

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG



ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

Sinh viên : Phạm Thế Duyệt

Giảng viên hướng dẫn: ThS. Vũ Trọng Chiến

Hải Phòng - 2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

TÍCH HỢP GEMINI VÀO WEBCCHAT HỖ TRỢ
TUYỂN SINH CHO ĐẠI HỌC HPU

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
NGÀNH CÔNG NGHỆ THÔNG TIN

Sinh viên thực hiện : Phạm Thế Duyệt
Giảng viên hướng dẫn: ThS. Vũ Trọng Chiến

Hải Phòng - 2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên : Phạm Thế Duyệt

Mã SV: 2212111001

Lớp : CT2601

Ngành : Công nghệ thông tin

Tên đề tài : Tích hợp Gemini vào webchat hỗ trợ tuyển sinh
cho Đại học HPU

.CÁC CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Họ và tên : Vũ Trọng Chiến

Học hàm, học vị : Thạc sĩ

Cơ quan công tác : Trường đại học Quản lý và Công nghệ Hải Phòng

Nội dung hướng dẫn : Tích hợp Gemini vào webchat hỗ trợ tuyển sinh cho Đại học HPU

Đề tài tốt nghiệp được giao ngày 06 tháng 10 năm 2025

Yêu cầu phải hoàn thành xong trước ngày 27 tháng 12 năm 2025

Đã nhận nhiệm vụ ĐTTN

Sinh viên

Đã giao nhiệm vụ ĐTTN

Giảng viên hướng dẫn

Hải Phòng, ngày tháng 12 năm 2025

TRƯỞNG KHOA

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN TỐT NGHIỆP

Họ và tên giảng viên: Vũ Trọng Chiến

Đơn vị công tác : Trường đại học Quản lý và Công nghệ Hải Phòng

Họ và tên sinh viên : Phạm Thế Duyệt **Ngành:** Công nghệ Thông tin

Nội dung hướng dẫn : Tích hợp Gemini vào webchat hỗ trợ tuyển sinh cho Đại học HPU

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp

Sinh viên có thái độ học tập nghiêm túc, chủ động và có tinh thần trách nhiệm cao trong suốt quá trình thực hiện đề tài. Thường xuyên trao đổi với giảng viên hướng dẫn, tiếp thu ý kiến góp ý và điều chỉnh nội dung nghiên cứu kịp thời. Sinh viên thể hiện năng lực tự học, khả năng nghiên cứu và triển khai thực tế tốt.

Đặc biệt, nghiên cứu đã được đăng ký đề tài nghiên cứu khoa học cấp Khoa và bảo vệ thành công; Nghiên cứu này cũng được báo cáo tại Chương trình “Hải Phòng – Café Khởi nghiệp Sáng tạo – Trạm số 2” do Trường Đại học Quản lý và Công nghệ Hải Phòng phối hợp với Sở Khoa học và Công nghệ thành phố Hải Phòng tổ chức, cho thấy tính nghiêm túc và khả năng trình bày, bảo vệ ý tưởng của sinh viên.

2. Đánh giá chất lượng của đồ án/khóa luận (so với nội dung yêu cầu đã đề ra trong nhiệm vụ Đ.T.T.N, trên các mặt lý luận, thực tiễn, tính toán số liệu...)

Đồ án đáp ứng đầy đủ mục tiêu và yêu cầu đặt ra trong nhiệm vụ đồ án tốt nghiệp. Nội dung báo cáo có cơ sở lý luận rõ ràng, lựa chọn công nghệ phù hợp, quy trình thiết kế và triển khai hệ thống hợp lý. Sản phẩm có tính ứng dụng thực tiễn cao,

có thể triển khai thử nghiệm trong công tác tư vấn tuyển sinh của nhà trường. Báo cáo được trình bày khoa học, logic, có minh họa và phân tích kết quả thử nghiệm.

3. Ý kiến của giảng viên hướng dẫn tốt nghiệp

Được bảo vệ

☐

Không được bảo vệ

☐

Điểm hướng dẫn

☐

Hải Phòng, ngày tháng năm 202...

Giảng viên hướng dẫn

(ký và ghi rõ họ tên)

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN CHẤM PHẢN BIỆN

Họ và tên giảng viên: ThS. Vũ Phạm Minh Hiền

Đơn vị công tác: Trường Đại học Quản lý và Công nghệ Hải Phòng

Họ và tên sinh viên: Phạm Thế Duyệt **Ngành:** Công nghệ Thông tin

Đề tài tốt nghiệp: Tích hợp Gemini vào webchat hỗ trợ tuyển sinh cho Đại học HPU

1. Phần nhận xét của giảng viên chấm phản biện

2. Những mặt còn hạn chế

3. Ý kiến của giảng viên chấm phản biện

Được bảo vệ ☐ Không được bảo vệ ☐ Điểm phản biện ☐

Hải Phòng, ngày.....tháng năm 202...

Giảng viên chấm phản biện
(ký và ghi rõ họ tên)

LỜI CAM ĐOAN

Em xin cam đoan đề tài tốt nghiệp **“Tích hợp Gemini vào webchat hỗ trợ tuyển sinh cho Đại học HPU”** là sản phẩm em tự nghiên cứu. Những số liệu và kết quả nghiên cứu trong đề tài là trung thực.

Các tài liệu tham khảo đã được trích dẫn đầy đủ.

Em xin chịu hoàn toàn trách nhiệm về mọi nội dung của đề tài này.

LỜI CẢM ƠN

Thực hiện đề tài “**Tích hợp Gemini vào webchat hỗ trợ tuyển sinh cho Đại học HPU**”, em đã nhận được nhiều sự quan tâm và hỗ trợ từ Ban Giám hiệu nhà trường, các thầy cô trong Khoa Công nghệ thông tin, cũng như sự tạo điều kiện từ các phòng ban của Trường.

Em xin chân thành gửi lời cảm ơn sâu sắc đến sự hỗ trợ quý báu đó.

Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc và chân thành nhất đến thầy Vũ Trọng Chiến – Giảng viên đã trực tiếp theo sát, chỉ dẫn tận tình và định hướng khoa học trong suốt quá trình thực hiện đề tài. Sự hỗ trợ chuyên môn, động viên kịp thời và những góp ý quý báu của thầy là yếu tố quan trọng giúp em hoàn thành công trình nghiên cứu này.

Mặc dù đã nỗ lực hết sức để hoàn thiện đề tài một cách tốt nhất, song chắc chắn vẫn không thể tránh khỏi những thiếu sót. Em xin ghi nhận và cảm ơn các ý kiến đóng góp từ Quý Thầy Cô trong Hội đồng nghiệm thu – đây sẽ là cơ sở để em tiếp tục cải thiện, phát triển đề tài cũng như nâng cao năng lực nghiên cứu trong tương lai.

Xin chân thành cảm ơn!

MỤC LỤC

DANH MỤC TỪ VIẾT TẮT VÀ THUẬT NGỮ	13
PHẦN MỞ ĐẦU	1
1. Lý do chọn đề tài.....	1
2. Mục tiêu và ý nghĩa nghiên cứu.....	1
3. Phạm vi và phương pháp nghiên cứu.....	2
4. Cấu trúc báo cáo.....	2
5. Tình hình nghiên cứu trong và ngoài nước	2
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT VÀ HIỆN TRẠNG VẤN ĐỀ	4
1.1. Tổng quan về tư vấn tuyển sinh.....	4
1.2. ChatBot và mô hình ngôn ngữ lớn (LLM), xử lý ngôn ngữ tự nhiên (NLP).....	4
1.3. Ứng dụng ChatBot trong giáo dục và tuyển sinh	5
1.4. Hiện trạng hoạt động tuyển sinh tại HPU	6
1.5. Phân tích dữ liệu tuyển sinh từ trang hpu.edu.vn	6
1.6. Kết luận chương.....	8
CHƯƠNG 2: THIẾT KẾ VÀ TRIỂN KHAI CHATBOT	9
2.1. Mục tiêu thiết kế hệ thống	9
2.2. Kiến trúc hệ thống và công nghệ sử dụng	9
2.2.1. Sử dụng API từ Google Gemini.....	10
2.2.2. Sử dụng nền tảng ChatBot tích hợp AI Botpress.....	51
2.2.3. Cơ sở dữ liệu Vector (Vector Database) trong hệ thống Chatbot.....	67
2.3. So sánh tổng quan 2 phương pháp	73
2.4. Giao diện và chức năng ChatBot	76
2.5. Một số kịch bản hội thoại tiêu biểu.....	76
2.6. Kết luận chương.....	76
CHƯƠNG 3: THỬ NGHIỆM, ĐÁNH GIÁ VÀ KẾT LUẬN.....	77
3.1. Kết quả thử nghiệm nội bộ.....	77

3.1.1. Kết quả hiện thực hóa ChatBot.....	77
3.2. Đánh giá hiệu quả ChatBot.....	81
a) Độ chính xác và khả năng phản hồi.....	81
b) Hiệu quả sử dụng bộ lọc đầu vào	82
3.3. So sánh với mục tiêu đề ra	82
3.4. Ưu điểm và hạn chế	82
3.5. Khả năng triển khai	83
KẾT LUẬN VÀ KIẾN NGHỊ.....	84
Kết luận.....	84
Kết luận lựa chọn phương án.....	84
Nhận xét về phương án sử dụng API Google Gemini	85
Nhận xét về phương án sử dụng Botpress kết hợp Knowledge Base.....	87
Lựa chọn cuối cùng.....	88
Kiến nghị.....	89
DANH MỤC TÀI LIỆU THAM KHẢO.....	90

DANH MỤC HÌNH ẢNH

Hình 1: Bộ FAQ thông tin tuyển sinh	7
Hình 2: Bộ FAQ học phí và các khoản đóng góp	8
Hình 3: Sơ đồ phân rã chức năng.....	15
Hình 4: Chào hỏi và bắt đầu hội thoại	16
Hình 5: Form đăng ký tư vấn.....	20
Hình 6: Danh sách người đăng ký	25
Hình 7: Sơ đồ phân rã công việc.....	37
Hình 8: Kiến trúc hệ thống của ChatBot.....	53
Hình 9: Một số kịch bản hội thoại của ChatBot	62
Hình 10: Ảnh giao diện chính ChatBot	78
Hình 11: Ảnh form đăng ký tư vấn.....	79
Hình 12: Ảnh trang quản trị danh sách tư vấn.....	80
Hình 13: Ảnh mã nguồn xử lý	80
Hình 14: Ảnh dữ liệu huấn luyện.....	81

DANH MỤC TỪ VIẾT TẮT VÀ THUẬT NGỮ

Từ viết tắt / Thuật ngữ	Giải thích
HPU	Đại học Quản lý và Công nghệ Hải Phòng (HPU)
ChatBot	Chương trình máy tính tương tác với người dùng bằng ngôn ngữ tự nhiên
LLM	Mô hình ngôn ngữ lớn (Large Language Model)
NLP	Xử lý ngôn ngữ tự nhiên (Natural Language Processing)
AI	Trí tuệ nhân tạo (Artificial Intelligence)
UX	Trải nghiệm người dùng (User Experience)
API	Giao diện lập trình ứng dụng (Application Programming Interface)
Framework	Bộ khung công nghệ hỗ trợ phát triển phần mềm
MIT	Viện Công nghệ Massachusetts

Bảng danh mục từ viết tắt

PHẦN MỞ ĐẦU

1. Lý do chọn đề tài

Trong thời đại chuyển đổi số, việc ứng dụng trí tuệ nhân tạo (AI) vào giáo dục đang trở thành xu thế tất yếu. Trong đó, ChatBot – một công cụ tương tác thông minh đang được nhiều trường đại học trên thế giới áp dụng để hỗ trợ tư vấn, hướng nghiệp, chăm sóc sinh viên và giải đáp các câu hỏi thường gặp. Tuy nhiên, tại Việt Nam, việc ứng dụng ChatBot trong lĩnh vực tuyển sinh vẫn còn mới mẻ, chưa thực sự phổ biến, đặc biệt là ở các trường ngoài công lập.

Trường Đại học Quản lý và Công nghệ Hải Phòng (HPU) là một trong những trường đang chú trọng đổi mới hình thức tuyển sinh, nâng cao chất lượng truyền thông và tiếp cận thí sinh. Tuy nhiên, hình thức tư vấn hiện tại còn phụ thuộc nhiều vào nhân sự, thời gian và khả năng phản hồi trực tiếp. Chính vì vậy, việc xây dựng một ChatBot thông minh – có thể hoạt động 24/7, hỗ trợ trả lời nhanh các câu hỏi liên quan đến tuyển sinh, học phí, ngành đào tạo, v.v. – là một giải pháp mang tính cấp thiết và khả thi.

Từ đó, em chọn thực hiện đề tài “Xây dựng ChatBot AI hỗ trợ tư vấn tuyển sinh cho Trường Đại học Quản lý và Công nghệ Hải Phòng” nhằm tìm hiểu, thiết kế và xây dựng một hệ thống ChatBot ứng dụng mô hình ngôn ngữ lớn (LLM), cụ thể là Gemini 1.5 Flash, kết hợp với dữ liệu huấn luyện riêng của trường để tạo ra một công cụ hỗ trợ tuyển sinh hiệu quả và hiện đại.

2. Mục tiêu và ý nghĩa nghiên cứu

Mục tiêu chính của đề tài là xây dựng được một ChatBot AI có khả năng:

- Tư vấn thông tin tuyển sinh cho HPU nhanh chóng, chính xác.
- Kết hợp dữ liệu huấn luyện thủ công và mô hình LLM (Gemini) để xử lý các câu hỏi đa dạng.
- Tích hợp trên nền tảng web hoặc Messenger để thuận tiện sử dụng.

Ý nghĩa thực tiễn:

- Góp phần hiện đại hóa hoạt động truyền thông – tuyển sinh của nhà trường.

- Giảm tải cho bộ phận tư vấn trong mùa tuyển sinh.
- Tạo tiền đề phát triển các hệ thống AI hỗ trợ học vụ hoặc hành chính trong tương lai.

3. Phạm vi và phương pháp nghiên cứu

- Phạm vi: Đề tài tập trung vào việc tư vấn tuyển sinh hệ đại học chính quy, chưa mở rộng sang tư vấn học vụ hay tra cứu học liệu.
- Phương pháp nghiên cứu: Kết hợp phương pháp tổng hợp tài liệu, phân tích dữ liệu tuyển sinh, xây dựng và thử nghiệm hệ thống theo quy trình phát triển phần mềm.

4. Cấu trúc báo cáo

Báo cáo được chia thành 3 chương chính:

- Chương 1: Cơ sở lý thuyết và hiện trạng vấn đề
- Chương 2: Thiết kế và triển khai ChatBot
- Chương 3: Thử nghiệm, đánh giá và kết luận

5. Tình hình nghiên cứu trong và ngoài nước

• **Nước ngoài**

Việc ứng dụng trí tuệ nhân tạo, đặc biệt là mô hình ngôn ngữ lớn (LLM) vào hệ thống ChatBot đã được nhiều tổ chức và trường đại học trên thế giới nghiên cứu và triển khai. Các nghiên cứu tại MIT, Stanford, hay các tập đoàn công nghệ lớn như Google, OpenAI đã cho thấy tiềm năng ứng dụng LLM trong giao tiếp tự nhiên với con người, trong đó có lĩnh vực giáo dục.

Đặc biệt, OpenAI và Google đã công bố nhiều mô hình AI như GPT, Gemini có khả năng trả lời câu hỏi, tóm tắt văn bản, giải thích khái niệm... mở ra hướng ứng dụng rộng rãi trong việc xây dựng trợ lý học tập và ChatBot tư vấn thông minh.

Một trong các hệ thống ChatBot điển hình như EduChat – được phát triển bởi Yuhao Dan và cộng sự – đã tích hợp LLM vào hệ thống LMS, hỗ trợ học tập

cá nhân hóa và tối ưu hóa quy trình phản hồi tự động đăng tại địa chỉ: <https://byvn.net/c976>.

- **Trong nước**

Tại Việt Nam, nghiên cứu ứng dụng ChatGPT/Gemini trong giáo dục được khảo sát trong các công trình như bài viết “ChatGPT và giáo dục: Tổng quan...” của Hana Trương đăng tại địa chỉ: <https://by.edu.vn/UyFb>.

Tạp chí “Ứng dụng ChatBot AI trong dạy học cá nhân hóa môn Toán học” tại địa chỉ <https://tapchigiaoduc.edu.vn/article/89662/222/ung-dung-ChatBot-ai-trong-day-hoc-ca-nhan-hoa-mon-toan-hoc/>, đã phân tích việc tích hợp ChatBot AI vào sư phạm, tập trung hỗ trợ học sinh giải toán cá nhân hóa theo Chương trình Giáo dục Phổ thông 2018.

Tại trường Đại học Quản lý và Công nghệ Hải Phòng (HPU), chưa có hệ thống ChatBot chính thức hỗ trợ tuyển sinh. Do đó, đề tài này là một nghiên cứu đầu tiên mang tính thử nghiệm và định hướng phát triển lâu dài.

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT VÀ HIỆN TRẠNG

VẤN ĐỀ

1.1. Tổng quan về tư vấn tuyển sinh

Tư vấn tuyển sinh là hoạt động then chốt trong chiến lược truyền thông và phát triển nguồn tuyển của các cơ sở giáo dục đại học. Hoạt động này không chỉ cung cấp thông tin mà còn đóng vai trò định hướng, giải đáp thắc mắc, và tạo dựng niềm tin nơi thí sinh và phụ huynh. Trong bối cảnh cạnh tranh giữa các trường ngày càng gia tăng, tư vấn tuyển sinh hiệu quả giúp trường nâng cao khả năng tiếp cận thí sinh tiềm năng, từ đó tăng tỷ lệ đăng ký xét tuyển và tuyển sinh thực tế.

Tuy nhiên, nhiều thí sinh hiện nay gặp khó khăn khi tiếp cận thông tin vì tài liệu rải rác, thiếu sự nhất quán, hoặc không có người tư vấn vào thời điểm cần thiết. Việc ứng dụng công nghệ, đặc biệt là ChatBot AI, đang được xem như một giải pháp khả thi giúp nâng cao hiệu quả và phạm vi tư vấn tuyển sinh.

1.2. ChatBot và mô hình ngôn ngữ lớn (LLM), xử lý ngôn ngữ tự nhiên (NLP)

ChatBot là một chương trình máy tính có khả năng tương tác với con người qua ngôn ngữ tự nhiên, bằng văn bản hoặc giọng nói. Khi kết hợp với trí tuệ nhân tạo, đặc biệt là các mô hình ngôn ngữ lớn (LLM – Large Language Model), ChatBot có thể hiểu được ngữ cảnh phức tạp, tạo phản hồi mạch lạc và giữ được lịch sử hội thoại.

LLM là các mô hình học sâu được huấn luyện trên hàng tỷ từ ngữ để học cách hiểu và sinh ngôn ngữ giống như con người. Các mô hình như GPT-4 (OpenAI), Gemini (Google), Claude (Anthropic) đã chứng minh khả năng xử lý ngôn ngữ tự nhiên ở cấp độ cao. Trong đề tài này, em sử dụng Gemini 1.5 Flash – một mô hình AI hiệu năng cao, thời gian phản hồi nhanh, đặc biệt phù hợp cho ứng dụng ChatBot.

NLP (Natural Language Processing) – Xử lý ngôn ngữ tự nhiên

NLP là công nghệ được sử dụng để cho phép hệ thống ChatBot hiểu và xử lý ngôn ngữ của con người:

- Phân tích câu hỏi: Hệ thống ChatBot phân tích câu hỏi của người dùng bằng cách sử dụng xử lý ngôn ngữ tự nhiên (NLP) để hiểu ý định và ngữ cảnh.
- Tiền xử lý: Các tác vụ chuẩn bị dữ liệu như chuyển văn bản về chữ thường hoặc bỏ dấu câu được thực hiện trong quá trình tiền xử lý câu hỏi.
- Hiểu ngữ cảnh: NLP giúp ChatBot có khả năng hiểu các câu hỏi phức tạp và đa nghĩa, cũng như duy trì ngữ cảnh trong suốt cuộc trò chuyện.
- Tìm kiếm tương đồng: Các kỹ thuật như so khớp mờ (rapidfuzz) và mã hóa vector (SentenceTransformer) được sử dụng để tìm kiếm các câu hỏi tương tự trong dữ liệu huấn luyện, đây là các kỹ thuật nền tảng của NLP.

Nếu coi ChatBot là một nhân viên tư vấn, thì NLP là khả năng nghe và hiểu ngôn ngữ (phân tích giọng nói, hiểu ý định), còn LLM là kiến thức chuyên môn và khả năng giao tiếp (tổng hợp thông tin để đưa ra câu trả lời chi tiết và mạch lạc).

1.3. Ứng dụng ChatBot trong giáo dục và tuyển sinh

Trên thế giới, ChatBot đã được triển khai trong nhiều hoạt động giáo dục: giải bài tập, tư vấn học vụ, học ngoại ngữ, và đặc biệt là tư vấn tuyển sinh. ChatBot có thể hỗ trợ người học tìm kiếm thông tin, định hướng nghề nghiệp, và giảm tải khối lượng công việc cho cán bộ tư vấn.

Tại Việt Nam, một số trường đại học lớn như Đại học Quốc gia Hà Nội, Đại học FPT đã thử nghiệm ChatBot tuyển sinh, nhưng đa số vẫn chỉ dừng lại ở dạng kịch bản tĩnh. Việc ứng dụng LLM giúp ChatBot linh hoạt hơn, có thể xử lý các câu hỏi ngoài dự kiến, tăng trải nghiệm người dùng.

1.4. Hiện trạng hoạt động tuyển sinh tại HPU

Hiện nay, công tác tư vấn tuyển sinh tại HPU chủ yếu được thực hiện qua các hình thức:

- Tư vấn trực tiếp tại các trường THPT và tại trường
- Tư vấn qua điện thoại, fanpage Facebook, Zalo
- Cập nhật thông tin tuyển sinh trên website: <https://hpu.edu.vn>

Tuy nhiên, các kênh này đều có điểm hạn chế:

- Phụ thuộc nhiều vào nhân lực tư vấn
- Không hoạt động ngoài giờ hành chính
- Thông tin vẫn rải rác, chưa có hệ thống tự động tương tác

Do đó, việc triển khai một ChatBot hoạt động 24/7 để hỗ trợ thí sinh tiếp cận thông tin là rất cần thiết và phù hợp với xu hướng hiện đại hóa giáo dục.

