# Chatbot Tư Vấn Tâm Lý Quân Nhân

Hệ thống chatbot tư vấn tâm lý cho quân nhân trong quân đội dựa trên kỹ thuật RAG (Retrieval Augmented Generation).

## Yêu cầu hệ thống

* Python 3.10.11
* Groq API key
* Kết nối internet để tải mô hình và gọi API

## Tính năng

* Tư vấn tâm lý cho quân nhân dựa trên cơ sở dữ liệu tâm lý quân sự
* Sử dụng kỹ thuật RAG để tìm kiếm và tạo câu trả lời chính xác
* Giao diện người dùng thân thiện với Streamlit
* Hệ thống logging đầy đủ
* Lưu trữ lịch sử hội thoại theo phiên (session)
* Hỗ trợ streaming response từ LLM

## Cấu trúc dự án

nguyenvietong\_chatbot/

├── data/ # Thư mục chứa dữ liệu

│ └── military\_psychology.csv

├── logs/ # Thư mục chứa log

├── history/ # Thư mục lưu trữ lịch sử hội thoại

├── src/ # Mã nguồn

│ ├── \_\_init\_\_.py

│ ├── config.py # Cấu hình

│ ├── logger.py # Hệ thống logging

│ ├── data\_processor.py # Xử lý dữ liệu

│ ├── embedding\_system.py # Hệ thống embedding

│ ├── llm\_system.py # Tương tác với LLM

│ ├── rag\_system.py # Hệ thống RAG

│ ├── chatbot.py # Chatbot

│ ├── memory\_system.py # Hệ thống quản lý bộ nhớ hội thoại

│ ├── history\_manager.py # Quản lý lịch sử hội thoại

│ ├── database\_setup.py # Thiết lập vector database

│ └── streamlit\_app.py # Ứng dụng Streamlit

├── vector\_db/ # Vector database (tạo tự động)

├── .env # Biến môi trường (cần tạo từ .env.example)

├── .env.example # Mẫu biến môi trường

├── main.py # File chạy chính

└── requirements.txt # Các thư viện cần thiết

## Cài đặt

### 1. Cài đặt Python 3.10.11

Tải và cài đặt Python 3.10.11 từ trang chủ Python:

* Windows: <https://www.python.org/downloads/release/python-31011/>
* Linux/MacOS: Sử dụng pyenv hoặc asdf để quản lý phiên bản Python

### 2. Clone repository:

git clone <repository-url>

cd military\_psychology\_chatbot

### 3. Tạo và kích hoạt môi trường ảo:

**Windows:**

python -m venv venv

.\venv\Scripts\activate

**Linux/MacOS:**

python -m venv venv

source venv/bin/activate

### 4. Cài đặt các thư viện cần thiết:

pip install -r requirements.txt

### 5. Lấy API key từ Groq:

1. Đăng ký tài khoản tại [Groq Console](https://console.groq.com)
2. Tạo API key mới từ mục "API Keys"
3. Sao chép API key để sử dụng trong bước tiếp theo

### 6. Tạo file .env:

Tạo file .env trong thư mục gốc của dự án với nội dung:

GROQ\_API\_KEY=your\_groq\_api\_key\_here

Thay your\_groq\_api\_key\_here bằng API key đã lấy từ Groq.

## Sử dụng

### 1. Khởi tạo vector database

Trước khi sử dụng chatbot, cần khởi tạo vector database từ dữ liệu trong thư mục data/:

python main.py --setup-db

Quá trình này sẽ:

* Tải mô hình embedding Alibaba-NLP/gte-multilingual-base
* Xử lý dữ liệu từ file CSV
* Tạo vector embeddings cho dữ liệu
* Lưu vector database vào thư mục vector\_db/

### 2. Khởi động ứng dụng

python main.py --run-app

Sau khi chạy lệnh trên, ứng dụng Streamlit sẽ được khởi động và có thể truy cập qua trình duyệt web tại địa chỉ http://localhost:8501.

## Luồng hoạt động

Chatbot hoạt động theo mô hình RAG (Retrieval Augmented Generation) với các bước chính:

1. **Tiếp nhận câu hỏi**: Người dùng nhập câu hỏi qua giao diện Streamlit
2. **Tìm kiếm thông tin liên quan**:
   * Chuyển đổi câu hỏi thành vector embedding
   * Tìm kiếm các đoạn văn bản tương tự trong vector database (FAISS)
   * Trích xuất thông tin liên quan nhất
3. **Tạo câu trả lời**:
   * Kết hợp câu hỏi, thông tin tìm được và lịch sử hội thoại
   * Gửi đến LLM (llama-3.3-70b-versatile) qua Groq API
   * Nhận câu trả lời từ LLM
4. **Hiển thị kết quả**: Hiển thị câu trả lời dưới dạng stream (từng phần) lên giao diện
5. **Lưu trữ hội thoại**:
   * Lưu câu hỏi và câu trả lời vào bộ nhớ để duy trì ngữ cảnh hội thoại
   * Lưu vào file CSV trong thư mục history/ để tham khảo sau này

## Công nghệ sử dụng

* **Embedding Model**: Alibaba-NLP/gte-multilingual-base
  + Mô hình đa ngôn ngữ chất lượng cao để chuyển đổi văn bản thành vector
* **Vector Database**: FAISS
  + Thư viện tìm kiếm tương tự hiệu quả từ Facebook AI Research
* **LLM**: llama-3.3-70b-versatile (qua Groq API)
  + Mô hình ngôn ngữ lớn từ Meta, cung cấp qua Groq để tối ưu tốc độ
* **Framework RAG**: LangChain
  + Framework tích hợp các thành phần của hệ thống RAG
* **Quản lý luồng**: LangGraph
  + Xây dựng và quản lý các luồng xử lý phức tạp
* **Giao diện**: Streamlit
  + Framework Python để tạo ứng dụng web đơn giản và hiệu quả
* **Logging**: Loguru
  + Thư viện logging Python hiện đại và linh hoạt
* **Quản lý bộ nhớ hội thoại**: LangChain Memory
  + Duy trì ngữ cảnh hội thoại giữa người dùng và chatbot

## Tính năng nâng cao

* **Streaming Response**: Hiển thị câu trả lời theo thời gian thực, từng phần một
* **Quản lý phiên (Session)**: Mỗi cuộc hội thoại được gán một ID phiên duy nhất
* **Lưu trữ lịch sử**: Tự động lưu lịch sử hội thoại vào file CSV theo ngày
* **Bộ nhớ hội thoại**: Duy trì ngữ cảnh của 5 tin nhắn gần nhất để tạo câu trả lời liên quan

## Lưu ý

* Đảm bảo đã cài đặt đúng phiên bản Python 3.10.11
* Cần có API key của Groq để sử dụng mô hình LLM
* Vector database cần được khởi tạo trước khi sử dụng chatbot
* Các thư mục logs/ và history/ sẽ được tạo tự động nếu chưa tồn tại