agents In Practice. Demo này sử dụng dữ liệu admission\_faq\_large.csv (300 câu hỏi) và một pipeline Agent đơn giản để trả lời câu hỏi tự nhiên.

# Tổng thời gian: 20 phút

1. Giới thiệu & đặt vấn đề – 3 phút  
2. Show pipeline & kiến trúc – 3 phút  
3. Demo code từng bước – 12 phút  
4. Tổng kết & gợi mở – 2 phút

# Chi tiết kịch bản

## 1. Giới thiệu & đặt vấn đề (3 phút)

Sau khi chúng ta vừa được xem hệ thống Agentic Chatbot phức tạp do nhóm nghiên cứu xây dựng, tôi sẽ giới thiệu một phiên bản thu gọn hơn – một Mini QA Agent. Hệ thống này có thể nhận câu hỏi tự nhiên từ người dùng, tìm kiếm thông tin tuyển sinh từ một file dữ liệu với hơn 300 câu hỏi, và trả lời chính xác chỉ trong vài giây.

Mục tiêu là giúp mọi người hình dung cách hoạt động cơ bản của một Agentic AI: từ việc nhận câu hỏi, tìm kiếm dữ liệu, đến tổng hợp và phản hồi.

## 2. Show pipeline & kiến trúc (3 phút)

Hiển thị slide sơ đồ pipeline đơn giản:  
Người dùng → Embedding → Tìm kiếm semantic → Gọi LLM → Trả lời

Giải thích: Pipeline gồm 4 bước chính – người dùng nhập câu hỏi, hệ thống embedding câu hỏi và tìm kiếm ngữ nghĩa trong dữ liệu tuyển sinh, kết quả tìm kiếm được gửi tới LLM để tổng hợp câu trả lời, và cuối cùng trả về câu trả lời tự nhiên.

## 3. Demo code từng bước (12 phút)

### Bước 1: Load dữ liệu & tạo vector store (3 phút)

Giải thích: Bắt đầu bằng cách tải file dữ liệu admission\_faq\_large.csv, sau đó tạo vector store để tìm kiếm nhanh chóng.  
  
Code:  
from langchain.vectorstores import Chroma  
from langchain.embeddings import OpenAIEmbeddings  
from langchain.document\_loaders import CSVLoader  
  
# Load dữ liệu từ CSV  
loader = CSVLoader(file\_path="admission\_faq\_large.csv")  
data = loader.load()  
  
# Tạo embeddings  
embeddings = OpenAIEmbeddings()  
  
# Tạo vector store  
db = Chroma.from\_documents(data, embeddings, persist\_directory="db")  
print("Vector store đã sẵn sàng.")

### Bước 2: Viết hàm trả lời câu hỏi (4 phút)

Giải thích: Xây dựng agent để nhận câu hỏi từ người dùng, tìm kiếm trong vector store và gọi LLM để tổng hợp câu trả lời.  
  
Code:  
from langchain.chains import RetrievalQA  
from langchain.llms import OpenAI  
  
# Khởi tạo LLM  
llm = OpenAI(model="gpt-3.5-turbo", temperature=0)  
  
# Tạo QA chain  
qa\_chain = RetrievalQA.from\_chain\_type(  
 llm=llm,  
 retriever=db.as\_retriever(),  
 return\_source\_documents=True  
)  
  
# Hàm trả lời  
def ask\_agent(question):  
 result = qa\_chain.run(question)  
 print(f"Q: {question}")  
 print(f"A: {result}")

### Bước 3: Chạy thử trực tiếp (5 phút)

Giải thích: Thử hỏi agent một số câu hỏi thực tế từ phụ huynh và học sinh.  
  
Code:  
ask\_agent("Học phí của chương trình tiên tiến là bao nhiêu?")  
ask\_agent("Thời gian đào tạo của chương trình kỹ sư?")  
ask\_agent("Trường có hỗ trợ học bổng toàn phần không?")

## 4. Tổng kết & gợi mở (2 phút)

Mini QA Agent này là một phiên bản đơn giản của Agentic AI, nhưng đã minh họa rõ ràng cách một agent phối hợp các thành phần: nhận thức, tìm kiếm thông tin và reasoning. Phần tiếp theo, các đội sẽ thực hành xây dựng một agent tương tự dựa trên code này để củng cố kiến thức.