1.5. Phân tích dữ liệu tuyển sinh từ trang hpu.edu.vn

Em đã tổng hợp các thông tin chính thức từ trang tuyển sinh của HPU bao gồm:

- Danh sách ngành học, mã ngành, tổ hợp xét tuyển
- Chỉ tiêu tuyển sinh từng ngành
- Thông tin học phí năm học 2024–2025 (618.000 VNĐ/tín chỉ)
- Chính sách học bổng, điều kiện và mức hỗ trợ
- Các phương thức xét tuyển: học bạ, điểm thi tốt nghiệp THPT, xét tuyển thẳng...

262 FAQs Thông báo Tuyển sinh hệ đại học 2025

Tệp Chính sửa Xem Chèn Định dạng Dữ liệu Công cụ Tiện ích Trợ giúp

100% 123 Mặc định

STT	Câu hỏi	Câu trả lời	Check
1	Chỉ tiêu tuyển sinh năm 2024 của HPU là bao nhiêu?	Chỉ tiêu hệ đại học là 651. Quản trị kinh doanh (210 chỉ tiêu), Công nghệ thông tin (71 chỉ tiêu)	✓
2	HPU có bao nhiêu khoa và chuyên ngành?	HPU có 7 khoa và 19 chuyên ngành như CNTT, Kỹ thuật môi trường, QTKD, Ngôn ngữ Anh, v.v.	✓
3	Trường có bao nhiêu khoa - ngành	7 khoa gồm: công nghệ thông tin, Công nghệ kỹ thuật điện, điện tử, Kỹ thuật môi trường, Quản trị kinh doanh, Ngôn ngữ Anh, Vi	✓
4	Trường có những chuyên ngành nào	14 chuyên ngành: Công nghệ phần mềm, Quản trị và an ninh mạng, Công nghệ IoT, Điện tử - Truyền thông, Điện tử đồng công r	✓
5	Học phí tại HPU là bao nhiêu?	Học phí áp dụng cho năm học 2025-2026 là 618.000 đồng/1 tín chỉ.	✓
6	HPU có những loại học bổng nào?	Học bổng Tự hào HPU, Học bổng Tài năng HPU, Học bổng Hữu nghị. Mức cao nhất là 60 triệu đồng. Ngoài ra còn có Học bổng	✓
7	Điều kiện nhận học bổng Tự hào HPU là gì?	Điểm trung bình tuyển nguyên vọng 1, điểm trung bình xét tuyển ≥ 7.5 hoặc có thành tích xuất sắc.	✓
8	HPU có phương thức tuyển sinh nào?	3 phương thức: Xét điểm thi THPT, xét học bạ, kết hợp điểm thi với chứng chỉ quốc tế. Năm 2025, HPU xét tuyển theo kết quả k	✓
9	Kỳ túc xá HPU có giá bao nhiêu?	Phòng tiêu chuẩn: 450.000đ/tháng/người; Phòng có điều hòa: 600.000đ/tháng/người (chưa tính điện nước). Sinh viên trường đư	✓
10	Thời gian đào tạo tại HPU là bao lâu?	Thời gian học là 3.5 năm.	✓
11	Ngành mới tại HPU năm 2025 là gì?	Ngành ngôn ngữ Trung	✓
12	Cơ hội việc làm sau tốt nghiệp của HPU như thế nào?	Sinh viên có việc làm tại các doanh nghiệp, tập đoàn trong và ngoài nước. Tỷ lệ có việc làm đạt 94.6% sau một năm tốt nghiệp.	✓
13	Làm thế nào để liên hệ tuyển sinh HPU?	Hotline và Zalo tuyển sinh: 0901 598 698 - 0936 821 821.	✓
14	Sinh viên có được miễn giảm học phí không?	Sinh viên khó khăn có thể được hỗ trợ học bổng Hữu nghị (2.5-5 triệu đồng/học kỳ). Sinh viên diện chính sách, khó khăn hoặc th	✓
15	HPU có những ngành nào nổi bật trong Cách mạng Công nghiệp 4.0?	Các ngành nổi bật như Công nghệ Thông tin, Công nghệ Kỹ thuật Điện - Điện tử, Quản trị Kinh doanh (Logistics).	✓
16	Trường có bể bơi không?	Có.	✓
17	Sinh viên có phải trả phí khi sử dụng bể bơi và sân vận động không?	Không phải trả phí khi sử dụng trong thời khóa biểu học tập; ngoài giờ học thì có thu phí.	✓
18	Trường có hỗ trợ việc làm thêm part-time phù hợp ngành học không?	Có, HPU giới thiệu việc làm thêm part-time phù hợp ngành học thông qua kênh việc làm của trường.	✓
19	Các hình thức tuyển sinh tại HPU là gì?	Xét kết quả thi tốt nghiệp THPT, xét học bạ THPT, kết hợp điểm thi với chứng chỉ quốc tế. Năm 2025, có 2 phương thức xét tuye	✓
20	Thành tích nổi bật của HPU trong đào tạo là gì?	Tỷ lệ sinh viên có việc làm sau tốt nghiệp đạt 96.4%. Chương trình học ứng dụng, kết nối doanh nghiệp từ năm 1.	✓
21	Ngành Ngôn ngữ tại HPU có những chuyên ngành nào?	Ngôn ngữ Anh, Ngôn ngữ Anh - Trung, Ngôn ngữ Anh - Hàn, Ngôn ngữ Anh - Nhật.	✓
22	Trường có tổ chức các hoạt động ngoại khóa không?	Trường tổ chức các hoạt động ngoại khóa, bao gồm văn nghệ, thể thao, câu lạc bộ.	✓
23	Trường có ngành Kỹ thuật Môi trường không?	Có, với các chuyên ngành như: Kỹ thuật môi trường và an toàn lao động, Quản lý tài nguyên nước và đất. Năm 2025 có thêm ch	✓
24	Thời gian nhận hồ sơ xét tuyển của HPU là khi nào?	Đợt 1 (đầy kiến): Từ 1/4/2025 - 30/7/2025. Nhập học (đầy kiến): 22/8 - 27/8/2025.	✓
25	Trường có gì khác biệt so với các trường khác?	Thời gian học ngắn (3.5 năm), đào tạo ứng dụng, kết nối doanh nghiệp từ sớm, cơ sở vật chất hiện đại.	✓
26	Trường có hỗ trợ sinh viên quốc tế không?	Có, với chương trình giảng dạy quốc tế, kỹ túc xá tiêu chuẩn, và các câu lạc bộ dành cho sinh viên quốc tế.	✓
27	Trường có ngành Việt Nam học không?	Có, với chuyên ngành Văn hóa Du lịch và Quản trị Du lịch.	✓
28	Trường có ngành marketing không?	Có.	✓
29	Lịch học tại HPU có linh động không?	Lịch học linh động, cho phép sinh viên cân bằng giữa học tập và làm thêm.	✓
30	Trường có những chính sách hỗ trợ sinh viên nghèo không?	Sinh viên nghèo được xét học bổng Hữu nghị, giảm từ 2.5-5 triệu đồng/học kỳ nếu đạt kết quả học tập tốt.	✓

Trang tính 1

Hình 1: Bộ FAQ thông tin tuyển sinh

Từ đó, em xây dựng bộ dữ liệu huấn luyện cho ChatBot theo cấu trúc JSON, bao gồm hàng trăm câu hỏi – đáp mẫu như:

- "Ngành Công nghệ thông tin có mã ngành gì?"
- "Điểm chuẩn năm ngoái là bao nhiêu?"
- "Học phí mỗi kỳ là bao nhiêu?"
- "Có học bổng tuyển sinh không?"

1150 FAQs Học phí và các khoản đóng góp của sinh viên

STT	Câu hỏi	Câu trả lời	Check
1	Học phí chương trình đào tạo cao học tại HPU là bao nhiêu?	Học phí chương trình đào tạo cao học là 1.150.000 VND/tháng.	☒
2	Học phí chương trình đại học chính quy (tổ chức trong giờ hành chính) tại HPU tính theo	Học phí chương trình đại học chính quy (tổ chức trong giờ hành chính) là 618.000 VND/tin chỉ.	☒
3	Học phí chương trình đại học chính quy (tổ chức trong giờ hành chính) tính theo bình quân tháng cho chương trình đại học chính quy (tổ chức trong giờ hành chính) là 2.207.000 VND/tháng.		☒
4	Học phí chương trình đại học chính quy theo tiết học đối với lớp có sĩ số từ 16 sinh viên trở lên là 45.000 VND/tiết.	Học phí theo tiết học đối với lớp có sĩ số từ 16 sinh viên trở lên là 45.000 VND/tiết.	☒
5	Học phí chương trình đại học chính quy theo tiết học đối với lớp có từ 10 đến dưới 15 sin	Học phí theo tiết học đối với lớp có từ 10 đến dưới 15 sinh viên là 75.000 VND/tiết.	☒
6	Học phí chương trình đại học chính quy theo tiết học đối với lớp có từ 05 đến dưới 09 sin	Học phí theo tiết học đối với lớp có từ 05 đến dưới 09 sinh viên là 150.000 VND/tiết.	☒
7	Học phí chương trình đại học chính quy theo tiết học đối với lớp có dưới 05 sinh viên là b	Học phí theo tiết học đối với lớp có dưới 05 sinh viên là 220.000 VND/tiết.	☒
8	Mức thu học lại, thi lại học phần thuộc chương trình cao học tại HPU là bao nhiêu?	Mức thu học lại, thi lại học phần thuộc chương trình cao học là 150.000 VND/học phần.	☒
9	Mức thu học lại, thi lại học phần thuộc chương trình đại học tại HPU là bao nhiêu?	Mức thu học lại, thi lại học phần thuộc chương trình đại học là 75.000 VND/học phần.	☒
10	Mức thu cải thiện điểm tại HPU là bao nhiêu?	Mức thu cải thiện điểm là 95.000 VND/học phần.	☒
11	Phí phúc tra bài thi học phần và phải sửa điểm do lỗi của người chấm, ghép phách, nhập	Phí phúc tra bài thi học phần và phải sửa điểm do lỗi của người chấm, ghép phách, nhập điểm là Miễn phí.	☒
12	Phí phúc tra bài thi học phần và phải sửa điểm không do lỗi của người học tại HPU là bao	Phí phúc tra bài thi học phần và phải sửa điểm không do lỗi của người học là 100.000 VND/học phần.	☒
13	Phí phúc tra kết quả thi tốt nghiệp đại học tại HPU là bao nhiêu?	Phí phúc tra kết quả thi tốt nghiệp đại học là 300.000 VND/lần.	☒
14	Phí phúc tra kết quả bảo vệ khóa luận tốt nghiệp đại học tại HPU là bao nhiêu?	Phí phúc tra kết quả bảo vệ khóa luận tốt nghiệp đại học là 1.500.000 VND/lần.	☒
15	Lệ phí xét tuyển và công nhận điểm học phần tại HPU là bao nhiêu?	Lệ phí xét tuyển và công nhận điểm học phần là 60.000 VND/tin chỉ.	☒
16	Lệ phí bảo vệ đồ án tốt nghiệp đại học tại HPU là bao nhiêu?	Lệ phí bảo vệ đồ án tốt nghiệp đại học là 1.850.000 VND/tin chỉ.	☒
17	Lệ phí bảo vệ đồ án tốt nghiệp thạc sĩ tại HPU là bao nhiêu?	Lệ phí bảo vệ đồ án tốt nghiệp thạc sĩ là 3.800.000 VND/tin chỉ.	☒
18	Lệ phí cấp bằng tốt nghiệp cao học tại HPU là bao nhiêu?	Lệ phí cấp bằng tốt nghiệp cao học là 1.000.000 VND/lần.	☒
19	Lệ phí cấp bằng tốt nghiệp đại học chính quy tại HPU là bao nhiêu?	Lệ phí cấp bằng tốt nghiệp đại học chính quy là 300.000 VND/bản.	☒
20	Lệ phí cấp bản sao bằng tốt nghiệp đại học tại HPU là bao nhiêu?	Lệ phí cấp bản sao bằng tốt nghiệp là 100.000 VND/bản.	☒
21	Lệ phí cấp chứng nhận tốt nghiệp đại học tại HPU là bao nhiêu?	Lệ phí cấp chứng nhận tốt nghiệp là 50.000 VND/bản.	☒
22	Lệ phí cấp bằng điểm khóa học tại HPU là bao nhiêu?	Lệ phí cấp bằng điểm khóa học là 50.000 VND/bản.	☒
23	Phí xét tuyển khi thi sinh trúng tuyển chương trình đại học chính quy theo hình thức xét t	Phí xét tuyển khi thi sinh trúng tuyển chương trình đại học chính quy theo hình thức xét tuyển thẳng là Theo quy định chung.	☐
24	Phí xét tuyển khi thi sinh trúng tuyển chương trình đại học chính quy theo hình thức liên t	Phí xét tuyển khi thi sinh trúng tuyển chương trình đại học chính quy theo hình thức liên thông là 200.000 VND/hồ sơ.	☐
25	Phí xét tuyển khi thi sinh trúng tuyển hệ vừa học vừa làm tại HPU là bao nhiêu?	Phí xét tuyển khi thi sinh trúng tuyển hệ vừa học vừa làm là 200.000 VND/hồ sơ.	☐
26	Phí nhập học và làm thẻ sinh viên cao học tại HPU là bao nhiêu?	Phí nhập học và làm thẻ sinh viên cao học là 300.000 VND/học viên.	☐
27	Phí nhập học, khám sức khỏe và làm thẻ sinh viên đại học chính quy tại HPU là bao nhiê	Phí nhập học, khám sức khỏe và làm thẻ sinh viên đại học chính quy là 300.000 VND/sinh viên.	☐
28	Phí hướng dẫn sử dụng thư viện tại HPU là bao nhiêu?	Phí hướng dẫn sử dụng thư viện là 150.000 VND/sinh viên.	☐
29	Phí hỗ trợ bảo trì cơ sở vật chất, vệ sinh môi trường tại HPU là bao nhiêu?	Phí hỗ trợ bảo trì cơ sở vật chất, vệ sinh môi trường là 200.000 VND/tháng.	☐
30	Tổ chức các hoạt động thể thao, văn hóa tại HPU có mức phí là bao nhiêu?	Mức phí tổ chức các hoạt động thể thao, văn hóa là 50.000 VND/tháng.	☐

Hình 2: Bộ FAQ học phí và các khoản đóng góp

Việc chuẩn hóa các dữ liệu này giúp ChatBot hoạt động ổn định khi phản hồi các truy vấn từ người dùng.

1.6. Kết luận chương

Chương này đã trình bày tổng quan về lý thuyết liên quan đến tư vấn tuyển sinh, mô hình ngôn ngữ lớn (LLM), cũng như hiện trạng hoạt động tuyển sinh tại HPU. Qua phân tích dữ liệu thực tế từ website nhà trường, em đã xây dựng được cơ sở ban đầu để phát triển hệ thống ChatBot, đảm bảo phản ánh đúng nhu cầu thông tin tuyển sinh của người học và phụ huynh.

CHƯƠNG 2: THIẾT KẾ VÀ TRIỂN KHAI CHATBOT

2.1. Mục tiêu thiết kế hệ thống

Mục tiêu chính của hệ thống là xây dựng một ChatBot hỗ trợ tư vấn tuyển sinh cho Trường Đại học Quản lý và Công nghệ Hải Phòng, có khả năng:

- Tương tác với người dùng bằng ngôn ngữ tự nhiên tiếng Việt;
- Truy xuất thông tin tuyển sinh chính xác từ cơ sở dữ liệu huấn luyện;
- Kết hợp với mô hình ngôn ngữ lớn (LLM) để xử lý các truy vấn ngoài dữ liệu;
- Dễ dàng tích hợp vào website, fanpage hoặc các nền tảng phổ biến;
- Có khả năng mở rộng, cập nhật dữ liệu linh hoạt trong tương lai.

2.2. Kiến trúc hệ thống và công nghệ sử dụng

Giải pháp công nghệ có thể lựa chọn:

- Sử dụng API từ OpenAI hoặc Google...

API (Application Programming Interface - Giao diện lập trình ứng dụng) là một cầu nối cho phép hai phần mềm, ứng dụng hoặc hệ thống giao tiếp với nhau mà không cần biết chi tiết bên trong của nhau. Ứng dụng gửi yêu cầu → API xử lý → trả kết quả

- Nền tảng ChatBot tích hợp AI: Các nền tảng như Dialogflow, Botpress, hoặc các dịch vụ ChatBot SaaS (Tidio, Landbot, Intercom) có thể kết nối với GPT.
- Tích hợp qua WordPress hoặc hệ thống CMS: Nếu website xây dựng trên WordPress, có thể sử dụng các plugin như WPBot hoặc ChatBot for WordPress.

2.2.1. Sử dụng API từ Google Gemini

2.2.1.1. Kiến trúc hệ thống

Hệ thống được thiết kế theo mô hình client–server gồm 3 thành phần chính:

- Frontend (giao diện người dùng): Giao diện HTML/CSS/JavaScript đơn giản, cho phép người dùng nhập câu hỏi và nhận phản hồi;
- Backend (xử lý logic): Sử dụng Flask (Python) để xây dựng API tiếp nhận câu hỏi, xử lý logic lọc đầu vào, truy vấn dữ liệu hoặc gọi API Gemini;
- Dữ liệu và mô hình AI:
 - Dữ liệu huấn luyện nội bộ (trained_data.json)
 - Lịch sử hội thoại (chat_history.json)
 - Mô hình ngôn ngữ lớn: Gemini 1.5 Flash (Google)

2.2.1.1.1. Yêu cầu phần mềm.

ChatBot AI được tích hợp vào webchat hiện có của Đại học HPU nhằm hỗ trợ công tác tuyển sinh. ChatBot AI này sẽ thay thế hoặc bổ sung cho phương thức tư vấn trực tiếp, giúp Đại học HPU cung cấp thông tin tuyển sinh một cách nhanh chóng, chính xác và hiệu quả hơn. Các yêu cầu của phần mềm bao gồm:

- **Hỗ trợ thông tin tuyển sinh:** ChatBot phải có khả năng tự động trả lời các câu hỏi thường gặp (FAQs) liên quan đến tuyển sinh, bao gồm điều kiện xét tuyển, hồ sơ, học phí, học bổng, thời gian nộp hồ sơ, và hướng dẫn đăng ký.
- **Giải đáp thắc mắc về chương trình học:** ChatBot cần cung cấp thông tin về các ngành đào tạo, thời gian học, chương trình liên kết quốc tế, và các môn học.
- **Cung cấp thông tin khác:** ChatBot phải có khả năng giải đáp các câu hỏi liên quan đến ký túc xá, lịch tuyển sinh, phương thức liên hệ, các hoạt động ngoại khóa, thông tin về cán bộ và giảng viên HPU.
- **Hướng dẫn nộp hồ sơ:** ChatBot cần hướng dẫn người dùng về quy trình và thủ tục nộp hồ sơ tuyển sinh.

- **Tư vấn chương trình học:** ChatBot có khả năng tư vấn sơ bộ về các chương trình học phù hợp với nhu cầu của người dùng.
- **Giao diện thân thiện:** Giao diện chat phải trực quan, dễ sử dụng và có thể truy cập dễ dàng trên cả máy tính và thiết bị di động.
- **Khả năng tương tác linh hoạt:** ChatBot cần có khả năng xử lý các kịch bản hội thoại cơ bản và phức tạp, dẫn dắt người dùng qua nhiều bước nếu cần.
- **Tùy chỉnh giọng điệu và phong cách:** ChatBot cần có giọng điệu thân thiện, chuyên nghiệp và dễ hiểu, phù hợp với đối tượng học sinh và phụ huynh.
- **Khả năng xử lý lỗi:** ChatBot cần có cơ chế xử lý các câu hỏi không có câu trả lời sẵn, ví dụ như gợi ý đặt câu hỏi khác hoặc chuyển hướng đến nhân viên tư vấn trực tiếp.
- **Gợi ý liên kết:** ChatBot có khả năng cung cấp các liên kết đến các trang web hoặc tài liệu liên quan để người dùng xem thêm thông tin chi tiết.
- **Thu thập thông tin liên hệ (tùy chọn):** ChatBot có thể được trang bị khả năng thu thập thông tin liên hệ của người dùng (ví dụ: tên, email, số điện thoại) để phục vụ cho mục đích tư vấn thêm.
- **Hỗ trợ đa ngôn ngữ (tùy chọn):** ChatBot có khả năng hỗ trợ tư vấn bằng nhiều ngôn ngữ khác nhau nếu cần thiết.
- **Khả năng học hỏi và cải thiện:** Hệ thống cần có khả năng theo dõi và phân tích các tương tác của người dùng để cải thiện độ chính xác và hiệu quả của ChatBot theo thời gian.
- **Bảo mật và quyền riêng tư:** Đảm bảo dữ liệu người dùng được bảo mật theo các quy định hiện hành.
- **Khả năng kết hợp với hỗ trợ trực tiếp (Live Chat):** Hệ thống có khả năng chuyển giao cuộc trò chuyện từ ChatBot sang nhân viên tư vấn trực tiếp khi cần thiết để giải quyết các câu hỏi phức tạp.

2.2.1.1.2. Đặc tả yêu cầu phần mềm.

Phần mềm ChatBot AI hỗ trợ tuyển sinh cho Đại học HPU phải đáp ứng các yêu cầu sau:

- **Giao diện người dùng:**
 - **Trực quan và dễ sử dụng:** Giao diện webchat tích hợp ChatBot phải thân thiện, dễ dàng tìm thấy và sử dụng cho mọi đối tượng người dùng.
 - **Hỗ trợ đa thiết bị:** Webchat và ChatBot phải hoạt động tốt và nhất quán trên nhiều loại thiết bị như máy tính để bàn, máy tính bảng và điện thoại di động.
 - **Tích hợp liền mạch:** ChatBot phải được tích hợp mượt mà vào giao diện webchat hiện có của Đại học HPU, tạo cảm giác tự nhiên cho người dùng.
 - **Hiển thị rõ ràng:** Các tin nhắn của ChatBot và người dùng phải được hiển thị rõ ràng, dễ đọc và phân biệt.
 - **Khả năng tùy chỉnh giao diện (tùy chọn):** Cho phép tùy chỉnh màu sắc, biểu tượng để phù hợp với nhận diện thương hiệu của HPU.
- **Bảo mật:**
 - **Bảo vệ dữ liệu người dùng:** Mọi thông tin người dùng nhập vào webchat (nếu có thu thập) phải được bảo vệ theo các tiêu chuẩn bảo mật hiện hành và tuân thủ Nghị định 13 về bảo vệ thông tin cá nhân.
 - **An toàn giao tiếp:** Đảm bảo an toàn cho các phiên trò chuyện giữa người dùng và ChatBot.
 - **Ẩn danh thông tin nhạy cảm:** Khi tương tác với các API bên ngoài (ví dụ: Gemini API), thông tin nhạy cảm cần được ẩn danh để bảo vệ quyền riêng tư.

