Lý do chọn Rasa thay vì chatgpt, việc sử lấy api của gpt xong tích hợp vào unity thì đó không khác mấy việc đem câu hỏi lên hỏi chatgpt sau đó lấy câu trả lời và đem vào trong unity, vì lượng data train gpt đã vô cùng lớn, nên với lượng data không đủ nhiều của mình khi cho vào thì sẽ không thay đổi được phần nào kết quả hiển thị của chatgpt, vì thế việc sử dụng rasa sẽ giúp mình train AI từ đầu, rasa sẽ hiển thị kết quả gần như là theo những gì mình mong muốn, và điều đó phụ thuộc khá nhiều vào độ lớn của data

* Sinh ra 2 file .yml, 1 file là về phía khách hàng, 1 file là về phía của nhân viên AI (git)

Clone 1 môi trường ảo cho project

Install thư viện và rasa ( zalo)

Khởi tạo rasa Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

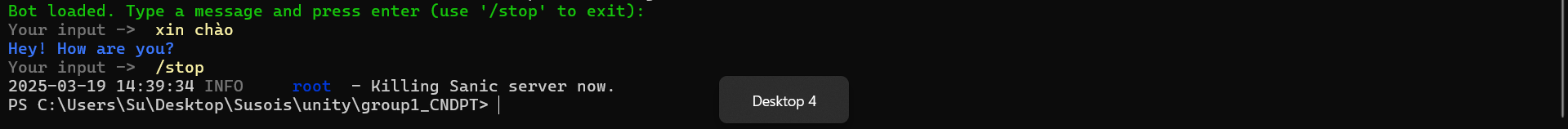
Sau đó sẽ tạo ra folder Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

Lúc này rasa sẽ tạo cho mình 1 số mô hình train cơ bản, gần như là mặc định của rasa

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

Nội dung do AI tạo ra có thể không chính xác.

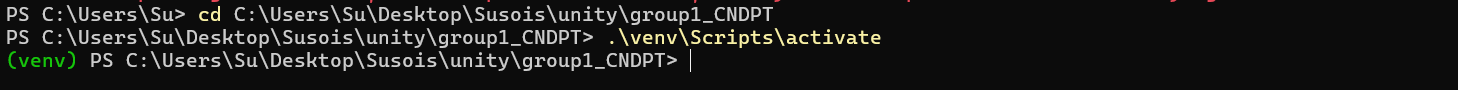
Sau khi Ai train xong mặc định thì mình có thể test trước xem đã cài thành công chưa 

* Tạo data để train AI, đầu tiên trong file NLU (Natural Language Understanding),

**Xây dựng danh sách intent liên quan**:

* **customer\_complaint** (Phàn nàn của khách hàng)
* **price\_negotiation** (Mặc cả giá)
* **product\_quality** (Chất lượng sản phẩm)
* **refund\_request** (Yêu cầu hoàn tiền)
* **customer\_support** (Hỗ trợ khách hàng)
* **promotion\_inquiry** (Hỏi về khuyến mãi)
* **stock\_availability** (Tình trạng hàng hóa)
* **discount\_request** (Yêu cầu giảm giá)
* **price\_comparison** (So sánh giá)
* **bulk\_purchase\_inquiry** (Hỏi về mua sỉ/số lượng lớn)
* **payment\_method** (Phương thức thanh toán)
* **product\_authenticity** (Xác thực sản phẩm)
* **warranty\_inquiry** (Hỏi về bảo hành)
* **freebie\_request** (Yêu cầu tặng kèm)
* **order\_cancellation** (Hủy đơn hàng)
* **product\_usage** (Hỏi về cách sử dụng sản phẩm)
* **competitor\_inquiry** (Hỏi về đối thủ)
* **greeting** (Chào hỏi)
* **farewell** (Tạm biệt)
* **health\_inquiry** (Hỏi thăm sức khỏe)
* **weather\_talk** (Nói về thời tiết)
* **social\_issue** (Vấn đề xã hội)
* **small\_talk** (Nói chuyện phiếm)
* **personal\_connection** (Kết nối cá nhân)

vì không có nhiều thời gian tạo data và không mấy khách hàng cũng như nhãn hàng sẵn sàng bỏ thời gian để giúp mình thu thập dữ diệu cho nên mình sử dụng grok để sinh dữ liệu, tại sao lại sử dụng con này vì đây là con AI theo mình thấy nó sử dụng ngôn ngữ tự nhiên linh hoạt nhất, ít bị cản trở về ngôn ngữ, không có nhiều từ khóa bị chặn, gần gũi, phù hợp với tác phong, giọng nói, ngôn ngữ tự nhiên của người Việt Nam

sau khi sinh xong data thì bắt đầu khởi chạy môi trường ảo venv để bắt đầu train Rasa 

tiếp theo train AI

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

Nội dung do AI tạo ra có thể không chính xác.

Sau khi train AI xong nó sẽ sinh ra 1 file .tar.gz trong models

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

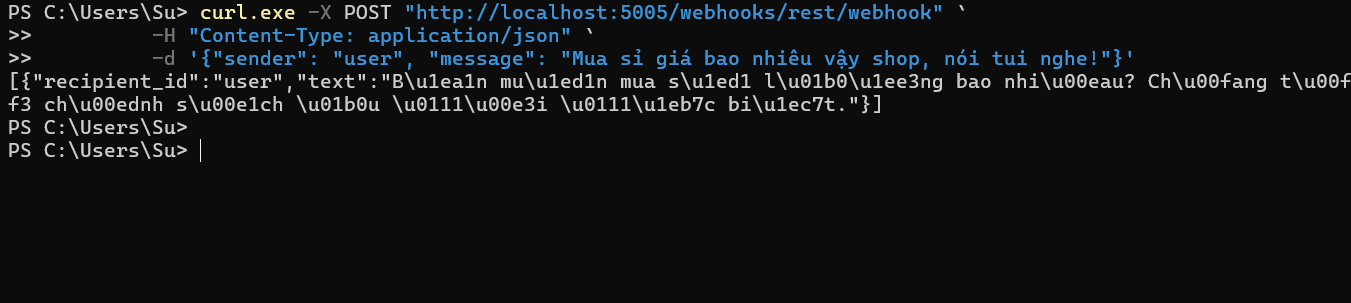
Nội dung do AI tạo ra có thể không chính xác.

Bước tiếp cần phải khởi chạy Rasa trên local

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

Nội dung do AI tạo ra có thể không chính xác.

Sau khi server rasa chạy thành công thì gửi thử tin tới rasa xem có phản hồi lại không



Ta nhận thấy rasa đã có phản hổi lại tuy nhiên là do chưa cài utf-8 nên ta nhận được dòng như kia

Sau đó ta tích hợp rasa vào trong unity với các 3 file C# là RasaService.cs, DialogData.cs và DialogManager.cs , gửi phản hồi từ unity tới rasa ta nhận được phản hồi của rasa đúng với định dạng utf-8

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

Nội dung do AI tạo ra có thể không chính xác.