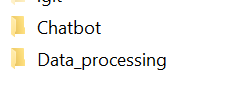
# **TÀI LIỆU CHATBOT TRÍCH XUẤT THÔNG TIN TỪ KẺ LỪA ĐẢO**

## **I. CẤU TRÚC CODE TỔNG QUÁT**



|  |  |
| --- | --- |
| Folder Data\_processing: Toàn bộ code xử lý dữ liệu | Folder Chatbot: toàn bộ code xây |

## **II. CÁCH CHẠY CHATBOT**

**Bước 1: Cài đặt**

**Python 3.12**

**Docker Desktop**

**Nodejs**

**Anaconda => tạo môi trg <create -n luadao\_env python=3.12>**

**Bước 2:** Mở Visual Studio Code -> Mở folder Chatbot -> Mở terminal

* Nếu có anaconda thì activate môi trường (không thì thôi)
* Tải các thư viện cần thiết

*pip install -r requirements.txt*

* Cài đặt openai\_key

**setx OPENAI\_API\_KEY “điền key nè”**

**Bước 3: Cài đặt Vector database Qdrant**

**Mở terminal:**

docker pull qdrant/qdrant

pip install qdrant-client

docker run -p 6333:6333 -v /absolute/path/here/qdrant\_storage:/qdrant/storage qdrant/qdrant

**Bước 4: Nhúng vector và lưu vào vector database Qdrant**

**Mở terminal**

python EmbeddingVectorManager.py

**Bước 5: Chạy chabot**

* chạy api

uvicorn main:app --reload

* chạy frontend

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

ĐÂY LÀ CHẠY VỚI GIAO DIỆN javascript

cd front-end

npm install

npm run dev

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

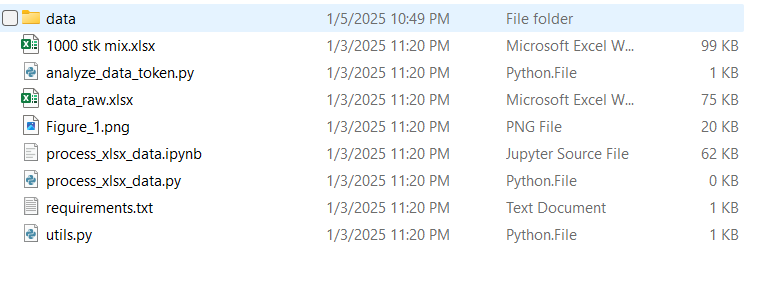
ĐÂY LÀ CHẠY VỚI GIAO DIỆN STREAMLIT

streamlit run streamlit\_app.py

## **III. CHI TIẾT CODE**

**1. Xử lý dữ liệu**

* Cấu trúc code



* data: folder chứa dữ liệu sau khi xử lý
* analyze\_data\_token.py -> phân tích độ dài các mẫu dữ liệu theo token
* process\_xlsx\_data.py -> file code đọc dữ liệu từ xlsx
* requirements.txt -> thư viện cần cài đặt và phiên bản
* utils.py -> các hàm xử lý file json (đọc, lưu)
* Cách chạy code:

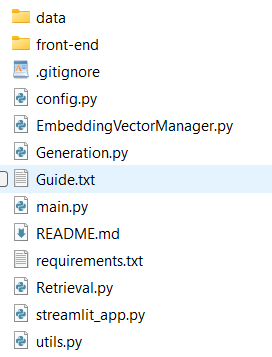
python process\_xlsx\_data.py

-> xử lý data -> lưu json vào folder processed

Python analyze\_data\_token.py

-> hiện ra biểu đồ phân tích độ dài các mẫu dữ liệu

**2. Build chatbot**

* Cấu trúc code:  
  
* Folder data: chứa các dữ liệu

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

data\_hoithoai.json: các cặp hội thoại từ dữ liệu thu thập đã được xử lý

data\_information.xlsx: thông tin cá nhân

image: folder lưu các ảnh như ảnh bill, cccd

* config.py : các tham số cài đặt chung
* data\_hoithoai.json -> data đã xử lý được lấy từ bên dữ liệu
* EmbeddingVectorManager.py -> code tạo vector và lưu vào vector DB Qdrant
* Generation.py -> code xây dựng hội thoại, trích xuất thông tin
* README.md
* requirements.txt -> lưu các thư viện và version cần cài đặt
* Retrieval.py -> code truy xuất các dữ liệu liên quan tới prompt của user
* streamlit\_app.py -> code build giao diện với streamlit
* front-end -> code buil giao diện với javascript
* utils.py -> các hàm cần thiết như làm sạch text, tải model nhúng vector

LƯU Ý; file config.py là file chứa các tham số tinh chỉnh