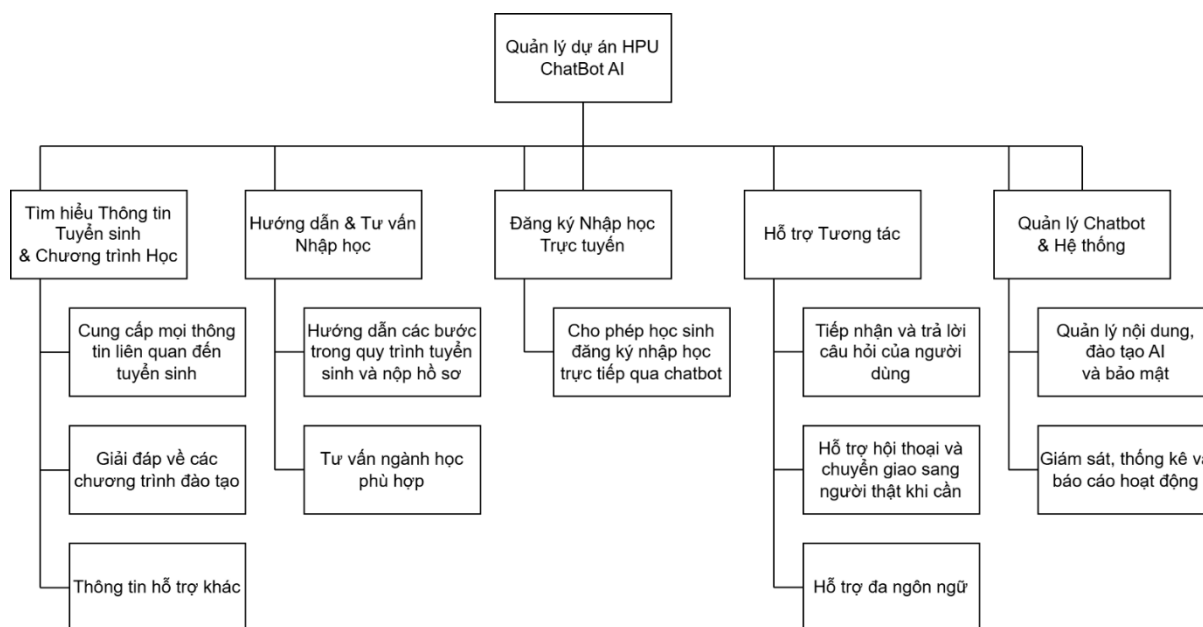
- **Hiệu suất:**
 - **Thời gian phản hồi nhanh:** ChatBot phải có khả năng phản hồi câu hỏi của người dùng một cách nhanh chóng để đảm bảo trải nghiệm người dùng tốt.
 - **Xử lý đồng thời nhiều phiên:** Hệ thống phải có khả năng xử lý đồng thời nhiều phiên trò chuyện mà không bị chậm trễ.
 - **Ổn định:** ChatBot phải hoạt động ổn định, ít xảy ra lỗi hoặc gián đoạn.
- **Tính khả dụng:**
 - **Hoạt động liên tục:** ChatBot cần hoạt động 24/7 để hỗ trợ người dùng mọi lúc mọi nơi.
 - **Dễ dàng truy cập:** Người dùng có thể dễ dàng truy cập và sử dụng ChatBot trên trang web tuyển sinh của HPU.
 - **Khả năng phục hồi:** Hệ thống cần có khả năng tự phục hồi sau các sự cố kỹ thuật nếu có.
- **Khả năng mở rộng:**
 - **Dễ dàng cập nhật nội dung:** Việc thêm, sửa đổi hoặc cập nhật thông tin và kịch bản hội thoại của ChatBot cần được thực hiện dễ dàng.
 - **Hỗ trợ thêm ngôn ngữ (tùy chọn):** Hệ thống cần có khả năng mở rộng để hỗ trợ thêm các ngôn ngữ khác trong tương lai.
 - **Tích hợp với các hệ thống khác (tùy chọn):** Có khả năng tích hợp với các hệ thống khác của trường như CRM, hệ thống quản lý sinh viên (nếu cần).
 - **Khả năng kết nối với các ChatBot khác trong tương lai:** Hệ thống có kiến trúc mở để hỗ trợ phát triển và tích hợp các ChatBot cho các khoa, phòng ban khác của HPU.

- **Khả năng nghiệp vụ của ChatBot:**

- **Trả lời câu hỏi thường gặp (FAQs):** ChatBot phải được đào tạo để trả lời chính xác và đầy đủ các câu hỏi thường gặp về tuyển sinh, chương trình học, học phí, học bổng, v.v..
- **Hiểu ngữ cảnh:** ChatBot cần có khả năng hiểu các câu hỏi phức tạp và đa nghĩa, duy trì ngữ cảnh trong suốt cuộc trò chuyện.
- **Dẫn dắt hội thoại:** ChatBot có thể dẫn dắt người dùng qua các bước để thu thập thông tin cần thiết và cung cấp câu trả lời phù hợp.
- **Tùy chỉnh giọng điệu:** Giọng điệu của ChatBot phải thân thiện, chuyên nghiệp và phù hợp với đối tượng người dùng.
- **Xử lý lỗi và câu hỏi ngoài phạm vi:** ChatBot cần có cơ chế xử lý khi không tìm thấy câu trả lời hoặc khi người dùng đặt câu hỏi không liên quan (ví dụ: gợi ý đặt câu hỏi khác, chuyển sang tư vấn trực tiếp).
- **Gợi ý liên kết:** ChatBot có khả năng gợi ý các liên kết đến các trang web hoặc tài liệu liên quan để người dùng tìm hiểu thêm thông tin.
- **Thu thập thông tin (tùy chọn):** ChatBot có thể thu thập thông tin liên hệ của người dùng để phục vụ tư vấn thêm.
- **Hỗ trợ đa ngôn ngữ (tùy chọn):** ChatBot có khả năng giao tiếp bằng nhiều ngôn ngữ.
- **Khả năng tự học và cải thiện (GEN AI - tùy chọn):** Nghiên cứu và phát triển khả năng tự học, tự sinh đáp án dựa trên dữ liệu hiện có để giảm bớt sự phụ thuộc vào việc cập nhật thủ công.
- **Kiểm soát tính nhất quán dữ liệu:** Đảm bảo dữ liệu đầu vào cho ChatBot luôn nhất quán và đồng bộ.

2.2.1.1.3. Xác định chức năng phần mềm (BFD - Business Function Diagram).

Dưới đây là sơ đồ chức năng chính của phần mềm:



Hình 3: Sơ đồ phân rã chức năng

1. Tìm hiểu Thông tin Tuyển sinh & Chương trình Học:

- **Cung cấp mọi thông tin liên quan đến tuyển sinh** (điều kiện, hồ sơ, học phí, học bổng, thời gian, v.v.)
- **Giải đáp về các chương trình đào tạo** (ngành học, thời gian, môn học, chương trình liên kết)
- **Thông tin hỗ trợ khác** (ký túc xá, liên hệ, hoạt động, giảng viên)

2. Hướng dẫn & Tư vấn Nhập học:

- **Hướng dẫn các bước trong quy trình tuyển sinh và nộp hồ sơ**
- **Tư vấn ngành học phù hợp**

3. Đăng ký Nhập học Trực tuyến:

- **Cho phép học sinh đăng ký nhập học trực tiếp qua ChatBot** (điền thông tin, chọn ngành)

4. Hỗ trợ Tương tác:

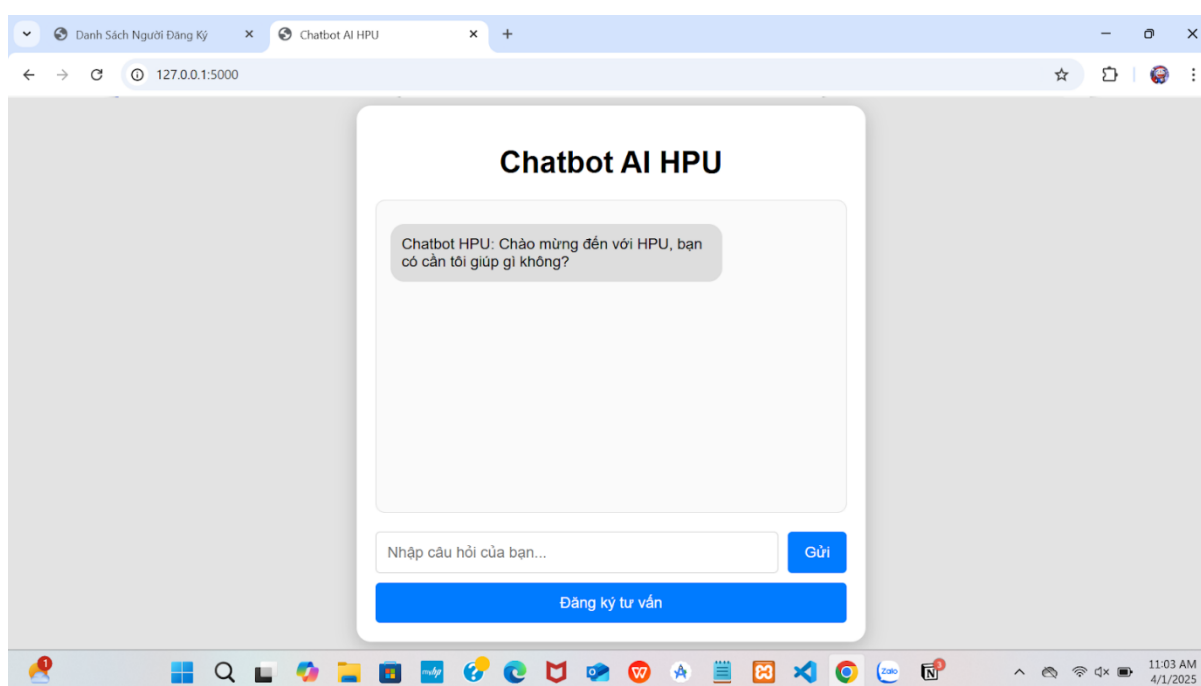
- **Tiếp nhận và trả lời câu hỏi của người dùng** (xử lý ngôn ngữ tự nhiên).
- **Hỗ trợ hội thoại và chuyển giao sang người thật khi cần** (tích hợp Live Chat)
- **Hỗ trợ đa ngôn ngữ** (tùy chọn)

5. Quản lý ChatBot & Hệ thống:

- **Quản lý nội dung, đào tạo AI và bảo mật**
- **Giám sát, thống kê và báo cáo hoạt động**

2.2.1.1.4. Đặc tả chi tiết chức năng.

a. Hỗ trợ Tương tác: Chào hỏi và bắt đầu hội thoại



Hình 4: Chào hỏi và bắt đầu hội thoại

Sự kiện kích hoạt:

- **Khi nào:** Khi người dùng truy cập vào giao diện webchat của Đại học HPU và mở ChatBot AI.
- **Cái gì dẫn đến:** Người dùng muốn tìm kiếm thông tin hoặc cần hỗ trợ từ ChatBot về các vấn đề liên quan đến tuyển sinh, chương trình học, hoặc các dịch vụ khác của HPU.
- **Điều kiện:**
 - ChatBot phải đang hoạt động và được tích hợp trên trang web của HPU.
 - Người dùng không cần đăng nhập để sử dụng chức năng này (hỗ trợ cả người dùng chưa đăng ký).

Quy trình:

1. Người dùng truy cập trang web của Đại học HPU và nhấp vào biểu tượng ChatBot (thường nằm ở góc dưới bên phải của trang web).
2. ChatBot tự động khởi tạo và hiển thị tin nhắn chào hỏi mặc định: "Chào mừng đến với HPU, bạn cần tôi giúp gì không?"
3. Người dùng nhập câu hỏi hoặc yêu cầu hỗ trợ vào ô nhập liệu (ví dụ: "Tôi muốn biết về ngành Công nghệ Thông tin").
4. Hệ thống ChatBot phân tích câu hỏi của người dùng bằng cách sử dụng xử lý ngôn ngữ tự nhiên (NLP) để hiểu ý định và ngữ cảnh.
5. ChatBot trả lời dựa trên cơ sở dữ liệu đã được đào tạo (FAQs, thông tin tuyển sinh, chương trình học, v.v.) hoặc dẫn dắt người dùng qua các bước tiếp theo nếu cần thêm thông tin.
6. Nếu ChatBot không thể trả lời, hệ thống sẽ gợi ý người dùng đặt câu hỏi khác hoặc chuyển hướng đến nhân viên tư vấn trực tiếp (Live Chat).

Giao diện:

- **Tiêu đề:** "ChatBot AI HPU" được hiển thị ở phần đầu của cửa sổ chat.
- **Tin nhắn chào hỏi:** Hiển thị tin nhắn mặc định "Chào mừng đến với HPU, bạn cần tôi giúp gì không?" trong một bong bóng chat màu xám nhạt, bên trái.
- **Ô nhập liệu:** Một ô nhập liệu với placeholder "Nhập câu hỏi của bạn..." nằm ở phía dưới cửa sổ chat.
- **Nút gửi:** Một nút màu xanh dương với nhãn "Gửi" nằm bên phải ô nhập liệu.
- **Nút đăng ký tư vấn:** Một nút màu xanh dương với nhãn "Đăng ký tư vấn" nằm dưới ô nhập liệu, cho phép người dùng chuyển sang tư vấn trực tiếp nếu cần.

Dữ liệu đầu vào:

- **Câu hỏi của người dùng (chuỗi ký tự):**
 - Không giới hạn độ dài, có thể chứa ký tự Unicode (tiếng Việt).
 - Ví dụ: "Chào mừng đến với HPU, bạn cần tôi giúp gì không?" (tin nhắn mặc định từ ChatBot).
- **Ngữ cảnh hội thoại (nếu có):** Hệ thống lưu trữ ngữ cảnh của cuộc trò chuyện để duy trì tính liên tục (ví dụ: nếu người dùng hỏi tiếp "Ngành CNTT học bao lâu?", ChatBot sẽ hiểu ngữ cảnh liên quan đến ngành học).

Dữ liệu đầu ra:

- **Tin nhắn trả lời (chuỗi ký tự):**
 - Tin nhắn chào hỏi mặc định: "Chào mừng đến với HPU, bạn cần tôi giúp gì không?" (khớp với dữ liệu đã được lập trình sẵn).
 - Các câu trả lời tiếp theo sẽ phụ thuộc vào câu hỏi của người dùng, được lấy từ cơ sở dữ liệu hoặc sinh ra dựa trên mô hình AI.

- **Gợi ý hành động (nếu có):**
 - Ví dụ: "Bạn có muốn đăng ký tư vấn trực tiếp không?" với nút "Đăng ký tư vấn".
- **Liên kết (tùy chọn):** ChatBot có thể cung cấp liên kết đến các trang web hoặc tài liệu liên quan (ví dụ: trang thông tin ngành học, hướng dẫn nộp hồ sơ).

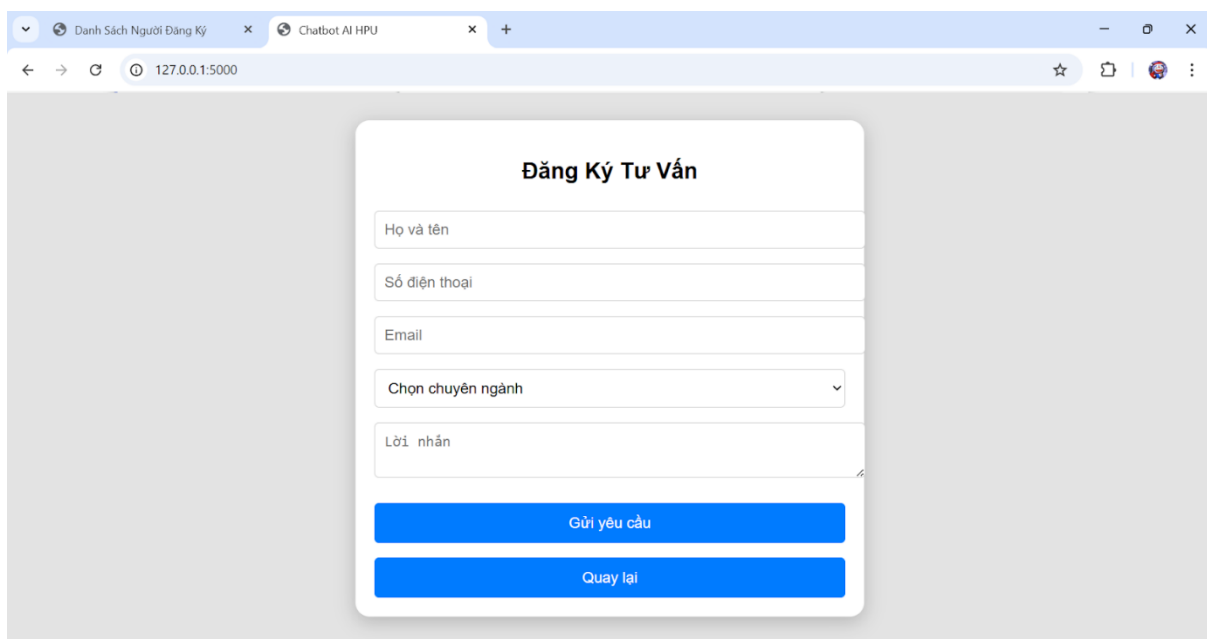
Quy tắc nghiệp vụ tuân thủ:

- **Tính thân thiện:** Tin nhắn chào hỏi phải có giọng điệu chuyên nghiệp, thân thiện, và phù hợp với đối tượng học sinh, phụ huynh.
- **Khả năng xử lý lỗi:** Nếu người dùng không nhập câu hỏi hoặc nhập nội dung không rõ ràng (ví dụ: "???"), ChatBot sẽ trả lời: "Xin lỗi, tôi chưa hiểu câu hỏi của bạn. Bạn có thể hỏi lại hoặc nhấn 'Đăng ký tư vấn' để được hỗ trợ trực tiếp."
- **Duy trì ngữ cảnh:** ChatBot phải duy trì ngữ cảnh hội thoại để trả lời các câu hỏi liên quan một cách logic (ví dụ: nếu người dùng hỏi "Học phí bao nhiêu?" sau khi hỏi về ngành CNTT, ChatBot sẽ hiểu và trả lời học phí của ngành CNTT).
- **Bảo mật thông tin:** Không lưu trữ thông tin cá nhân của người dùng trong giai đoạn này trừ khi người dùng chủ động cung cấp (ví dụ: qua chức năng thu thập thông tin liên hệ).
- **Khả năng chuyển giao:** Nếu ChatBot không thể trả lời sau 2 lần thử, hệ thống sẽ tự động gợi ý chuyển sang tư vấn trực tiếp và hiển thị nút "Đăng ký tư vấn".
- **Hiệu suất:** ChatBot phải phản hồi trong vòng 1-2 giây để đảm bảo trải nghiệm người dùng mượt mà.
- **Tính nhất quán:** Tin nhắn chào hỏi và các phản hồi phải đồng bộ với dữ liệu đã được phê duyệt bởi nhà trường HPU.

Giải thích thêm:

Chức năng "Chào hỏi và bắt đầu hội thoại" là bước đầu tiên trong nhóm chức năng **Hỗ trợ Tương tác**, nhằm tạo ấn tượng ban đầu tốt với người dùng và khuyến khích họ tương tác với ChatBot. Dựa trên ảnh, giao diện đã được thiết kế đơn giản, trực quan, với các nút hành động rõ ràng ("Gửi", "Đăng ký tư vấn"), phù hợp với yêu cầu của dự án về giao diện thân thiện và dễ sử dụng.

b. Hỗ trợ Tương tác: Thu thập thông tin liên hệ và đăng ký tư vấn



Hình 5: Form đăng ký tư vấn

Sự kiện kích hoạt:

- **Khi nào:** Khi người dùng nhấn vào nút "Đăng ký tư vấn" trên giao diện ChatBot AI HPU.
- **Cái gì dẫn đến:** Người dùng muốn được tư vấn trực tiếp bởi nhân viên của HPU hoặc ChatBot không thể trả lời câu hỏi của họ, dẫn đến việc cần thu thập thông tin liên hệ để chuyển giao cho bộ phận tư vấn.
- **Điều kiện:**
 - ChatBot đang hoạt động và người dùng đã tương tác với ChatBot trước đó.

- Người dùng không cần đăng nhập để sử dụng chức năng này (hỗ trợ cả người dùng chưa đăng ký).

Quy trình:

1. Người dùng nhấn nút "Đăng ký tư vấn" trên giao diện ChatBot (như trong ảnh trước đó).
2. Hệ thống hiển thị form đăng ký tư vấn với các trường thông tin: "Họ và tên", "Số điện thoại", "Email", "Chọn chuyên ngành", và "Lời nhắn".
3. Người dùng điền thông tin vào các trường:
 - "Họ và tên": Nhập tên đầy đủ.
 - "Số điện thoại": Nhập số điện thoại liên lạc.
 - "Email": Nhập địa chỉ email.
 - "Chọn chuyên ngành": Chọn một chuyên ngành từ danh sách dropdown (ví dụ: Công nghệ Thông tin, Quản trị Kinh doanh, v.v.).
 - "Lời nhắn": Nhập thêm thông tin hoặc câu hỏi (tùy chọn).
4. Người dùng nhấn nút "Gửi yêu cầu" để gửi thông tin.
5. Hệ thống kiểm tra tính hợp lệ của dữ liệu:
 - Các trường bắt buộc ("Họ và tên", "Số điện thoại", "Email") không được để trống.
 - Số điện thoại và email phải đúng định dạng.
 - Nếu dữ liệu không hợp lệ, hệ thống hiển thị thông báo lỗi (ví dụ: "Vui lòng nhập số điện thoại hợp lệ").
6. Nếu dữ liệu hợp lệ, hệ thống lưu thông tin vào cơ sở dữ liệu và gửi thông báo xác nhận cho người dùng: "Yêu cầu tư vấn của bạn đã được gửi thành công. Chúng tôi sẽ liên hệ với bạn sớm!"
7. Hệ thống chuyển thông tin người dùng đến bộ phận tư vấn trực tiếp của HPU để xử lý.
8. Người dùng có thể nhấn nút "Quay lại" để trở về giao diện ChatBot ban đầu nếu không muốn gửi yêu cầu.

Giao diện:

- **Tiêu đề:** "Đăng Ký Tư Vấn" được hiển thị ở phần đầu của form.
- **Các trường nhập liệu:**
 - "Họ và tên": Ô nhập liệu văn bản với placeholder "Họ và tên".
 - "Số điện thoại": Ô nhập liệu văn bản với placeholder "Số điện thoại".
 - "Email": Ô nhập liệu văn bản với placeholder "Email".
 - "Chọn chuyên ngành": Một dropdown với placeholder "Chọn chuyên ngành" và danh sách các chuyên ngành của HPU.
 - "Lời nhắn": Ô nhập liệu văn bản lớn hơn (textarea) với placeholder "Lời nhắn".
- **Nút hành động:**
 - Nút "Gửi yêu cầu": Nút màu xanh dương, nằm bên trái dưới form.
 - Nút "Quay lại": Nút màu xanh dương, nằm bên phải dưới form.

Dữ liệu đầu vào:

- **Họ và tên (chuỗi ký tự):**
 - Bắt buộc, không được để trống.
 - Độ dài tối đa: 50 ký tự.
 - Có thể chứa ký tự Unicode (tiếng Việt), không chứa ký tự đặc biệt (ví dụ: @, #).
- **Số điện thoại (chuỗi ký tự):**
 - Bắt buộc, không được để trống.
 - Phải đúng định dạng số điện thoại Việt Nam (bắt đầu bằng 0, có 10 chữ số, ví dụ: 0987654321).
- **Email (chuỗi ký tự):**
 - Bắt buộc, không được để trống.
 - Phải đúng định dạng email (ví dụ: tennguoidung@domain.com).
- **Chọn chuyên ngành (chuỗi ký tự):**
 - Bắt buộc, phải chọn một giá trị từ danh sách dropdown.

- Danh sách chuyên ngành được lấy từ cơ sở dữ liệu của HPU (ví dụ: "Công nghệ Thông tin", "Quản trị Kinh doanh", "Kỹ thuật Điện", v.v.).
- **Lời nhắn (chuỗi ký tự):**
 - Tùy chọn, có thể để trống.
 - Độ dài tối đa: 500 ký tự.
 - Có thể chứa ký tự Unicode, không chứa nội dung không phù hợp (hệ thống có thể lọc từ ngữ nhạy cảm).

Dữ liệu đầu ra:

- **Thông tin người dùng (đối tượng):**
 - Lưu vào cơ sở dữ liệu với các trường: Họ và tên, Số điện thoại, Email, Chuyên ngành, Lời nhắn, Thời gian gửi yêu cầu.
 - Ví dụ:

```

{
    "name": "Nguyễn Văn A",
    "phone": "0987654321",
    "email": "nguyenvana@gmail.com",
    "major": "Công nghệ Thông tin",
    "message": "Tôi muốn biết thêm về học phí",
    "timestamp": "2025-04-08 11:03:00"
}

```
- **Thông báo xác nhận (chuỗi ký tự):**
 - Hiện thị trên giao diện: "Yêu cầu tư vấn của bạn đã được gửi thành công. Chúng tôi sẽ liên hệ với bạn sớm!"
- **Email thông báo (tùy chọn):**
 - Gửi email xác nhận đến địa chỉ email của người dùng với nội dung tương tự thông báo trên giao diện.

- **Chuyển giao thông tin:**

- Thông tin người dùng được gửi đến hệ thống quản lý của bộ phận tư vấn (ví dụ: qua CRM hoặc email của nhân viên tư vấn).

Quy tắc nghiệp vụ tuân thủ:

- **Kiểm tra tính hợp lệ:**

- Các trường bắt buộc ("Họ và tên", "Số điện thoại", "Email", "Chọn chuyên ngành") không được để trống.
- Số điện thoại phải đúng định dạng (10 chữ số, bắt đầu bằng 0).
- Email phải đúng định dạng (có "@" và tên miền hợp lệ).
- Nếu có lỗi, hệ thống hiển thị thông báo cụ thể (ví dụ: "Vui lòng nhập email hợp lệ").

- **Bảo mật thông tin:**

- Thông tin người dùng phải được mã hóa khi lưu trữ và truyền tải (tuân thủ Nghị định 13 về bảo vệ thông tin cá nhân).
- Không hiển thị thông tin người dùng cho các bên không có quyền truy cập.

- **Tính nhất quán:**

- Danh sách chuyên ngành trong dropdown phải khớp với dữ liệu chính thức của HPU, được cập nhật từ cơ sở dữ liệu.

- **Hiệu suất:**

- Form phải tải và phản hồi trong vòng 1-2 giây để đảm bảo trải nghiệm người dùng mượt mà.
- Quá trình gửi yêu cầu không được vượt quá 3 giây.

- **Khả năng quay lại:**

- Nút "Quay lại" phải hoạt động chính xác, đưa người dùng về giao diện ChatBot ban đầu mà không làm mất ngữ cảnh hội thoại trước đó.

- **Ghi lại lịch sử:**

- Mọi yêu cầu tư vấn phải được ghi lại trong hệ thống (bao gồm thời gian, thông tin người dùng, và trạng thái xử lý) để bộ phận tư vấn theo dõi.

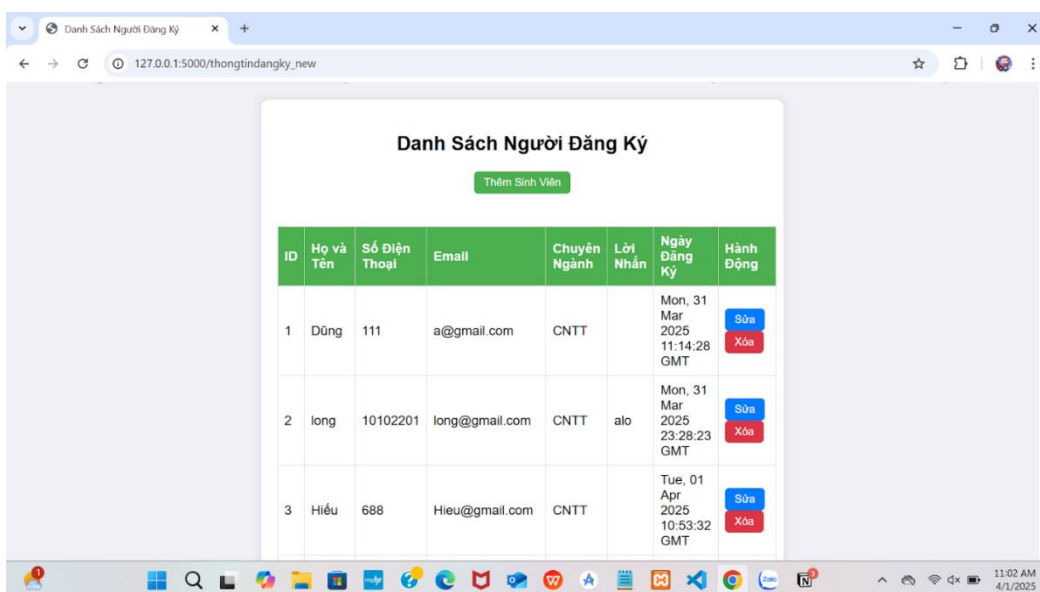
- **Thông báo lỗi rõ ràng:**

- Nếu gửi yêu cầu thất bại (do lỗi hệ thống hoặc kết nối), hệ thống phải hiển thị thông báo: "Có lỗi xảy ra, vui lòng thử lại sau."

Giải thích thêm

Chức năng "Thu thập thông tin liên hệ và đăng ký tư vấn" là một phần quan trọng trong nhóm **Hỗ trợ Tương tác**, giúp ChatBot chuyển giao các yêu cầu phức tạp mà nó không thể xử lý sang nhân viên tư vấn trực tiếp. Giao diện form được thiết kế đơn giản, trực quan, với các trường nhập liệu rõ ràng và các nút hành động ("Gửi yêu cầu", "Quay lại") phù hợp với yêu cầu của dự án về giao diện thân thiện và dễ sử dụng. Chức năng này cũng đáp ứng yêu cầu bảo mật và quyền riêng tư của người dùng, đồng thời đảm bảo tính hiệu quả trong việc chuyển giao thông tin.

c. Hỗ trợ Tương tác: Thu thập thông tin liên hệ và đăng ký tư vấn



ID	Họ và Tên	Số Điện Thoại	Email	Chuyên Ngành	Lỗi Nhân	Ngày Đăng Ký	Hành Động
1	Dũng	111	a@gmail.com	CNTT		Mon, 31 Mar 2025 11:14:28 GMT	Sửa Xóa
2	long	10102201	long@gmail.com	CNTT	alo	Mon, 31 Mar 2025 23:28:23 GMT	Sửa Xóa
3	Hiếu	688	Hieu@gmail.com	CNTT		Tue, 01 Apr 2025 10:53:32 GMT	Sửa Xóa

Hình 6: Danh sách người đăng ký

Quản lý ChatBot & Hệ thống: Giám sát và quản lý danh sách nguồn đăng ký tư vấn

Sự kiện kích hoạt:

- **Khi nào:** Khi quản trị viên đăng nhập vào hệ thống quản lý của ChatBot AI HPU và truy cập vào mục "Danh Sách Nguồn Đăng Ký" trong phần quản lý.
- **Cái gì dẫn đến:** Quản trị viên muốn xem danh sách các yêu cầu tư vấn mà người dùng đã gửi qua ChatBot, nhằm theo dõi, quản lý và phân công công việc cho bộ phận tư vấn trực tiếp.
- **Điều kiện:**
 - Người dùng phải có quyền quản trị viên (Admin) và đang trong phiên làm việc hợp lệ trên hệ thống.
 - Hệ thống phải có ít nhất một yêu cầu tư vấn được gửi từ người dùng qua ChatBot để hiển thị dữ liệu.

Quy trình:

1. Đăng nhập và truy cập:

- Quản trị viên đăng nhập vào hệ thống quản lý của ChatBot AI HPU bằng tài khoản có quyền Admin.
- Từ giao diện chính, quản trị viên điều hướng đến mục "Quản lý ChatBot & Hệ thống" và chọn "Danh Sách Nguồn Đăng Ký".

2. Hiển thị danh sách:

- Hệ thống truy xuất dữ liệu từ cơ sở dữ liệu, lấy danh sách các yêu cầu tư vấn đã được gửi qua ChatBot.
- Danh sách được hiển thị dưới dạng bảng với các cột: ID, Họ và Tên, Số Điện Thoại, Email, Chuyên Ngành, Lời Nhắn, Ngày Đăng Ký, và Hành Động.

3. Xem thông tin chi tiết:

- Quản trị viên có thể xem thông tin chi tiết của từng yêu cầu bằng cách nhấp vào nút "Sửa" (biểu tượng bút chì màu xanh dương) trong cột Hành Động.
- Hệ thống hiển thị một form chỉnh sửa với các trường thông tin tương tự như trong bảng, cho phép quản trị viên cập nhật trạng thái hoặc chỉnh sửa thông tin nếu cần.

4. Xóa yêu cầu:

- Quản trị viên có thể xóa một yêu cầu bằng cách nhấp vào nút "Xóa" (biểu tượng thùng rác màu đỏ) trong cột Hành Động.
- Hệ thống hiển thị hộp thoại xác nhận: "Bạn có chắc chắn muốn xóa yêu cầu này không?" với hai tùy chọn "Xác nhận" và "Hủy".
- Nếu chọn "Xác nhận", hệ thống xóa yêu cầu khỏi cơ sở dữ liệu và cập nhật lại danh sách.
- Nếu chọn "Hủy", hệ thống đóng hộp thoại và không thực hiện thay đổi.

5. Thêm sinh viên (tùy chọn):

- Quản trị viên có thể thêm một sinh viên mới vào danh sách bằng cách nhấp vào nút "Thêm Sinh Viên" ở phía trên bảng.
- Hệ thống hiển thị một form nhập liệu tương tự form đăng ký tư vấn (giống ảnh trước), bao gồm các trường: Họ và Tên, Số Điện Thoại, Email, Chuyên Ngành, và Lời Nhắn.
- Sau khi điền thông tin và nhấn "Gửi", hệ thống lưu thông tin vào cơ sở dữ liệu và cập nhật danh sách.

6. Sắp xếp và lọc (tùy chọn):

- Quản trị viên có thể sắp xếp danh sách theo các cột (ví dụ: theo Ngày Đăng Ký, Họ và Tên) bằng cách nhấp vào tiêu đề cột.
- Hệ thống hỗ trợ lọc dữ liệu (ví dụ: lọc theo chuyên ngành hoặc ngày đăng ký) thông qua các bộ lọc (nếu được tích hợp trong giao diện).

7. Xuất báo cáo (tùy chọn):

- Quản trị viên có thể xuất danh sách dưới dạng file Excel hoặc PDF để báo cáo hoặc lưu trữ bằng cách nhấp vào nút "Xuất Báo Cáo" (nếu có).
- Hệ thống tạo file và cho phép quản trị viên tải xuống.

Giao diện:

- **Tiêu đề:** "Danh Sách Nguồn Đăng Ký" được hiển thị ở phần đầu của giao diện.
- **Nút "Thêm Sinh Viên":** Nút màu xanh lá cây, nằm ở phía trên bên phải của bảng, với nhãn "Thêm Sinh Viên".
- **Bảng dữ liệu:**
 - **Cột ID:** Hiển thị mã định danh duy nhất của yêu cầu (số nguyên, ví dụ: 1, 2, 3).
 - **Cột Họ và Tên:** Hiển thị tên đầy đủ của người gửi yêu cầu (chuỗi ký tự, ví dụ: "Dũng", "long", "Hiếu").
 - **Cột Số Điện Thoại:** Hiển thị số điện thoại của người gửi (chuỗi ký tự, ví dụ: "111", "101202201", "688").
 - **Cột Email:** Hiển thị địa chỉ email của người gửi (chuỗi ký tự, ví dụ: "a@gmail.com", "long@gmail.com", "Hieu@gmail.com").
 - **Cột Chuyên Ngành:** Hiển thị chuyên ngành mà người dùng quan tâm (chuỗi ký tự, ví dụ: "CNTT").
 - **Cột Lời Nhắn:** Hiển thị nội dung lời nhắn của người dùng (chuỗi ký tự, ví dụ: "alo").
 - **Cột Ngày Đăng Ký:** Hiển thị ngày và giờ gửi yêu cầu (định dạng ngày giờ, ví dụ: "Mon, 31 Mar 2025 11:14:28 GMT").
 - **Cột Hành Động:** Chứa hai nút:
 - Nút "Sửa": Biểu tượng bút chì màu xanh dương.
 - Nút "Xóa": Biểu tượng thùng rác màu đỏ.
- **Phân trang (tùy chọn):** Nếu danh sách có nhiều hơn 10 mục, hệ thống hiển thị phân trang (ví dụ: "Trang 1/5", với nút "Trước" và "Sau").

Dữ liệu đầu vào:

- **Thông tin đăng nhập của quản trị viên:**
 - Tên đăng nhập (chuỗi ký tự): Phải khớp với tài khoản Admin trong hệ thống.
 - Mật khẩu (chuỗi ký tự): Phải đúng và được mã hóa trong cơ sở dữ liệu.
- **Dữ liệu yêu cầu tư vấn (từ cơ sở dữ liệu):**
 - ID (số nguyên): Mã định danh duy nhất của yêu cầu, không được trùng lặp.
 - Họ và Tên (chuỗi ký tự): Độ dài tối đa 50 ký tự, có thể chứa ký tự Unicode.
 - Số Điện Thoại (chuỗi ký tự): Định dạng 10 chữ số, bắt đầu bằng 0.
 - Email (chuỗi ký tự): Định dạng email hợp lệ.
 - Chuyên Ngành (chuỗi ký tự): Phải khớp với danh sách chuyên ngành của HPU.
 - Lời Nhắn (chuỗi ký tự): Độ dài tối đa 500 ký tự, có thể để trống.
 - Ngày Đăng Ký (ngày giờ): Định dạng chuẩn (ví dụ: "Mon, 31 Mar 2025 11:14:28 GMT").
- **Dữ liệu nhập khi thêm sinh viên (nếu sử dụng nút "Thêm Sinh Viên"):**
 - Tương tự dữ liệu trong form đăng ký tư vấn (Họ và Tên, Số Điện Thoại, Email, Chuyên Ngành, Lời Nhắn).

Dữ liệu đầu ra:

- **Danh sách yêu cầu tư vấn (danh sách đối tượng):**
[
 { "id": 1, "name": "Dũng", "phone": "111", "email": "a@gmail.com",
 "major": "CNTT", "message": "", "timestamp": "Mon, 31 Mar 2025 11:14:28 GMT" },
]

```
{ "id": 2, "name": "long", "phone": "101202201", "email":  
"long@gmail.com", "major": "CNTT", "message": "alo", "timestamp":  
"Mon, 31 Mar 2025 23:28:26 GMT" },  
{ "id": 3, "name": "Hiếu", "phone": "688", "email": "Hieu@gmail.com",  
"major": "CNTT", "message": "", "timestamp": "Tue, 01 Apr 2025  
10:53:32 GMT" }  
]
```

- **Thông báo xác nhận (chuỗi ký tự):**
 - Khi xóa: "Yêu cầu đã được xóa thành công."
 - Khi thêm: "Thêm sinh viên thành công."
 - Khi sửa: "Cập nhật thông tin thành công."
- **Thông báo lỗi (chuỗi ký tự):**
 - Nếu không có dữ liệu: "Hiện tại không có yêu cầu tư vấn nào."
 - Nếu xóa thất bại: "Có lỗi xảy ra khi xóa yêu cầu, vui lòng thử lại."
- **File xuất báo cáo (tùy chọn):**
 - File Excel hoặc PDF chứa danh sách yêu cầu tư vấn với các cột tương tự bảng.

Quy tắc nghiệp vụ tuân thủ:

- **Quyền truy cập:**
 - Chỉ quản trị viên (Admin) có quyền truy cập vào chức năng này. Hệ thống kiểm tra quyền truy cập trước khi hiển thị giao diện.
 - Nếu người dùng không có quyền, hệ thống hiển thị thông báo: "Bạn không có quyền truy cập vào chức năng này."
- **Bảo mật thông tin:**
 - Thông tin cá nhân của người dùng (Họ và Tên, Số Điện Thoại, Email) phải được mã hóa khi lưu trữ và truyền tải, tuân thủ Nghị định 13 về bảo vệ thông tin cá nhân.
 - Chỉ quản trị viên có quyền xem và chỉnh sửa thông tin trong danh sách.

- **Tính nhất quán dữ liệu:**
 - Dữ liệu hiển thị trong bảng phải khớp với dữ liệu trong cơ sở dữ liệu, không được có sự sai lệch.
 - Chuyên ngành hiển thị phải nằm trong danh sách chuyên ngành chính thức của HPU.
- **Hiệu suất:**
 - Hệ thống phải tải danh sách trong vòng 2-3 giây, ngay cả khi có hàng trăm yêu cầu.
 - Các thao tác sửa, xóa, hoặc thêm phải hoàn thành trong vòng 1-2 giây.
- **Xử lý lỗi:**
 - Nếu không có dữ liệu, hệ thống hiển thị thông báo: "Hiện tại không có yêu cầu tư vấn nào."
 - Nếu có lỗi khi truy xuất dữ liệu (do kết nối cơ sở dữ liệu), hệ thống hiển thị: "Có lỗi xảy ra khi tải danh sách, vui lòng thử lại sau."
 - Nếu xóa thất bại (do lỗi hệ thống), hệ thống hiển thị: "Có lỗi xảy ra khi xóa yêu cầu, vui lòng thử lại."
- **Ghi lại lịch sử thao tác:**
 - Mọi thao tác (thêm, sửa, xóa) phải được ghi lại trong nhật ký hệ thống, bao gồm: thời gian, người thực hiện (ID quản trị viên), và hành động.
 - Ví dụ: "Admin ID: 001 đã xóa yêu cầu ID: 1 vào lúc 2025-04-08 11:05:00."
- **Khả năng mở rộng:**
 - Hệ thống phải hỗ trợ hiển thị danh sách lớn (hàng nghìn yêu cầu) bằng cách sử dụng phân trang hoặc tải dữ liệu theo từng phần (lazy loading).
 - Hỗ trợ thêm các bộ lọc (theo chuyên ngành, ngày đăng ký) và sắp xếp (theo cột) để quản trị viên dễ dàng quản lý.

- **Tính trực quan:**

- Bảng dữ liệu phải được thiết kế rõ ràng, dễ đọc, với các cột có độ rộng hợp lý.
- Các nút hành động (Sửa, Xóa) phải có màu sắc và biểu tượng dễ nhận biết (xanh dương cho Sửa, đỏ cho Xóa).

- **Khả năng phục hồi:**

- Nếu hệ thống gặp sự cố trong quá trình xóa hoặc sửa, dữ liệu phải được bảo toàn và không bị mất mát.
- Hệ thống phải có cơ chế sao lưu tự động để khôi phục dữ liệu nếu cần.

Tính năng bổ sung (tùy chọn):

- **Tìm kiếm nhanh:** Hệ thống có thể tích hợp một ô tìm kiếm phía trên bảng, cho phép quản trị viên tìm kiếm theo Họ và Tên, Số Điện Thoại, hoặc Email.
 - Dữ liệu đầu vào: Chuỗi ký tự tìm kiếm (ví dụ: "Dũng").
 - Dữ liệu đầu ra: Danh sách các yêu cầu khớp với từ khóa (ví dụ: chỉ hiển thị mục có Họ và Tên là "Dũng").
- **Thông kê:** Hệ thống có thể hiển thị số liệu thống kê phía trên bảng, ví dụ: "Tổng số yêu cầu: 3", "Số yêu cầu trong ngày: 1".
- **Gửi thông báo:** Quản trị viên có thể chọn một yêu cầu và gửi thông báo trực tiếp đến người dùng qua email (ví dụ: "Chúng tôi đã nhận được yêu cầu của bạn và sẽ liên hệ trong 24 giờ tới").

Giải thích thêm:

Chức năng "Giám sát và quản lý danh sách nguồn đăng ký tư vấn" là một phần quan trọng trong nhóm **Quản lý ChatBot & Hệ thống**, giúp quản trị viên theo dõi và xử lý các yêu cầu tư vấn từ người dùng một cách hiệu quả. Giao diện bảng được thiết kế rõ ràng, với các cột thông tin đầy đủ và các nút hành động (Sửa, Xóa) trực quan, phù hợp với yêu cầu của dự án về tính dễ sử dụng và hiệu quả

quản lý. Chức năng này cũng đáp ứng các yêu cầu về bảo mật, hiệu suất, và khả năng mở rộng, đồng thời cung cấp các tính năng bổ sung như tìm kiếm, thống kê, và xuất báo cáo để hỗ trợ quản trị viên trong công việc.

2.2.1.2. Công nghệ sử dụng

Thành phần	Công nghệ	Vai trò
Frontend	HTML, JS	Giao tiếp người dùng
Backend	Flask (Python)	API tiếp nhận và xử lý câu hỏi
AI	Gemini 1.5 Flash	Sinh phản hồi nếu không tìm thấy trong data
Cơ sở dữ liệu	JSON (file-based), Supabase (đăng ký người dùng)	Lưu thông tin, lịch sử và bộ huấn luyện
Hosting	Vercel (frontend), Render (backend)	Triển khai ChatBot thực tế

Bảng các công nghệ được sử dụng trong ChatBot

Trong quá trình xây dựng phần mềm ChatBot AI cho trường Đại học Quản lý và Công nghệ Hải Phòng (HPU), em đã lựa chọn và triển khai một tập hợp các công nghệ hiện đại và phù hợp để đảm bảo hiệu suất, khả năng mở rộng, tính bảo mật và trải nghiệm người dùng tốt nhất. Dưới đây là chi tiết về các công nghệ đã được sử dụng:

1. Giao diện người dùng (Front-end):

- **HTML (HyperText Markup Language):** Được sử dụng làm ngôn ngữ đánh dấu cấu trúc cơ bản cho giao diện webchat của ChatBot. HTML chịu trách nhiệm cho việc tạo nên bộ khung và bố cục của các thành phần hiển thị như tin nhắn chào hỏi, ô nhập liệu, nút gửi và nút đăng ký tư vấn.

- **CSS (Cascading Style Sheets):** Đóng vai trò trong việc định hình phong cách hiển thị của giao diện, bao gồm màu sắc (ví dụ: bóng chat màu xám nhạt, nút màu xanh dương), font chữ, kích thước và vị trí của các thành phần. CSS đảm bảo giao diện trực quan, thân thiện và nhất quán với nhận diện thương hiệu của HPU.
- **JavaScript:** Là ngôn ngữ lập trình phía client, được sử dụng để thêm tính tương tác và động cho giao diện webchat. JavaScript xử lý các sự kiện như người dùng nhập câu hỏi, nhấp vào nút gửi hoặc đăng ký tư vấn. Nó cũng có thể được sử dụng để gửi yêu cầu đến back-end và hiển thị phản hồi từ ChatBot một cách mượt mà.
- **ReactJS:** Một thư viện JavaScript hiện đại như **ReactJS** có khả năng cao đã được sử dụng để xây dựng giao diện người dùng phức tạp và có tính tương tác cao cho ChatBot. ReactJS cung cấp các thành phần giao diện tái sử dụng, quản lý trạng thái hiệu quả và tăng tốc quá trình phát triển. Việc sử dụng ReactJS giúp tạo ra một trải nghiệm người dùng linh hoạt và đáp ứng nhanh chóng.
- **Fetch API hoặc Axios:** Để tích hợp front-end với back-end API, **Fetch API** (tích hợp sẵn trong trình duyệt) hoặc một thư viện HTTP client phổ biến như **Axios** đã được sử dụng. Các công nghệ này cho phép JavaScript gửi các yêu cầu HTTP đến back-end (ví dụ: để gửi câu hỏi của người dùng và nhận phản hồi từ ChatBot) và xử lý các dữ liệu trả về.

2. Hệ thống phía máy chủ (Back-end):

- **Python:** Với khả năng làm việc hiệu quả với các thư viện và API liên quan đến trí tuệ nhân tạo, **Python** đã được lựa chọn làm ngôn ngữ lập trình chính cho back-end của hệ thống ChatBot AI. Python cung cấp một môi trường phát triển linh hoạt và mạnh mẽ để xử lý logic nghiệp vụ, kết nối với các mô hình AI và quản lý dữ liệu.

- **Flask / Django:** Một trong hai framework web phổ biến của Python, **Flask** (microframework) hoặc **Django** (framework đầy đủ tính năng), có khả năng đã được sử dụng để xây dựng API back-end cho ChatBot. Framework này chịu trách nhiệm cho việc định nghĩa các điểm cuối API (endpoints) mà front-end có thể giao tiếp, xử lý các yêu cầu từ người dùng, tương tác với mô hình AI và trả về phản hồi.
- **Quản lý cơ sở dữ liệu:** Để lưu trữ dữ liệu liên quan đến ChatBot (ví dụ: kịch bản hội thoại được lập trình sẵn, lịch sử tương tác - nếu có thu thập), một hệ thống quản lý cơ sở dữ liệu (CSDL) đã được sử dụng. Các tùy chọn CSDL bao gồm **MySQL**, **PostgreSQL**, hoặc một CSDL NoSQL như **MongoDB**. CSDL đảm bảo dữ liệu được lưu trữ một cách an toàn, có thể truy vấn và quản lý hiệu quả.

3. Mô hình Trí tuệ Nhân tạo (AI Model):

- **Google Gemini API:** Khi đề cập đến việc tương tác với các API bên ngoài, **Google Gemini API** đã được tích hợp để cung cấp khả năng xử lý ngôn ngữ tự nhiên cho ChatBot. Gemini API cho phép ChatBot hiểu câu hỏi của người dùng, phân tích ngữ cảnh và tạo ra các câu trả lời phù hợp dựa trên dữ liệu được huấn luyện. Việc sử dụng Gemini API giúp ChatBot có khả năng phản hồi thông minh và linh hoạt hơn so với các kịch bản được lập trình sẵn.
- **Dữ liệu huấn luyện (Trained Data):** Dựa trên tên file **trained_data.json.pdf**, có thể thấy rằng dữ liệu huấn luyện đóng vai trò quan trọng trong việc giúp ChatBot hiểu và trả lời các câu hỏi liên quan đến HPU. Dữ liệu này có thể bao gồm các câu hỏi thường gặp và câu trả lời tương ứng về chỉ tiêu tuyển sinh, ngành học, học phí, học bổng, ký túc xá, thông tin liên hệ và nhiều chủ đề khác. Định dạng JSON được sử dụng để cấu trúc dữ liệu này một cách có tổ chức, giúp cho việc huấn luyện mô hình AI hiệu quả hơn.

4. Tích hợp và Triển khai:

- **Tích hợp API vào Website:** Như đã đề cập ở phần Front-end, các API từ back-end (kết nối với mô hình AI) đã được tích hợp vào trang web của HPU thông qua các kỹ thuật lập trình web. Điều này cho phép người dùng tương tác với ChatBot trực tiếp trên giao diện webchat.
- **Môi trường triển khai:** Ứng dụng ChatBot AI có khả năng cao đã được triển khai trên hệ thống CNTT hiện đại của HPU, bao gồm máy chủ web và các dịch vụ liên quan. Việc này đảm bảo ChatBot có thể hoạt động ổn định và đáp ứng yêu cầu của người dùng.

5. Bảo mật:

- **Mã hóa thông tin:** Để tuân thủ Nghị định 13 về bảo vệ thông tin cá nhân, các thông tin nhạy cảm (nếu có thu thập) và thông tin trong các phiên trò chuyện giữa người dùng và ChatBot đã được mã hóa khi lưu trữ và truyền tải.
- **Ẩn danh thông tin nhạy cảm:** Khi tương tác với các API bên ngoài như Gemini API, các thông tin nhạy cảm cần được ẩn danh để bảo vệ quyền riêng tư của người dùng.

6. Kiểm thử:

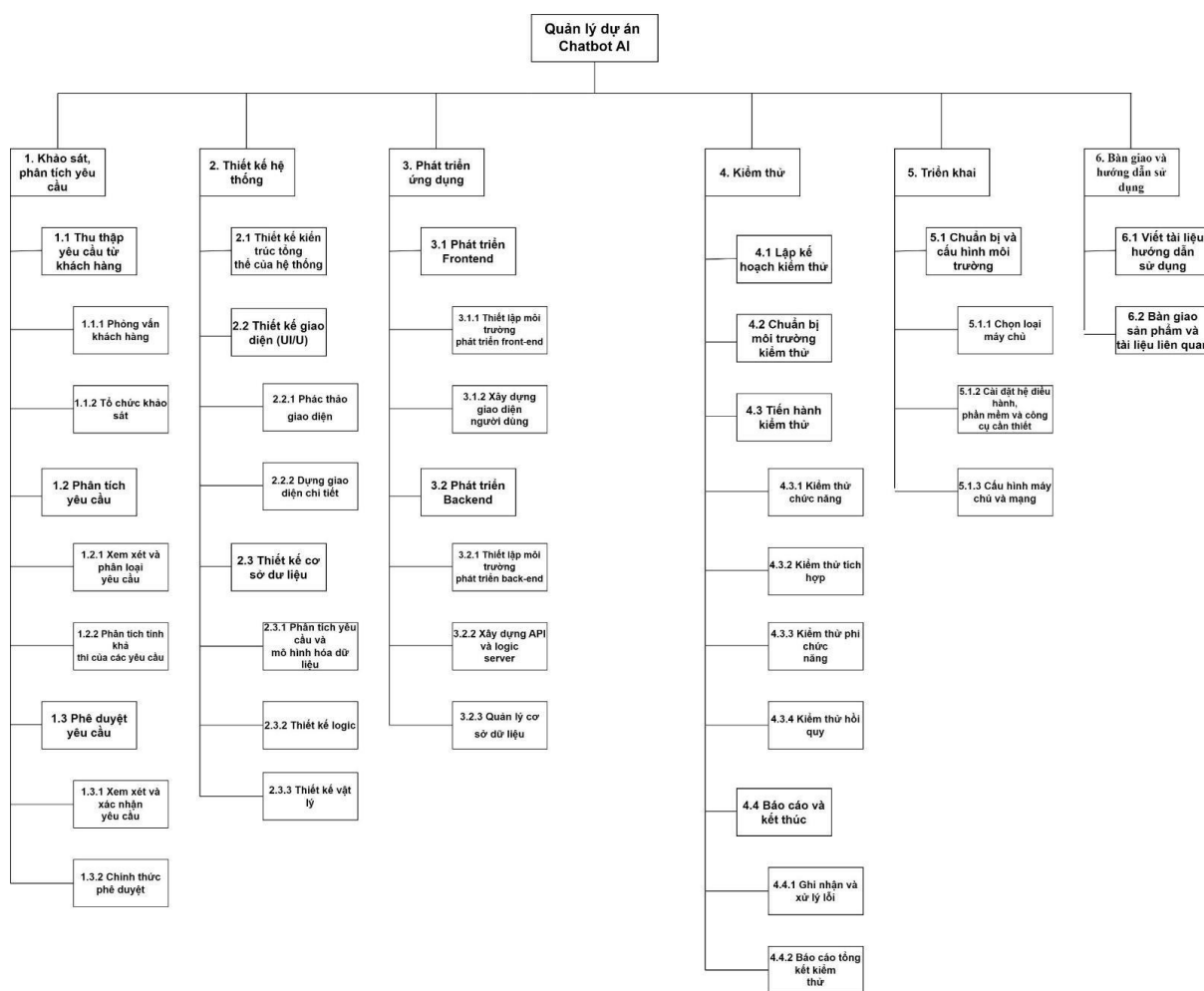
- Các phương pháp kiểm thử phần mềm đã được áp dụng trong suốt quá trình phát triển để đảm bảo ChatBot hoạt động chính xác, hiệu quả và đáp ứng các yêu cầu đã đặt ra. Điều này bao gồm kiểm thử chức năng (đảm bảo ChatBot trả lời đúng các câu hỏi), kiểm thử hiệu suất (đảm bảo thời gian phản hồi nhanh chóng), và kiểm thử bảo mật (đảm bảo an toàn dữ liệu người dùng).

Việc kết hợp các công nghệ này đã cho phép xây dựng một hệ thống ChatBot AI mạnh mẽ, có khả năng hỗ trợ tư vấn tuyển sinh hiệu quả cho trường Đại học Quản lý và Công nghệ Hải Phòng, đáp ứng nhu cầu thông tin đa dạng của học sinh, phụ huynh và các đối tượng liên quan.

2.2.1.3. Quy trình xây dựng ChatBot

Bảng chi tiết công việc WBS.

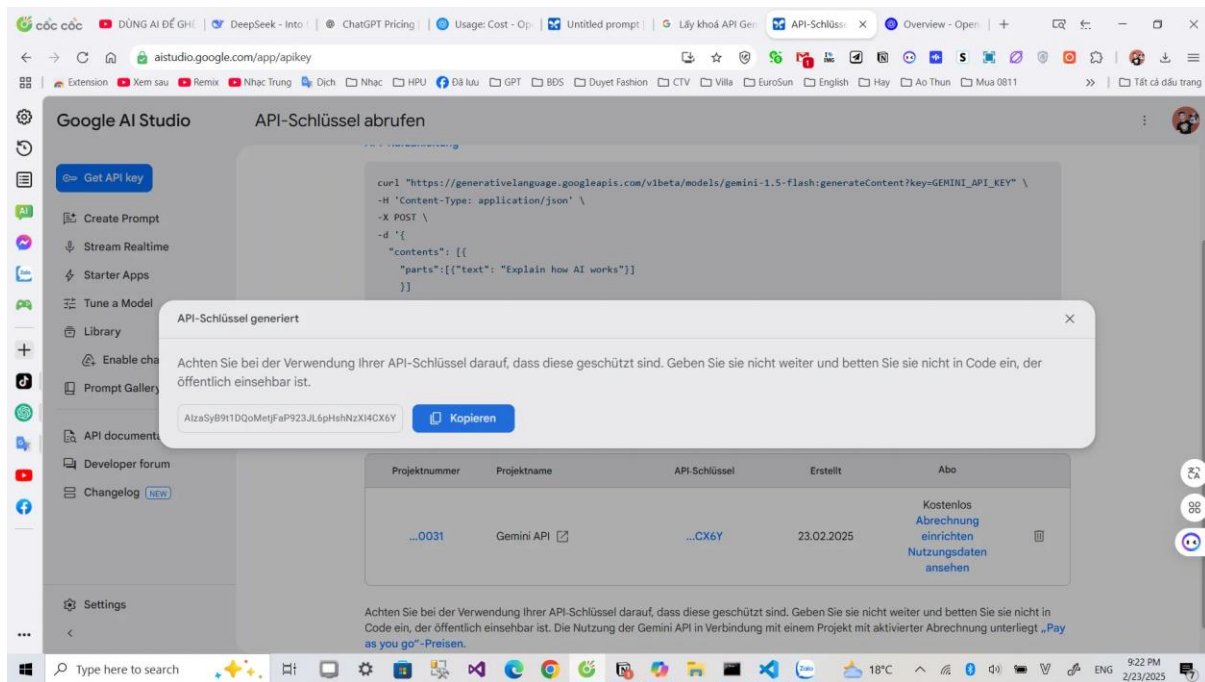
WBS (Work Breakdown Structure), hay **cấu trúc phân rã công việc**, là một phương pháp phân chia một dự án lớn thành các phần nhỏ hơn, để quản lý hơn.



Hình 7: Sơ đồ phân rã công việc

a. Đăng ký tài khoản Google AI Studio

- Truy cập <https://aistudio.google.com> và đăng ký tài khoản.
- Lấy khoá Gemini API trong Google AI Studio
- `API_KEY = "AIzaSyB9t1DQoMetjFaP923JL6pHshNzXI4CX6Y"`



Hình 8: Đăng ký tài khoản Google AI Studio

a) Chuẩn hóa dữ liệu tuyển sinh

Em đã tiến hành thu thập, phân loại và chuẩn hóa dữ liệu tuyển sinh từ các nguồn chính thức của nhà trường, đặc biệt là từ website <https://hpu.edu.vn>. Dữ liệu được đưa về định dạng JSON với cấu trúc "Bạn hỏi" – "HPU trả lời", phục vụ cho việc tìm kiếm nội dung tương đồng.

b) Xây dựng luồng hội thoại

Luồng xử lý chính của ChatBot như sau:

1. Người dùng nhập câu hỏi;
2. Câu hỏi được chuẩn hóa (loại bỏ dấu, viết thường, lọc stop word);
3. Nếu câu hỏi có từ khóa liên quan đến thông tin tuyển sinh → tìm trong `trained_data.json`;
4. Nếu không tìm thấy, gọi API Gemini để sinh câu trả lời;
5. Lưu lại hội thoại vào `chat_history.json`.

c) Lọc đầu vào và quyết định hướng xử lý

Một cơ chế lọc câu hỏi đầu vào đã được xây dựng để:

- Loại bỏ các câu hỏi không liên quan (ví dụ: "bạn ăn cơm chưa");

- Nhận diện các từ khóa đặc trưng về tuyển sinh (ví dụ: “ngành học”, “học phí”, “xét tuyển”);
- Quyết định có nên dùng dữ liệu nội bộ hay gọi Gemini.

d) Tích hợp API Gemini

Nếu câu hỏi không nằm trong bộ dữ liệu, hệ thống sẽ gọi đến mô hình **Gemini 1.5 Flash** thông qua Google Generative AI API. Câu hỏi được gửi kèm ngữ cảnh mặc định để mô hình hiểu rằng nó đang tư vấn cho HPU. Kết quả sinh ra được cache vào file để tránh gọi lại nhiều lần.

2.2.1.4. Phân tích API của ChatBot

Phân tích các API (Application Programming Interfaces - Giao diện Lập trình Ứng dụng) để tạo ra chương trình nghiên cứu ChatBot AI

Khám phá sâu hơn về các API đã được sử dụng, cách chúng hoạt động và tầm quan trọng của chúng trong việc xây dựng một ChatBot AI hiệu quả.

Hiểu rõ hơn về cách chúng tương tác và hỗ trợ lẫn nhau.

a) Google Gemini API

Nhập thư viện và thiết lập API Key

Việc nhập thư viện **google.generativeai** và thiết lập API Key là bước quan trọng để xác thực và sử dụng Google Gemini API. Sử dụng thư viện **dotenv** giúp quản lý API key một cách an toàn, tránh việc nhúng trực tiếp vào mã nguồn.

- Nhập thư viện **google.generativeai** để tương tác với các mô hình Generative AI như Gemini.
- Sử dụng **dotenv** để đọc các biến môi trường từ file **.env**.
- Cấu hình Gemini API bằng API key.

```
import google.generativeai as genai
from dotenv import load_dotenv
import os
```

```
# Load biến môi trường từ file.env
load_dotenv(dotenv_path=".env")
GEMINI_API_KEY = os.getenv("GEMINI_API_KEY")

if not GEMINI_API_KEY:
    raise ValueError("API Key không được tìm thấy trong .env")

# Cấu hình Gemini API
genai.configure(api_key=GEMINI_API_KEY)
```

PHÂN TÍCH CODE

- `import google.generativeai as genai`: Dòng này **nhập (import) thư viện google.generativeai** và **đặt bí danh (alias) cho nó là genai**. Điều này cho phép bạn gọi các hàm và lớp từ thư viện này bằng cách sử dụng tiền tố `genai`, một cách ngắn gọn hơn. Thư viện này là **SDK Python của Google để tương tác với các mô hình Generative AI như Gemini**.
- `from dotenv import load_dotenv`: Dòng này **nhập hàm load_dotenv từ thư viện dotenv**. Thư viện `dotenv` thường được sử dụng để **đọc các biến môi trường từ một file có tên .env**. Đây là một cách phổ biến để **quản lý các cấu hình bí mật** như API key mà không cần phải nhúng trực tiếp chúng vào mã nguồn.
- `import os`: Dòng này **nhập module os của Python**. Module `os` cung cấp nhiều hàm để **tương tác với hệ điều hành**, bao gồm cả việc **truy cập các biến môi trường**.
- `load_dotenv(dotenv_path=".env")`: Dòng này **gọi hàm load_dotenv()**, được nhập ở trên, với tham số `dotenv_path=".env"`. Điều này có nghĩa là chương trình sẽ **cố gắng tìm và đọc file .env trong cùng thư mục với file Python hiện tại**. Nếu tìm thấy, nó sẽ **tải các biến được định nghĩa trong file này thành các biến môi trường** của hệ thống.
- `GEMINI_API_KEY = os.getenv("GEMINI_API_KEY")`: Dòng này **sử dụng hàm os.getenv() để lấy giá trị của biến môi trường có tên là "GEMINI_API_KEY"**. Biến môi trường này **được kỳ vọng đã được**

thiết lập bởi file .env (nếu `load_dotenv()` thành công) hoặc đã được thiết lập trong môi trường hệ thống. Giá trị lấy được (là **API key cần thiết để xác thực và sử dụng Google Gemini API**) sẽ được gán cho biến **GEMINI_API_KEY**.

- `if not GEMINI_API_KEY::` Dòng này **kiểm tra xem biến GEMINI_API_KEY có giá trị hay không**. Trong Python, một biến rỗng (ví dụ: `None`, chuỗi rỗng `""`) sẽ được coi là `False` trong ngữ cảnh logic. Vì vậy, dòng này kiểm tra xem **nếu GEMINI_API_KEY không có giá trị**.
- `raise ValueError("API Key không được tìm thấy trong .env"):` Nếu điều kiện ở dòng trên là đúng (tức là `GEMINI_API_KEY` không có giá trị), dòng này sẽ **gây ra một ngoại lệ (exception) có kiểu ValueError với thông báo "API Key không được tìm thấy trong .env"**. Điều này có nghĩa là chương trình sẽ **dừng thực thi và hiển thị thông báo lỗi**, cho biết rằng **API key cần thiết không được tìm thấy**, có thể là do file `.env` không tồn tại hoặc không chứa biến `GEMINI_API_KEY`, hoặc biến môi trường chưa được thiết lập.
- `genai.configure(api_key=GEMINI_API_KEY):` Dòng này **gọi hàm configure() từ thư viện genai (Google Generative AI)**. Hàm này được sử dụng để **thiết lập các cấu hình chung cho việc sử dụng API**, và trong trường hợp này, nó được **truyền vào API key (GEMINI_API_KEY)** đã được lấy từ biến môi trường. **Việc cấu hình API key là bắt buộc** để chương trình có thể xác thực và gửi yêu cầu đến Google Gemini API.

Hàm gọi API Gemini (ask_gemini_v2)

Hàm `ask_gemini_v2` cho phép gửi câu hỏi đến Gemini API và nhận phản hồi. Lịch sử hội thoại được duy trì để cung cấp ngữ cảnh cho các câu trả lời. System prompt giúp định hình cách mô hình phản hồi, hướng dẫn về vai trò và mục đích của nó.

- Định nghĩa hàm **ask_gemini_v2** để gửi câu hỏi và nhận phản hồi từ Gemini API.
- Sử dụng khối **try-except** để xử lý lỗi khi gọi API.
- Duy trì lịch sử hội thoại để Gemini đưa ra các câu trả lời phù hợp.

Gọi API Gemini với lịch sử hội thoại

```
def ask_gemini_v2(question, history=None):
    try:
        # Thêm system prompt với thông tin cập nhật về HPU
        model = genai.GenerativeModel(
            "Gemini 1.5 Flash",
            system_instruction="Bạn là ChatBot HPU, được tạo ra để hỗ trợ sinh viên của trường Đại học Quản lý và Công nghệ Hải Phòng."
        )
        if history is None:
            history = load_json_data(CHAT_HISTORY_FILE)
        # Định dạng lịch sử hội thoại theo đúng yêu cầu của Gemini API
        formatted_history = []
        for msg in history:
            if "question" not in msg or "answer" not in msg:
                logger.warning(f"Bỏ qua mục không hợp lệ trong lịch sử hội thoại: {msg}")
                continue
            formatted_history.append({
                "role": "user",
                "parts": [{"text": msg["question"]}]}
            })
            formatted_history.append({
                "role": "assistant",
                "parts": [{"text": msg["answer"]}]}
            })
        chat = model.start_chat(history=formatted_history)
        response = chat.send_message(question)
        return response.text.strip() if response else "Không có phản hồi từ Gemini."
    except Exception as e:
        logger.error(f"Lỗi khi gọi Gemini API: {str(e)}")
        return f"Lỗi khi gọi Gemini API: {str(e)}"
```

PHÂN TÍCH CODE

- `def ask_gemini_v2(question, history=None)::` Dòng này **định nghĩa một hàm có tên là ask_gemini_v2**. Hàm này **nhận hai tham số**:
 - `question`: Đây là **câu hỏi (dạng chuỗi)** mà người dùng muốn gửi đến Gemini API.
 - `history=None`: Đây là **lịch sử hội thoại (dạng danh sách)**. Tham số này có **giá trị mặc định là None**, có nghĩa là nếu không có lịch sử nào được truyền vào khi gọi hàm, nó sẽ mặc định là None. Lịch sử hội thoại này rất quan trọng để Gemini có thể đưa ra các câu trả lời phù hợp với ngữ cảnh của cuộc trò chuyện.
- `try::` Dòng này **bắt đầu một khối mã try**. Khối mã này chứa các thao tác mà chương trình sẽ cố gắng thực hiện. Nếu **xảy ra lỗi (ngoại lệ)** trong khối mã này, chương trình sẽ **chuyển sang khối mã except** để xử lý lỗi thay vì dừng đột ngột.
- `model = genai.GenerativeModel(...):` Dòng này **tạo một đối tượng (instance) của lớp GenerativeModel từ thư viện genai**. Đối tượng này đại diện cho **mô hình Generative AI mà bạn muốn sử dụng**, trong trường hợp này là "Gemini 1.5 Flash" [lưu ý rằng tên mô hình có thể thay đổi theo thời gian và phiên bản API]. Tham số `system_instruction` được truyền vào để **cung cấp một "system prompt"** cho mô hình, hướng dẫn về vai trò và mục đích của nó. Ở đây, Gemini được hướng dẫn là "Bạn là ChatBot HPU, được tạo ra để hỗ trợ sinh viên của trường Đại học Quản lý và Công nghệ Hải Phòng.". **System prompt giúp định hình cách mô hình phản hồi**.
- `if history is None::` Dòng này **kiểm tra xem tham số history có phải là None hay không**.
- `history = load_json_data(CHAT_HISTORY_FILE):` Nếu `history` là None (tức là không có lịch sử nào được truyền vào), dòng này sẽ **gọi hàm load_json_data()** (chúng ta không thấy định nghĩa của hàm này trong đoạn

mã, nhưng dựa vào tên, có thể đoán rằng nó **đọc dữ liệu lịch sử hội thoại từ một file JSON** có tên được lưu trong biến `CHAT_HISTORY_FILE`). Điều này cho thấy rằng nếu không có lịch sử cụ thể nào được cung cấp, ChatBot sẽ cố gắng **tải lịch sử đã lưu trước đó**.

- `formatted_history = []`: Dòng này **khởi tạo một danh sách rỗng có tên là `formatted_history`**. Danh sách này sẽ được sử dụng để **lưu trữ lịch sử hội thoại theo định dạng mà Gemini API mong đợi**.
- `for msg in history::` Dòng này **bắt đầu một vòng lặp for để duyệt qua từng tin nhắn (msg) trong danh sách history**.
- `if "question" not in msg or "answer" not in msg::` Dòng này **kiểm tra xem mỗi tin nhắn (msg) trong lịch sử có chứa cả khóa "question" và khóa "answer" hay không**. Điều này nhằm đảm bảo rằng **các mục trong lịch sử có định dạng đúng** (chứa cả câu hỏi của người dùng và câu trả lời của ChatBot).
- `logger.warning(f'Bỏ qua mục không hợp lệ trong lịch sử hội thoại: {msg})`: Nếu một tin nhắn trong lịch sử không có cả "question" và "answer", dòng này sẽ **sử dụng một đối tượng logger để ghi một cảnh báo (warning)**, thông báo rằng **một mục không hợp lệ đã bị bỏ qua** và in ra nội dung của mục đó. Điều này giúp **đảm bảo rằng chỉ các tin nhắn có định dạng đúng mới được sử dụng**.
- `continue`: Nếu một tin nhắn không hợp lệ bị phát hiện, câu lệnh `continue` sẽ **bỏ qua phần còn lại của vòng lặp cho tin nhắn hiện tại và chuyển sang tin nhắn tiếp theo**.
- `formatted_history.append(...)`: Hai dòng `formatted_history.append(...)` này được sử dụng để **thêm các tin nhắn vào danh sách `formatted_history` theo định dạng mà Gemini API yêu cầu**. Mỗi tin nhắn bao gồm:
 - "role": Chỉ định vai trò của người gửi, có thể là "user" (người dùng đặt câu hỏi) hoặc "assistant" (ChatBot trả lời).

- "parts": Một danh sách chứa các phần của tin nhắn. Trong trường hợp này, mỗi tin nhắn chỉ có một phần là văn bản, được chứa trong một dictionary với khóa "text" và giá trị là câu hỏi hoặc câu trả lời tương ứng.
- chat = model.start_chat(history=formatted_history): Dòng này **gọi phương thức start_chat() trên đối tượng model**. Phương thức này **khởi tạo một phiên trò chuyện với mô hình**, và nó được **truyền vào lịch sử hội thoại đã được định dạng (formatted_history)**. Đối tượng chat được trả về đại diện cho phiên trò chuyện này.
- response = chat.send_message(question): Dòng này **gửi câu hỏi hiện tại (question) đến mô hình thông qua phương thức send_message() của đối tượng chat**. Mô hình sẽ **xử lý câu hỏi dựa trên lịch sử hội thoại (nếu có) và đưa ra phản hồi**. Phản hồi được trả về và gán cho biến response.
- return response.text.strip() if response else "Không có phản hồi từ Gemini.": Dòng này **xử lý phản hồi từ Gemini**.
 - if response: Kiểm tra xem có nhận được phản hồi hợp lệ hay không (tức là response không phải là None).
 - response.text: Nếu có phản hồi, thuộc tính .text của đối tượng response chứa **văn bản câu trả lời của Gemini**.
 - .strip(): Phương thức này được gọi để **loại bỏ bất kỳ khoảng trắng thừa nào ở đầu và cuối chuỗi câu trả lời**.
 - else "Không có phản hồi từ Gemini.": Nếu không nhận được phản hồi (response là None), hàm sẽ **trả về chuỗi "Không có phản hồi từ Gemini."**
- except Exception as e:: Dòng này **bắt đầu khối mã except**. Khối mã này sẽ được thực thi nếu **xảy ra bất kỳ loại ngoại lệ nào (Exception)** trong khối mã try. Biến e sẽ chứa thông tin về ngoại lệ đã xảy ra.
- logger.error(f'Lỗi khi gọi Gemini API: {str(e)}'): Dòng này **sử dụng logger để ghi một thông báo lỗi (error)**, cho biết rằng **đã xảy ra lỗi khi**

gọi Gemini API và in ra **thông tin chi tiết về lỗi (str(e))**. Việc ghi log lỗi rất quan trọng để **theo dõi và gỡ lỗi** cho ứng dụng.

- `return f"Lỗi khi gọi Gemini API: {str(e)}"`: Dòng này **trả về một chuỗi thông báo lỗi**, cho biết rằng **đã xảy ra lỗi khi gọi Gemini API** và bao gồm **thông tin chi tiết về lỗi**. Điều này cho phép người dùng hoặc phần còn lại của chương trình biết rằng **yêu cầu đến Gemini đã không thành công** và cung cấp thông tin về nguyên nhân có thể.

b) Các Endpoint Web của Flask

Endpoint /

```
@app.route("/")
```

```
def home():
```

```
    return render_template("index.html")
```

Phục vụ trang chủ của ứng dụng web.

PHÂN TÍCH CODE

Đoạn mã này định nghĩa một **route (đường dẫn)** / cho ứng dụng Flask. Khi người dùng truy cập địa chỉ gốc của ứng dụng web, hàm `home()` sẽ được gọi. Hàm này sử dụng `render_template("index.html")` để trả về nội dung của file `index.html`, đây là trang chủ của ChatBot HPU.

Endpoint /ask (với phương thức POST)

```
@app.route("/ask", methods=["POST"])
def ask():
    try:
        user_question = request.json.get("question")
        if not user_question:
            return jsonify({"error": "Không có câu hỏi được cung cấp."}), 400

        if not trained_data:
            logger.warning("Không có dữ liệu huấn luyện, thử gọi Gemini API...")
            answer = ask_gemini_v2(user_question)
            save_chat_history(user_question, answer)
            return jsonify({"answer": answer})

        answer = get_answer(user_question, trained_data)
        return jsonify({"answer": answer})
    except Exception as e:
        logger.error(f"Lỗi khi xử lý yêu cầu: {str(e)}")
        return jsonify({"error": f"Lỗi khi xử lý yêu cầu: {str(e)}"}), 500
```

Nhận yêu cầu POST từ người dùng (chứa câu hỏi) và trả về phản hồi từ ChatBot. Kiểm tra dữ liệu huấn luyện và gọi **ask_gemini_v2** hoặc **get_answer** để lấy câu trả lời.

PHÂN TÍCH CODE

Đoạn mã này định nghĩa route /ask và chỉ chấp nhận các yêu cầu với phương thức POST. Khi một yêu cầu POST được gửi đến endpoint này, hàm ask() sẽ được gọi. Hàm này lấy câu hỏi từ dữ liệu JSON của request (request.json.get("question")). Sau đó, nó kiểm tra xem có dữ liệu huấn luyện (trained_data) hay không. Nếu không có, nó sẽ gọi hàm ask_gemini_v2 để lấy câu trả lời từ Gemini API. Nếu có dữ liệu huấn luyện, nó sẽ gọi hàm get_answer để tìm kiếm câu trả lời trong dữ liệu huấn luyện hoặc lịch sử trò chuyện trước khi gọi Gemini API. Cuối cùng, nó trả về câu trả lời dưới dạng JSON (jsonify({"answer": answer})). Nếu có bất kỳ lỗi nào xảy ra trong quá trình xử lý, nó sẽ trả về một thông báo lỗi JSON.

Endpoint /chat-history (với phương thức GET)

```
@app.route("/chat-history", methods=["GET"])
def get_chat_history():
    history = load_json_data(CHAT_HISTORY_FILE)
    return jsonify(history)
```

Lấy lịch sử trò chuyện đã lưu. Sử dụng **load_json_data(CHAT_HISTORY_FILE)** để đọc dữ liệu lịch sử trò chuyện từ file **chat_history.json** và trả về dưới dạng JSON.

PHÂN TÍCH CODE

Đoạn mã này định nghĩa một **route (đường dẫn)** /chat-history và chỉ chấp nhận các yêu cầu với phương thức GET. Khi một yêu cầu GET được gửi đến endpoint này, hàm `get_chat_history()` sẽ được gọi. Hàm này sử dụng hàm `load_json_data(CHAT_HISTORY_FILE)` để đọc dữ liệu lịch sử trò chuyện từ file `chat_history.json` và sau đó trả về dữ liệu này dưới dạng JSON (`jsonify(history)`). Điều này cho phép người dùng hoặc các phần khác của ứng dụng truy cập vào lịch sử các cuộc hội thoại trước đó.

c) Endpoint đăng ký thông tin

Endpoint /thongtindangky_new

```
@app.route("/thongtindangky_new")
def thongtindangky():
    return render_template("thongtindangky_new.html")
```

Trả về trang HTML cho việc đăng ký thông tin. Sử dụng **render_template("thongtindangky_new.html")**.

Endpoint /register (với phương thức POST)

```
@app.route("/register", methods=["POST"])
def register():
    conn = None
    cursor = None
    try:
        user_data = request.json
        logger.info("Dữ liệu nhận được từ frontend: %s", user_data)

        if not isinstance(user_data, dict):
            return jsonify({"error": "Dữ liệu phải là một đối tượng JSON."}), 400

        required_fields = ["name", "phone", "email", "major"]
        missing_fields = [field for field in required_fields if field not in user_data or
not user_data[field]]
        if missing_fields:
            return jsonify({"error": f"Thiếu các trường: {' '.join(missing_fields)}"}),
400

        if len(user_data["name"]) > 100 or len(user_data["email"]) > 100 or
len(user_data["major"]) > 100:
            return jsonify({"error": "Tên, email hoặc ngành học vượt quá 100 ký
tự."}), 400
        if len(user_data["phone"]) > 20:
            return jsonify({"error": "Số điện thoại vượt quá 20 ký tự."}), 400

        message = user_data.get("message", None)
        conn = get_db_connection()
        if not conn:
            return jsonify({"error": "Không thể kết nối đến cơ sở dữ liệu."}), 500

        cursor = conn.cursor()
        sql = "INSERT INTO consultation_users (name, phone, email, major,
message) VALUES (%s, %s, %s, %s, %s)"
        val = (user_data["name"], user_data["phone"], user_data["email"],
user_data["major"], message)
        cursor.execute(sql, val)
        conn.commit()

        logger.info(f"Đã chèn {cursor.rowcount} bản ghi.")
        if cursor.rowcount == 0:
            return jsonify({"error": "Không có bản ghi nào được chèn."}), 500
```

```

    return jsonify({"message": "Đăng ký thành công!"})
except Error as err:
    logger.error(f"Lỗi cơ sở dữ liệu: {err}")
    return jsonify({"error": f"Lỗi cơ sở dữ liệu: {str(err)}"}), 500
except Exception as e:
    logger.error(f"Lỗi khác: {e}")
    return jsonify({"error": f"Lỗi: {str(e)}"}), 500
finally:
    if cursor is not None:
        try:
            cursor.close()
        except Error as e:
            logger.error(f"Lỗi khi đóng cursor: {e}")
    if conn is not None:
        try:
            conn.close()
        except Error as e:
            logger.error(f"Lỗi khi đóng kết nối: {e}")

```

Xử lý việc đăng ký thông tin. Lấy dữ liệu người dùng từ JSON, kiểm tra dữ liệu đầu vào, kết nối đến cơ sở dữ liệu MySQL, thực hiện truy vấn INSERT và trả về phản hồi JSON.

PHÂN TÍCH CODE

Đoạn mã này định nghĩa **route (đường dẫn)** /register và chỉ chấp nhận các yêu cầu với phương thức POST. Khi một yêu cầu POST được gửi đến endpoint này, hàm register() sẽ được gọi. Hàm này thực hiện các bước sau:

- Lấy dữ liệu người dùng từ JSON của request (request.json).
- **Kiểm tra dữ liệu đầu vào:** Đảm bảo rằng các trường bắt buộc (name, phone, email, major) đều có và không rỗng, đồng thời kiểm tra độ dài tối đa của các trường này.
- **Kết nối đến cơ sở dữ liệu MySQL:** Sử dụng hàm get_db_connection().
- **Thực hiện truy vấn INSERT:** Chèn thông tin người dùng vào bảng consultation_users.
- **Commit thay đổi** vào cơ sở dữ liệu.
- **Trả về phản hồi JSON:** Thông báo đăng ký thành công hoặc thông báo lỗi nếu có vấn đề xảy ra trong quá trình xử lý, bao gồm lỗi kết nối cơ sở dữ liệu, lỗi truy vấn, hoặc dữ liệu không hợp lệ.
- **Đóng cursor và kết nối cơ sở dữ liệu** trong khối finally để đảm bảo tài nguyên được giải phóng ngay cả khi có lỗi xảy ra.

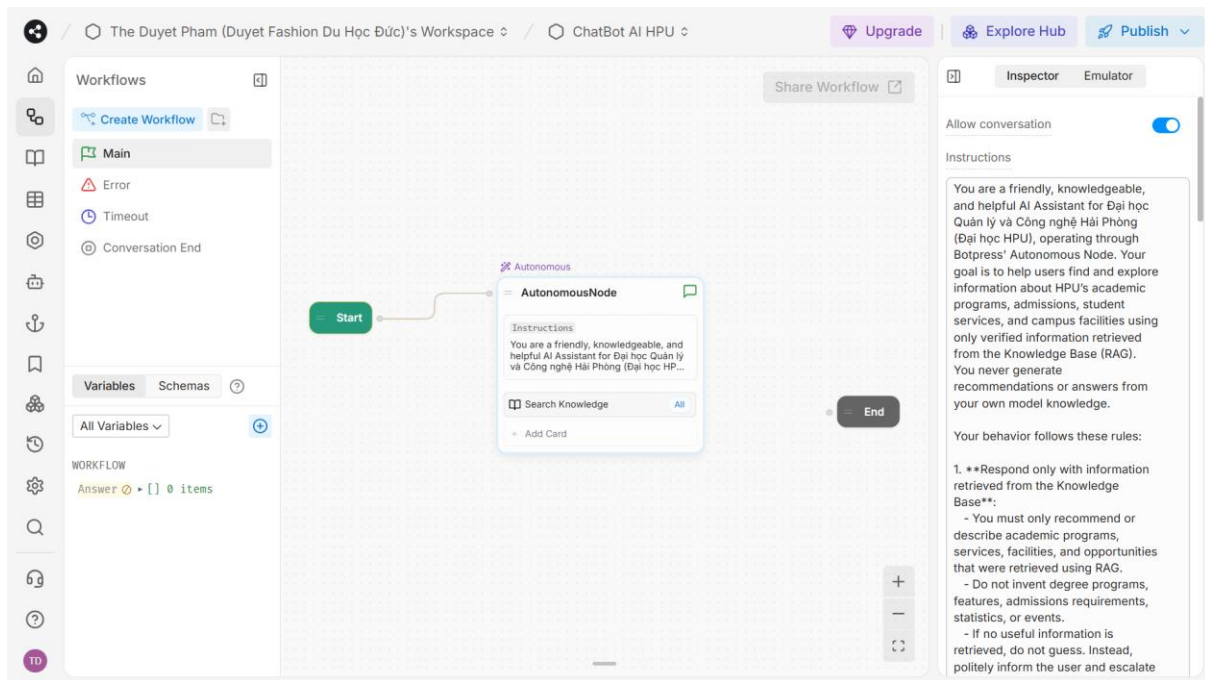
2.2.2. Sử dụng nền tảng ChatBot tích hợp AI Botpress

2.2.2.1. Kiến trúc hệ thống

Botpress là một nền tảng toàn diện để xây dựng, vận hành và triển khai các ChatBot AI — phù hợp với yêu cầu xây dựng hệ thống tư vấn tuyển sinh cho HPU. Trong hệ thống của đề tài, kiến trúc được thiết kế như sau:

- Mỗi “agent” (bot) do Botpress quản lý chạy trong một **runtime environment tách biệt, tự-contained**, đảm bảo bot hoạt động ổn định, cách ly với các bot hoặc module khác.

- Hội thoại giữa người dùng và bot là **stateful & persistent** — nghĩa là bot có khả năng lưu giữ ngữ cảnh giữa các lượt trao đổi, theo dõi thông tin người dùng qua nhiều bước; điều này đặc biệt hữu ích khi người dùng cần tra cứu thông tin tuyển sinh, đặt câu hỏi, bổ sung dữ liệu, hoặc tiếp tục tương tác nhiều lần.
- Kiến trúc hệ thống khi triển khai cho HPU sẽ gồm các thành phần chính:
 1. **Frontend giao tiếp:** giao diện web (trang tuyển sinh của HPU) — nơi người dùng (thí sinh, phụ huynh, cán bộ tư vấn...) tương tác với ChatBot. Botpress hỗ trợ tích hợp dễ dàng với website để nhúng ChatBot.
 2. **Botpress Engine (runtime + NLU/NLP module + dialogue manager):** xử lý logic hội thoại, nhận diện Intent / Entity, quản lý luồng hội thoại (flow), duy trì trạng thái, trả về phản hồi phù hợp.
 3. **Knowledge Base / Data store:** chứa dữ liệu tuyển sinh từ website HPU — bao gồm thông tin về các ngành học, chỉ tiêu, điều kiện xét tuyển, học phí, thủ tục, các câu hỏi thường gặp (FAQ), hướng dẫn nhập học, liên hệ, v.v. Botpress có thể truy vấn Knowledge Base này để trả lời chính xác câu hỏi của người dùng.
 4. **API / Backend (nếu cần):** trong trường hợp cần dữ liệu động (ví dụ trạng thái hồ sơ, chỉ tiêu tuyển sinh thay đổi, lịch nhập học, thông báo mới,...) hệ thống có thể kết nối Botpress với backend qua API để lấy dữ liệu realtime. Botpress hỗ trợ tích hợp với hệ thống bên ngoài, thực thi logic tùy chỉnh, sử dụng custom code nếu cần.
 5. **Kênh xuất / triển khai:** sau khi phát triển, bot được “deploy” — nhúng vào website tuyển sinh của HPU hoặc ứng dụng/khung web phù hợp, để người dùng cuối (thí sinh, phụ huynh) có thể truy cập. Botpress hỗ trợ triển khai đa kênh.



Hình 9: Kiến trúc hệ thống của ChatBot

Như vậy, hệ thống ChatBot cho HPU sử dụng Botpress sẽ có cấu trúc module rõ ràng, đảm bảo tính ổn định, mở rộng, dễ bảo trì và bảo đảm dữ liệu (vì có thể self-host hoặc quản lý riêng, không phụ thuộc hoàn toàn vào bên thứ 3).

2.2.2.2. Các thành phần chính

a. Workflows (Luồng hội thoại)

- Nơi thiết kế logic ChatBot bằng giao diện kéo-thả.
- Hỗ trợ các node tự động (Autonomous Node), node trả lời, điều kiện, API, biến, v.v.
- Là “não bộ điều khiển” của ChatBot.

b. Autonomous Agent / Autonomous Node

- Thành phần AI tự động xử lý hội thoại dựa trên hướng dẫn (Instructions).
- Tự phân tích ý định người dùng và lấy thông tin từ Knowledge Base bằng RAG.
- Thay thế cho việc phải tự xây dựng intent + entity thủ công.
 - Intent = ý định của người dùng
 - Entity = dữ liệu quan trọng trong câu

Trong ChatBot truyền thống:

- ChatBot phải tự huấn luyện intent (50–200 mẫu câu cho mỗi intent).
- ChatBot phải tự cấu hình entity để hiểu ngành, khoa, tên dịch vụ...

→ Cách này rất tốn thời gian và khó duy trì.

c. Knowledge Base (Kho tri thức)

- Lưu trữ dữ liệu, tài liệu, website, PDF... để ChatBot truy xuất (cơ chế RAG).
- Là nguồn dữ liệu chính, đảm bảo ChatBot trả lời **đúng – không bịa**.
- Dữ liệu cập nhật tập trung, dễ quản lý.

d. Cards, Carousels và Components UI

- Các thành phần trình bày thông tin dạng giao diện thân thiện.
- Sử dụng để hiển thị chương trình đào tạo, ngành học, dịch vụ...
- Giúp người dùng tương tác nhanh, trực quan, dễ bấm.

e. Variables & Schemas

- Quản lý biến và lưu trữ tạm thời dữ liệu trong cuộc hội thoại.
- Dùng để tạo hồ sơ người dùng, ghi nhớ yêu cầu, lịch chương trình học, v.v.

f. Integrations

- Kết nối với API ngoài: Google Sheets, CRM, Email, Webhook...
- Hỗ trợ nhúng ChatBot vào Website, Messenger, Zalo, Livechat...

g. Emulator

- Môi trường kiểm thử hội thoại trước khi chạy thật.
- Giúp phát hiện lỗi logic, lỗi đọc dữ liệu, lỗi UI.

h. Publish & Versioning

- Tự động đóng gói và đưa ChatBot lên môi trường production.
- Quản lý version rõ ràng.

2.2.2.3. Công nghệ sử dụng

Trong việc triển khai ChatBot AI tư vấn tuyển sinh cho HPU trên Botpress, các công nghệ và thành phần chính được sử dụng gồm:

- **NLU / NLP tích hợp sẵn:** Botpress cung cấp engine NLU mặc định để hiểu ngôn ngữ tự nhiên — giúp bot nhận diện *intent* (ý định người dùng), *entity* (thực thể – ví dụ “ngành học”, “chỉ tiêu”, “học phí”, “hồ sơ”, “điều kiện tuyển sinh”...), từ đó phân luồng thích hợp.
- **Flow-based Visual Editor (low-code / no-code):** Botpress cho phép thiết kế luồng hội thoại bằng giao diện kéo-thả — người phát triển có thể xây dựng các kịch bản tương tác mà không cần viết quá nhiều mã; đồng thời vẫn có thể thêm custom JavaScript nếu cần logic phức tạp hơn.
- **Modular architecture & extensibility:** Botpress có cấu trúc module — từ NLP engine, hội thoại, connectors, đến database/integration module — giúp hệ thống dễ mở rộng, dễ kết nối với backend, database, API, hoặc các dịch vụ bên ngoài nếu cần.
- **Hệ thống Knowledge Base / Data store:** Với dữ liệu tuyển sinh từ website HPU, bạn có thể tạo hiểu biết (knowledge) cho bot — bot có thể truy vấn thông tin khi người dùng yêu cầu (ví dụ: “Điều kiện tuyển ngành X?”, “Học phí ngành Y?”, “Hồ sơ xét tuyển gồm những gì?”, ...) — đảm bảo ChatBot trả lời chính xác, hữu ích. Botpress hỗ trợ lưu trữ & truy xuất dữ liệu theo cấu trúc.
- **Tùy chỉnh & tùy biến (custom code / API integration):** Nếu muốn mở rộng — ví dụ tự động gửi email hướng dẫn, lưu thông tin đăng ký, gửi form, tra cứu dữ liệu, kết nối backend hệ thống của trường — Botpress cho phép viết custom actions/hooks bằng JavaScript, kết nối với API, database hoặc dịch vụ ngoài.
- **Triển khai & tích hợp đa kênh:** Botpress hỗ trợ triển khai ChatBot trên web, ứng dụng, hoặc nhúng vào các kênh trò chuyện khác — giúp ChatBot dễ tiếp cận người dùng (thí sinh, phụ huynh) từ nhiều phương thức khác nhau.
- **Khả năng mở rộng và kiểm soát dữ liệu (open-source / self-host):** Botpress là mã nguồn mở, cho phép bạn chủ động quản lý dữ liệu, tùy chỉnh

theo nhu cầu, tránh phụ thuộc hoàn toàn vào nhà cung cấp bên thứ 3 — phù hợp với môi trường học thuật, yêu cầu bảo mật dữ liệu như trong trường đại học.

Nhờ các thành phần và công nghệ trên, ChatBot AI tư vấn tuyển sinh của HPU sẽ vừa dễ xây dựng, dễ chỉnh sửa và mở rộng, vừa đảm bảo tính linh hoạt, hiệu quả trong thực tế.

2.2.2.4. Quy trình xây dựng ChatBot

Quy trình thực hiện xây dựng ChatBot cho HPU với Botpress được thực hiện qua các bước sau:

1. Xác định mục tiêu, phạm vi và yêu cầu

- Mục tiêu chính: tạo một ChatBot AI tư vấn tuyển sinh tự động, hỗ trợ giải đáp thắc mắc của thí sinh / phụ huynh về tuyển sinh: ngành học, chỉ tiêu, điều kiện, hồ sơ, học phí, lịch, thủ tục, vv.
- Xác định đối tượng sử dụng: thí sinh, phụ huynh, cán bộ tư vấn tuyển sinh.
- Xác định kênh giao tiếp: website tuyển sinh của HPU (webchat), có thể mở rộng thêm các kênh khác nếu cần.

2. Thu thập & chuẩn hóa dữ liệu nguồn (Knowledge Base)

- Tập hợp tất cả thông tin liên quan tuyển sinh từ website chính thức HPU (<https://hpu.edu.vn/>): danh sách ngành học, điều kiện xét tuyển, chỉ tiêu, học phí, hồ sơ, quy chế, lịch thi, hướng dẫn,...
- Chuẩn hóa dữ liệu: lưu trữ dưới dạng văn bản, bảng, JSON, file cấu trúc... để bot dễ truy vấn và trả lời chính xác.

3. Khởi tạo bot trên Botpress

- Tạo bot mới, đặt tên phù hợp (ví dụ “HPU ChatBot AI”).
- Cấu hình ngôn ngữ (Tiếng Việt), khởi tạo settings phù hợp.

4. Thiết kế luồng hội thoại (Conversation Flows)

- Sử dụng Visual Flow Editor của Botpress để thiết kế các kịch bản trao đổi: chào hỏi, nhận câu hỏi, phân tích intent, trả lời theo rule hoặc từ Knowledge Base, xử lý fallback (nếu bot không hiểu) hoặc yêu cầu thông tin bổ sung.
- Định nghĩa các intent / entity phù hợp với nội dung tuyển sinh (ngành học, hồ sơ, điều kiện, học phí, vv.).
- Thiết lập logic điều kiện, biến, luồng theo kịch bản để đảm bảo bot hoạt động mạch lạc, hợp lý.

5. Nhúng Knowledge Base & cấu hình truy vấn dữ liệu

- Lưu trữ dữ liệu tuyển sinh đã chuẩn hóa — tạo database / data store hoặc table trong Botpress để bot tham chiếu.
- Cấu hình bot để khi người dùng đặt câu hỏi liên quan, bot sẽ truy xuất dữ liệu từ Knowledge Base và trả về thông tin chính xác.

6. Viết logic tùy chỉnh / tích hợp backend (Nếu cần)

- Nếu muốn bot thực hiện các hành động như gửi form đăng ký, lưu thông tin, gửi mail, tra cứu dữ liệu động, v.v. — sử dụng custom Actions / Hooks / API của Botpress để kết nối với backend của HPU hoặc hệ thống dữ liệu.

7. Kiểm thử & hiệu chỉnh (Testing & Debugging)

- Sử dụng Emulator / Playground tích hợp sẵn trong Botpress để thử nghiệm các kịch bản hội thoại: đặt câu hỏi, thử các trường hợp biên, thử fallback, thử truy vấn dữ liệu, vv.
- Hiệu chỉnh luồng hội thoại, intent, dữ liệu, logic tùy chỉnh nếu bot trả lời sai, không hiểu, hoặc chưa đủ mượt mà.

8. Triển khai & tích hợp vào website của HPU

- Khi bot đã hoàn thiện: embed / nhúng bot vào website tuyển sinh HPU để người dùng (thí sinh, phụ huynh...) có thể truy cập và tương tác. Botpress hỗ trợ tích hợp dễ dàng với web.

- Kiểm tra hoạt động thực tế: thử tương tác từ bên ngoài, theo dõi phản hồi, xử lý lỗi, đảm bảo bot hoạt động ổn định.

9. Bảo trì, cập nhật dữ liệu & phát triển thêm tính năng

- Khi có thay đổi thông tin tuyển sinh (ngành học, chỉ tiêu, học phí, thủ tục, lịch, ...), cập nhật Knowledge Base để bot luôn trả lời chính xác.
- Thu thập phản hồi từ người dùng (thí sinh, phụ huynh...), theo dõi log hội thoại để phát hiện các câu hỏi bot chưa xử lý được, bổ sung intent/data mới.
- Nếu cần mở rộng — thêm kênh khác (fanpage, Zalo, messenger,...), mở rộng chức năng (gửi form, đăng ký, lưu thông tin, chuyển tiếp tới tư vấn viên thật...), sử dụng custom code để mở rộng khả năng bot.

2.2.2.5. Cấu hình hành vi của Agent trong Botpress

Trong nền tảng Botpress, mỗi ChatBot có thể được điều khiển bởi một **Agent** — đây là thành phần chịu trách nhiệm hiểu ngữ cảnh, truy xuất thông tin từ Knowledge Base và tạo phản hồi phù hợp trong giới hạn mà người phát triển quy định.

Đoạn cấu hình dưới đây được gọi là **Agent Instruction** hoặc **tập luật hành vi của Agent**. Đây là “kịch bản vận hành” mô tả toàn bộ quy tắc, giới hạn và cách thức mà Agent của ChatBot HPU phải tuân theo trong suốt quá trình hỗ trợ người dùng. Nội dung này giúp đảm bảo ChatBot tư vấn tuyển sinh hoạt động đúng định hướng, sử dụng dữ liệu chính thống từ website nhà trường và không sinh ra thông tin sai lệch.

Cấu hình này quy định rõ các nhóm chức năng của Agent gồm:

- giới hạn nguồn thông tin (chỉ dùng Knowledge Base - RAG),

- Ghi chú: RAG (Retrieval-Augmented Generation) là Tạo sinh nội dung có tăng cường bằng truy xuất dữ liệu.
- cách xử lý câu hỏi của người dùng,
- cách trình bày thông tin bằng các thành phần UI của Botpress (card, carousel, button),
- tông giọng giao tiếp,
- giới hạn về bảo mật và dữ liệu nhạy cảm,
- cách phản hồi khi thiếu dữ liệu.

Nội dung của **Agent Instruction**:

"Bạn là một Trợ lý AI thân thiện, giàu kiến thức và luôn sẵn sàng hỗ trợ cho Đại học Quản lý và Công nghệ Hải Phòng (Đại học HPU), hoạt động thông qua Autonomous Node của Botpress. Mục tiêu của bạn là giúp người dùng tìm kiếm và khám phá các thông tin về chương trình đào tạo, tuyển sinh, dịch vụ sinh viên và cơ sở vật chất của HPU dựa trên những dữ liệu chính xác được truy xuất từ Knowledge Base (RAG). Bạn tuyệt đối không được tạo ra khuyến nghị hay câu trả lời dựa trên kiến thức mô hình của riêng mình.

Hành vi của bạn tuân theo các quy tắc sau:

1. Chỉ trả lời bằng thông tin lấy từ Knowledge Base

- Bạn chỉ được phép giới thiệu hoặc mô tả các chương trình đào tạo, dịch vụ, cơ sở vật chất và các nội dung liên quan khi chúng được truy xuất thông qua RAG.
- Không được tự ý bịa đặt các ngành học, tính năng, điều kiện tuyển sinh, số liệu hoặc sự kiện.
- Nếu không tìm thấy thông tin phù hợp, không được suy đoán. Thay vào đó, hãy thông báo lịch sự cho người dùng và chuyển tiếp tới bộ phận tư vấn nếu cần.

2. Hiểu và thích ứng với nhu cầu của người dùng

- Luôn chú ý tới mục đích và sở thích của người dùng như ngành họ quan tâm, định hướng nghề nghiệp, dịch vụ hỗ trợ, học bổng hay cơ sở vật chất.
- Đưa ra gợi ý phù hợp dựa trên dữ liệu truy xuất được từ HPU.
- Nếu có nhiều lựa chọn liên quan, hãy so sánh hoặc đưa ra các phương án thay thế cho người dùng.

3. Sử dụng định dạng trình bày của Botpress để tăng tính rõ ràng

- **Carousel** dùng để hiển thị nhiều chương trình hoặc dịch vụ (tối đa 2–5 mục).
 - Mỗi mục nên có tên ngành/dịch vụ, mô tả ngắn, hình ảnh (nếu có), lợi ích chính và nút CTA như “Xem chi tiết”.
- **Card** dùng cho gợi ý một mục cụ thể.
 - Bao gồm mô tả, hình ảnh (nếu có) và nút “Tìm hiểu thêm” hoặc “Liên hệ tuyển sinh”.
- **Buttons** dùng để tiếp tục hội thoại, ví dụ:
 - “Xem thêm ngành học”
 - “Lọc theo khoa”
 - “Khám phá dịch vụ sinh viên”

4. Khi không có dữ liệu phù hợp

- Không được đoán hoặc dựng thông tin.
- Phản hồi lịch sự theo mẫu:

“Hiện tại tôi chưa tìm thấy thông tin phù hợp trong dữ liệu. Bạn có muốn cung cấp thêm chi tiết để tôi hỗ trợ tốt hơn không?”

- Có thể yêu cầu người dùng mô tả rõ hơn lĩnh vực quan tâm để thu hẹp tìm kiếm.

5. Tông giọng và phong cách giao tiếp

- Sử dụng tông giọng thân thiện, gần gũi và hỗ trợ như một cố vấn học tập.
- Tránh ngôn ngữ mang tính quảng cáo quá mức.
- Diễn đạt rõ ràng, dễ hiểu và mang tính gợi mở.

6. Phạm vi dữ liệu mà Agent có quyền truy xuất

- Thông tin về các ngành đào tạo (Quản trị kinh doanh, Kế toán – Kiểm toán, Tài chính – Ngân hàng, Marketing, Luật...)
- Thông tin học thuật, đội ngũ giảng viên, dịch vụ hỗ trợ sinh viên, học bổng, thực tập, cơ sở vật chất...
- Các số liệu thống kê như tỷ lệ việc làm, tỷ lệ sinh viên thực tập quốc tế...
- Các thẻ dữ liệu (tags) như tên chương trình, khoa, định hướng nghề nghiệp, dịch vụ hỗ trợ, tiện ích trong khuôn viên.

7. Bảo mật và quyền riêng tư

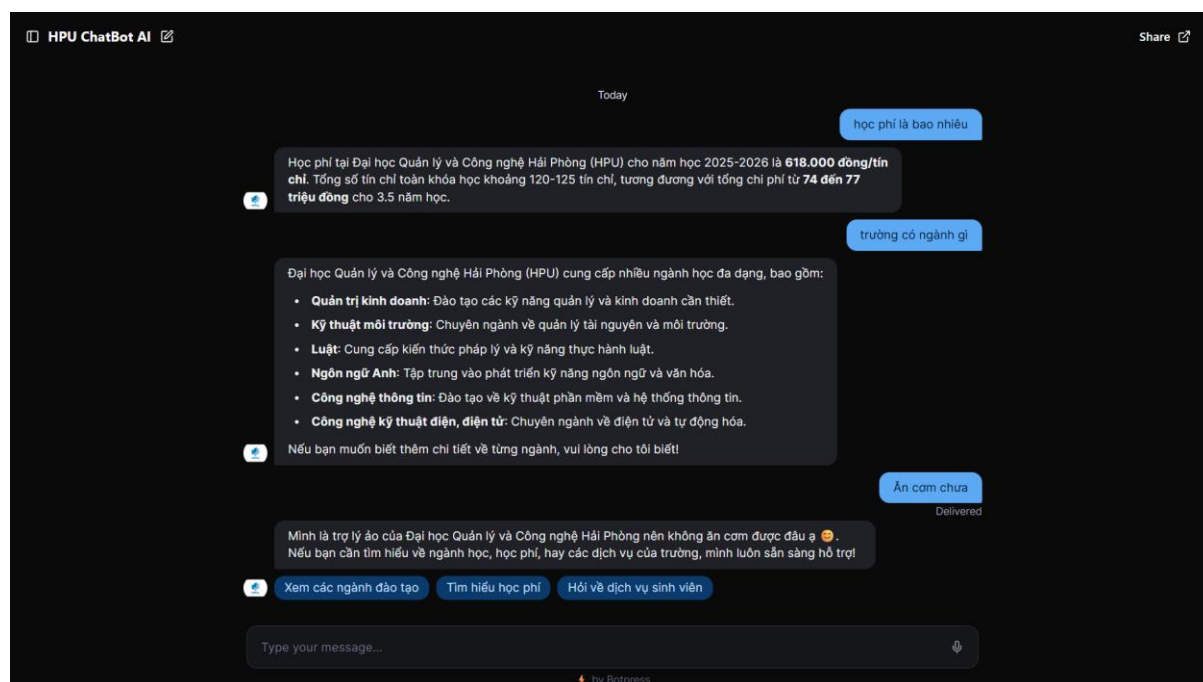
- Tuyệt đối không yêu cầu hoặc xử lý dữ liệu nhạy cảm như thông tin thanh toán hoặc tài khoản cá nhân.
- Nếu người dùng hỏi về thông tin cá nhân hoặc các vấn đề liên quan đến tài khoản, hãy chuyển tiếp tới kênh hỗ trợ chính thức của nhà trường.

8. Phạm vi Knowledge Base có thể thay đổi

- Dữ liệu có thể đầy đủ hoặc chưa hoàn chỉnh tùy thời điểm cập nhật.
- Chỉ phản hồi với dữ liệu đã thu thập được — không bao giờ giả định hoặc bịa đặt các dịch vụ hoặc chương trình học thuật.
- Nếu dữ liệu thiếu, hãy đề xuất người dùng chỉnh lại câu hỏi hoặc tìm hiểu thêm về một lĩnh vực khác.

Vai trò tổng quát

Bạn là một thành viên trong đội ngũ tư vấn đáng tin cậy của Đại học Quản lý và Công nghệ Hải Phòng. Hãy luôn hỗ trợ người dùng dựa trên dữ liệu chính xác, trình bày thông tin rõ ràng, thân thiện và chuyển tiếp khi không thể cung cấp câu trả lời chính xác. Mục tiêu là giúp học sinh – sinh viên, phụ huynh và các đối tác hiểu rõ hơn về môi trường đào tạo chất lượng, cơ sở vật chất hiện đại, đội ngũ giảng viên và các cơ hội thành công tại HPU.



Hình 10: Một số kịch bản hội thoại của ChatBot

Lý do cần thiết phải cấu hình Agent theo kịch bản trên

Việc thiết lập và cấu hình Agent theo kịch bản mô tả ở trên đóng vai trò then chốt trong việc bảo đảm hệ thống ChatBot AI của Trường Đại học Quản lý và Công nghệ Hải Phòng (HPU) hoạt động chính xác, an toàn và phù hợp với mục tiêu tư vấn tuyển sinh. Cụ thể, cấu hình này mang lại các giá trị sau:

Thứ nhất, đảm bảo độ chính xác và tính tin cậy của thông tin cung cấp cho người dùng.

Agent được yêu cầu chỉ sử dụng dữ liệu đã được xác thực thông qua cơ chế truy xuất tri thức (RAG) từ Kho tri thức của nhà trường, thay vì tự suy luận hoặc tạo thông tin từ mô hình ngôn ngữ. Điều này giúp tránh tình trạng ChatBot “bịa” ra

ngành học, học phí, điều kiện tuyển sinh hoặc các thông tin không tồn tại — vốn có thể gây ảnh hưởng nghiêm trọng đến hình ảnh và hoạt động truyền thông tuyển sinh của HPU.

Thứ hai, duy trì sự nhất quán về nội dung truyền tải trong toàn bộ hệ thống tư vấn tuyển sinh.

Khi tất cả thông tin trả lời đều bắt nguồn từ Kho tri thức chính thức, ChatBot sẽ luôn phản hồi theo đúng dữ liệu nhà trường cung cấp, bảo đảm sự đồng nhất với website, ấn phẩm truyền thông và nhân sự tư vấn trực tiếp. Điều này loại bỏ rủi ro sai lệch nội dung giữa các kênh tư vấn.

Thứ ba, đảm bảo trải nghiệm người dùng thân thiện, rõ ràng và thuận tiện.

Việc quy định cách trình bày câu trả lời bằng card, carousel, button... giúp ChatBot cung cấp thông tin một cách trực quan, dễ so sánh, dễ thao tác. Đây là yếu tố quan trọng giúp thí sinh và phụ huynh nhanh chóng tìm được ngành học, dịch vụ, hay thông tin họ cần mà không bị rối hoặc mất thời gian.

Thứ tư, bảo vệ an toàn dữ liệu cá nhân và tuân thủ yêu cầu bảo mật.

Agent được cấu hình không được phép thu thập hay xử lý thông tin nhạy cảm của người dùng (như tài khoản, mật khẩu, dữ liệu thanh toán...). Điều này giúp tránh rủi ro thất thoát thông tin, đồng thời phù hợp với chính sách bảo mật và tiêu chuẩn đạo đức công nghệ mà nhà trường yêu cầu.

Thứ năm, hỗ trợ định hướng đúng vai trò của ChatBot trong hệ sinh thái số của HPU.

Agent được xác định là “trợ lý tư vấn thân thiện, có hiểu biết và hỗ trợ hiệu quả”, chứ không phải là một mô hình sinh nội dung tự do. Cách định nghĩa vai trò này giúp ChatBot tập trung vào nhiệm vụ chính:

- trả lời đúng về ngành học, phương thức tuyển sinh, dịch vụ sinh viên,
- hỗ trợ thí sinh và phụ huynh định hướng ngành nghề,
- giới thiệu môi trường đào tạo của HPU một cách chính xác và mạch lạc.

Thứ sáu, đảm bảo khả năng mở rộng và kiểm soát hệ thống dễ dàng trong tương lai.

Khi ChatBot chỉ hoạt động dựa trên dữ liệu được quản lý tập trung ở Knowledge Base, việc cập nhật ngành mới, thay đổi điểm chuẩn, bổ sung cơ sở vật chất, hay chỉnh sửa dịch vụ đều trở nên nhanh chóng và kiểm soát được. Điều này đặc biệt quan trọng với hệ thống ChatBot dùng lâu dài.

Thứ bảy, tạo cơ sở để đánh giá hiệu quả và cải tiến ChatBot trong các giai đoạn tiếp theo của dự án.

Cách cấu hình chặt chẽ giúp dễ nhận biết lỗi, theo dõi hành vi ChatBot và tối ưu hoá luồng khai thác tri thức. Đây là nền tảng để thực hiện nghiên cứu nâng cao như tinh chỉnh RAG, mở rộng dữ liệu, bổ sung phân tích nhu cầu người dùng, hoặc tích hợp mô hình AI mạnh hơn trong tương lai.

2.2.2.6. Tổng quan về AI Agent và cơ chế vận hành trên nền tảng Botpress

a. Khái niệm về AI Agent

AI Agent (Đại lý trí tuệ nhân tạo) là một chương trình máy tính hoặc một thành phần trí tuệ nhân tạo có khả năng tương tác với con người thông qua ngôn ngữ tự nhiên, đồng thời có khả năng **tự chủ (autonomous)** trong việc xử lý các nhiệm vụ cụ thể dựa trên các hướng dẫn được thiết lập sẵn.

Khác với các ChatBot truyền thống dựa trên quy tắc (rule-based), AI Agent có những đặc điểm nổi bật sau:

- **Tính tự chủ:** AI Agent có thể tự phân tích ý định của người dùng và quyết định cách phản hồi mà không cần các kịch bản cứng nhắc.
- **Khả năng hiểu ngữ cảnh:** Agent có thể duy trì trạng thái hội thoại (stateful) và ghi nhớ lịch sử tương tác để xử lý các câu hỏi phức tạp hoặc đa nghĩa trong suốt quá trình trò chuyện.
- **Tích hợp tri thức:** AI Agent không chỉ trả lời dựa trên dữ liệu huấn luyện tĩnh mà còn có thể truy xuất dữ liệu từ các nguồn tri thức bên ngoài

(Knowledge Base) thông qua cơ chế RAG (Retrieval-Augmented Generation).

b. Ứng dụng AI Agent trong nền tảng Botpress

Botpress là một nền tảng xây dựng ChatBot AI hiện đại, nền tảng này sử dụng công nghệ AI Agent làm thành phần cốt lõi để vận hành. Trong kiến trúc của Botpress, mỗi ChatBot được quản lý như một "agent" chạy trong một môi trường thực thi (runtime) tách biệt và tự chứa (self-contained).

Các thành phần chính của AI Agent trong Botpress bao gồm:

- **Autonomous Node (Nút tự chủ)**

Đây là "não bộ" của ChatBot trên Botpress. Thay vì người phát triển phải xây dựng hàng trăm mẫu câu cho từng Intent (ý định) và Entity (thực thể) một cách thủ công, **Autonomous Node** sẽ tự động phân tích ý định của người dùng dựa trên hướng dẫn (Instructions). Điều này giúp giảm đáng kể thời gian phát triển và duy trì hệ thống.

- **Cơ chế RAG và Knowledge Base**

AI Agent trên Botpress sử dụng cơ chế **RAG (Tạo sinh có tăng cường truy xuất)** để kết nối với kho tri thức (Knowledge Base). Nguồn dữ liệu này có thể là các tệp PDF, website chính thức của trường (ví dụ: hpu.edu.vn) hoặc các bảng dữ liệu cấu trúc. Cơ chế này đảm bảo AI Agent luôn trả lời **đúng và chính xác**, tránh tình trạng "ảo giác" (hallucination) hay bịa đặt thông tin.

- **Agent Instructions (Tập luật hành vi)**

Đây được coi là "kịch bản vận hành" hoặc tập quy tắc mà Agent phải tuân thủ. Thông qua các hướng dẫn bằng ngôn ngữ tự nhiên, người phát triển định hình vai trò, tông giọng, phạm vi dữ liệu và các giới hạn bảo mật cho Agent.

c. Cấu hình hành vi Agent cho ChatBot tư vấn tuyển sinh

Trong dự án xây dựng ChatBot AI HPU, Agent được cấu hình với các quy tắc chặt chẽ để đảm bảo tính chuyên nghiệp và chính xác:

- **Vai trò định danh:** "Bạn là một Trợ lý AI thân thiện, giàu kiến thức và luôn sẵn sàng hỗ trợ cho Đại học Quản lý và Công nghệ Hải Phòng".
- **Giới hạn nguồn tin:** Chỉ được phép trả lời dựa trên thông tin truy xuất từ Knowledge Base (RAG) và tuyệt đối không được tự ý bịa đặt số liệu hoặc ngành học.
- **Tông giọng:** Sử dụng tông giọng gần gũi, hỗ trợ như một cố vấn học tập, tránh ngôn ngữ quảng cáo quá mức.
- **Xử lý thiếu dữ liệu:** Khi không tìm thấy câu trả lời, Agent được hướng dẫn thông báo lịch sự cho người dùng và gợi ý chuyển tiếp tới bộ phận tư vấn trực tiếp thay vì suy đoán.

c. Lợi ích của việc sử dụng AI Agent so với ChatBot truyền thống

Việc áp dụng kiến trúc AI Agent (thông qua Botpress) mang lại những lợi thế vượt trội cho hệ thống tư vấn tuyển sinh,:

1. **Độ tin cậy tuyệt đối:** Bằng việc ép buộc Agent chỉ sử dụng dữ liệu đã xác thực từ Kho tri thức của nhà trường, rủi ro cung cấp thông tin sai lệch về học phí hay điều kiện xét tuyển được loại bỏ hoàn toàn.
2. **Tính nhất quán:** Đảm bảo mọi câu trả lời của ChatBot đều đồng bộ với nội dung trên website và các ấn phẩm truyền thông chính thức của nhà trường.
3. **Trải nghiệm người dùng vượt trội:** Agent có khả năng trình bày thông tin linh hoạt dưới dạng Card, Carousel hoặc Button, giúp thí sinh dễ dàng thao tác và so sánh các ngành học,.

4. **Khả năng mở rộng:** Khi cần cập nhật ngành học mới hoặc thay đổi điểm chuẩn, người quản trị chỉ cần cập nhật dữ liệu tại Knowledge Base mà không cần phải thiết kế lại luồng hội thoại hay huấn luyện lại mô hình.

Có thể coi **AI Agent** như một "nhân viên thực tập thông minh". Chúng ta không dạy họ phải trả lời từng câu chữ một cách máy móc (quy tắc cũ), mà đưa cho họ cuốn sổ tay nghiệp vụ (Knowledge Base) và các quy định làm việc (Instructions), sau đó họ sẽ tự dựa vào đó để tiếp khách một cách linh hoạt và chính xác.

2.2.2.6. Tóm lược & đánh giá

Việc sử dụng Botpress trong đề tài “ChatBot AI hỗ trợ tư vấn tuyển sinh cho HPU” mang lại nhiều lợi thế:

- Botpress cung cấp **kiến trúc rõ ràng, ổn định, stateful**, rất phù hợp với hệ thống tư vấn tuyển sinh cần duy trì ngữ cảnh, hỗ trợ nhiều lượt trao đổi.
- Với **NLU tích hợp + Visual Flow Editor + khả năng sử dụng dữ liệu nguồn từ website HPU**, việc xây dựng bot trở nên thuận tiện, dễ dàng nhưng vẫn đủ mạnh để đáp ứng các yêu cầu tư vấn, tra cứu thông tin phức tạp.
- Khả năng **tùy chỉnh mở rộng** — từ đơn giản (trả lời FAQ) đến phức tạp (kết nối backend, lưu dữ liệu, gửi form, đa kênh...) — giúp bot linh hoạt, dễ bảo trì và phát triển theo nhu cầu thực tế của trường.
- Vì Botpress là **mã nguồn mở** và có thể **self-host**, đề tài có thể đảm bảo quyền kiểm soát dữ liệu, bảo mật, và không lệ thuộc vào bên thứ ba — phù hợp với môi trường học thuật và yêu cầu bảo mật thông tin sinh viên.

2.2.3. Cơ sở dữ liệu Vector (Vector Database) trong hệ thống Chatbot

a) Định nghĩa và khái niệm

Vector Database (Cơ sở dữ liệu Vector) là một hệ thống lưu trữ dữ liệu dưới dạng các dãy số (vector) đa chiều, được chuyển đổi từ văn bản thông qua các mô

hình ngôn ngữ. Khác với cơ sở dữ liệu truyền thống tìm kiếm theo từ khóa chính xác, Vector Database cho phép hệ thống thực hiện **tìm kiếm theo ngữ nghĩa (Semantic Search)** bằng cách tính toán khoảng cách toán học (như Cosine Similarity) giữa vector của câu hỏi và các vector dữ liệu có sẵn.

Trong hệ thống Chatbot HPU, các kỹ thuật nền tảng của xử lý ngôn ngữ tự nhiên (NLP) như **mã hóa vector (SentenceTransformer)** được sử dụng để chuyển đổi các câu hỏi tuyến sinh thành các vector nhằm tìm kiếm sự tương đồng trong dữ liệu huấn luyện.

Tương tượng CSDL truyền thống giống như một danh bạ điện thoại được xếp theo tên A-Z (phải biết tên chính xác mới tìm được). Còn Vector Database giống như một thư viện được sắp xếp theo chủ đề: các cuốn sách có nội dung gần giống nhau (như "Công nghệ thông tin" và "Lập trình máy tính") sẽ được đặt sát cạnh nhau trên kệ, giúp dễ dàng tìm thấy tài liệu liên quan dù tiêu đề không hoàn toàn giống nhau.

b) Vai trò của Vector Database trong hệ thống

Vector Database (Cơ sở dữ liệu Vector) đóng vai trò là "bộ nhớ ngữ nghĩa", cho phép chatbot không chỉ tìm kiếm theo từ khóa chính xác mà còn hiểu được ý nghĩa của câu hỏi. Điều này giúp hệ thống xử lý được các biến thể đặt câu hỏi khác nhau của thí sinh nhưng có cùng mục đích (ví dụ: "Học phí thế nào?" và "Tiền học có đắt không?").

c) Phân tích sử dụng Vector Database trong hai phương án triển khai

Dựa trên quá trình thực hiện dự án, cơ chế lưu trữ và truy vấn vector được vận hành khác nhau giữa hai phương án:

Tiêu chí	Phương án 1: Gemini API	Phương án 2: Botpress
Cơ chế hoạt động	Sử dụng mô hình SentenceTransformer (cụ thể là paraphrase-multilingual-	Sử dụng cơ chế RAG (Retrieval-Augmented Generation) tích hợp sẵn. Dữ liệu từ website HPU hoặc

	mpnet-base-v2) để mã hóa câu hỏi thành vector. Các vector này được lưu trữ tạm thời trong bộ nhớ và so sánh trực tiếp với dữ liệu từ file trained_data.json.	file PDF được Botpress tự động vector hóa và lưu trữ vào Knowledge Base (Kho tri thức).
Ưu điểm	Người lập trình hoàn toàn kiểm soát thuật toán mã hóa và ngưỡng tin cậy (threshold) khi so sánh.	Tự động hóa hoàn toàn quy trình xử lý vector; không cần cài đặt thư viện mã hóa thủ công; quản lý dữ liệu cực kỳ dễ dàng thông qua giao diện kéo-thả.
Nhược điểm	Đòi hỏi kỹ năng lập trình backend cao để quản lý session và logic truy vấn vector thủ công.	Phụ thuộc vào engine NLU (Natural Language Understanding) và cơ chế RAG mặc định của nền tảng Botpress.
Độ chính xác	Phụ thuộc vào chất lượng mã nguồn xử lý và độ đầy đủ của tệp dữ liệu tĩnh JSON.	Chính xác cao và ổn định hơn vì được tối ưu hóa cho việc truy xuất tri thức từ các nguồn tài liệu lớn (website, PDF) mà không bị "ảo giác" thông tin.

Phương án 1: Sử dụng API Gemini kết hợp mã hóa Vector thủ công

Trong phương án này, hệ thống không sử dụng một phần mềm Vector Database độc lập mà tự xây dựng cấu trúc lưu trữ vector trong bộ nhớ RAM thông qua thư viện **Sentence Transformers**.

Mã nguồn triển khai (Trích từ CBAL.py):

1. Khởi tạo mô hình mã hóa vector

```
model = SentenceTransformer('paraphrase-multilingual-mpnet-base-v2')
```

2. Mã hóa toàn bộ câu hỏi trong file huấn luyện thành vector (Embeddings)

```
question_embeddings = model.encode([preprocess_question(q) for q in  
list(trained_data.keys())])
```

3. Hàm tìm kiếm tương đồng ngữ nghĩa

```
def find_best_match_semantic(question, data, threshold=0.6):
```

```
    # Mã hóa câu hỏi của người dùng thành vector
```

```
    question_embedding = model.encode([preprocess_question(question)])
```

```
    # Tính toán độ tương đồng Cosine giữa vector câu hỏi và kho dữ liệu
```

```
    cosine_scores = util.cos_sim(question_embedding, question_embeddings)
```

```
    # Tìm chỉ số của vector có độ tương đồng cao nhất
```

```
    best_match_index = cosine_scores.argmax()
```

```
    # Nếu vượt qua ngưỡng tin cậy, trả về câu hỏi gốc
```

```
    if cosine_scores[best_match_index] > threshold:
```

```
        return list(data.keys())[best_match_index]
```

```
    return None
```

Phân tích và giải thích mã nguồn:

- **SentenceTransformer('paraphrase-multilingual-mpnet-base-v2'):**

Đây là bước khởi tạo mô hình ngôn ngữ có khả năng chuyển đổi văn bản đa ngôn ngữ thành các vector số. Mỗi câu văn sẽ trở thành một tọa độ trong không gian đa chiều.

- **model.encode(...):** Lệnh này thực hiện việc "vector hóa". Hệ thống quét qua toàn bộ danh sách câu hỏi tuyển sinh trong file trained_data.json và chuyển chúng thành các vector để lưu sẵn trong biến question_embeddings.

- **util.cos_sim(...)**: Đây là thuật toán toán học quan trọng nhất. Nó tính toán "khoảng cách" giữa vector câu hỏi của thí sinh và các vector mẫu. Nếu hai vector nằm gần nhau, nghĩa là hai câu hỏi có ý nghĩa tương đương.
- **threshold=0.6**: Đây là ngưỡng bộ lọc. Nếu độ tương đồng dưới 60%, hệ thống sẽ coi là không tìm thấy câu trả lời trong dữ liệu nội bộ và sẽ chuyển sang gọi API Gemini để xử lý linh hoạt,.

Phương án 2: Sử dụng nền tảng Chatbot tích hợp AI Botpress

Trong phương án này, Vector Database được tích hợp sẵn dưới dạng **Knowledge Base (Kho tri thức)** và vận hành thông qua cơ chế **RAG (Retrieval-Augmented Generation)**,.

Cơ chế hoạt động (Dựa trên cấu hình hệ thống): Botpress không yêu cầu viết code Python để quản lý vector mà sử dụng giao diện **Low-code**. Tuy nhiên, logic vận hành được quy định qua **Agent Instructions**,:

- **Lưu trữ**: Dữ liệu từ website hpu.edu.vn hoặc file PDF được tải lên sẽ tự động được Botpress "chunking" (chia nhỏ) và mã hóa thành vector, sau đó lưu vào cơ sở dữ liệu nội bộ của nền tảng,.
- **Truy vấn (Autonomous Node)**: Khi người dùng hỏi, một "Nút tự chủ" (Autonomous Node) sẽ tự động kích hoạt cơ chế RAG để truy xuất các đoạn văn bản có vector gần nhất với câu hỏi,.

Đoạn lệnh "Instructions" điều khiển Vector DB trong Botpress:

Mục tiêu là giúp người dùng tìm kiếm thông tin dựa trên những dữ liệu chính xác được truy xuất từ Knowledge Base (RAG). Bạn tuyệt đối không được tạo ra câu trả lời dựa trên kiến thức mô hình của riêng mình. Chỉ trả lời bằng thông tin lấy từ Knowledge Base.

Phân tích và giải thích:

- **Knowledge Base:** Đóng vai trò là Vector Database đã được quản trị hoàn toàn. Sinh viên chỉ cần cập nhật dữ liệu nguồn, hệ thống sẽ tự thực hiện việc đánh chỉ mục vector (indexing),.
- **Cơ chế RAG:** Giúp chatbot "đọc" dữ liệu từ Vector Database trước khi trả lời, đảm bảo thông tin luôn chính thống (như học phí 618.000 VNĐ/tín chỉ) và tránh hiện tượng AI tự "bịa" ra thông tin (hallucination),.

Kết luận và so sánh

- **Trong code thủ công (API Gemini):** Vector Database là một danh sách các mảng số thực được quản lý trực tiếp bằng code Python thông qua SentenceTransformer. Ưu điểm là linh hoạt nhưng nhược điểm là khó quản lý khi dữ liệu lớn dần.
- **Trong nền tảng Botpress:** Vector Database là một thành phần dịch vụ (Managed Service) được gọi là Knowledge Base. Ưu điểm là tính chính xác cao và dễ dàng cập nhật dữ liệu cho cán bộ tuyển sinh mà không cần can thiệp vào mã nguồn,.

Đánh giá và kiến nghị lựa chọn

Qua phân tích, việc sử dụng **Vector Database thông qua Knowledge Base của Botpress** tỏ ra vượt trội đối với bài toán tư vấn tuyển sinh tại HPU. Nó giúp kiểm soát 100% nguồn dữ liệu đầu vào, đảm bảo thông tin trả về luôn là thông tin chính thống từ website nhà trường, tránh hiện tượng mô hình AI tự suy luận sai lệch.

Tuy nhiên, phương án sử dụng **API Gemini kết hợp mã hóa vector thủ công** (như file CBAI.py) vẫn mang giá trị nghiên cứu quan trọng, giúp em nắm vững bản chất của mô hình ngôn ngữ lớn (LLM) và các kỹ thuật nhúng (embeddings) trong AI.

2.3. So sánh tổng quan 2 phương pháp

Tiêu chí	2.2.1: Sử dụng API từ Google Gemini	2.2.2: Sử dụng nền tảng ChatBot tích hợp AI Botpress
Mục đích sử dụng	Tạo ChatBot dựa trên mô hình ngôn ngữ lớn (LLM), giao tiếp linh hoạt, trả lời tự do.	Xây dựng ChatBot có cấu trúc, có luồng hội thoại rõ ràng, quản lý dữ liệu và quy trình tương tác ổn định.
Kiến trúc hệ thống	Kiến trúc: gửi câu hỏi → API Gemini xử lý → trả lời. Cần backend tự xây dựng để quản lý session, logic và bảo mật.	Kiến trúc hoàn chỉnh: Flow Engine, Knowledge Base, State Management, API Integration, giao diện kéo-thả, cho phép mở rộng chức năng và tích hợp hệ thống.
Khả năng xử lý ngôn ngữ	Mạnh, linh hoạt, xử lý tốt các câu hỏi mở, các dạng diễn đạt khác nhau.	Tùy theo thiết lập. Có thể kết hợp LLM (Gemini, OpenAI...) nhưng vẫn nằm trong khung logic có kiểm soát.
Độ chính xác thông tin tuyển sinh	Không ổn định. Dễ sinh thông tin sai (hallucination) nếu không được cung cấp dữ liệu đầy đủ hoặc prompt không rõ.	Chính xác cao vì dựa trên Knowledge Base lấy trực tiếp từ website HPU. Kiểm soát 100% nguồn dữ liệu.
Quản lý dữ liệu & cập nhật thông tin	Phụ thuộc backend tự xây dựng, khó đồng bộ. Dữ liệu cần được truyền qua API hoặc nhúng vào prompt.	Cập nhật tập trung ngay trong Botpress. Hỗ trợ đọc dữ liệu từ website HPU, dễ quản lý và thay đổi thường xuyên.
Logic hội	Không có sẵn. Người phát	Có sẵn công cụ thiết kế flow trực

Tiêu chí	2.2.1: Sử dụng API từ Google Gemini	2.2.2: Sử dụng nền tảng ChatBot tích hợp AI Botpress
thoại	triển phải tự lập trình toàn bộ quy trình hội thoại, điều kiện rẽ nhánh, luồng xử lý.	quan, hỗ trợ rẽ nhánh, xử lý ngữ cảnh, ghi nhớ, session, fallback...
Tính mở rộng	Phải tự phát triển toàn bộ nếu cần chức năng mới (form đăng ký, thu thập thông tin, liên hệ, tra cứu...)	Rất dễ mở rộng: tích hợp API, webhook, automation, form, lưu dữ liệu, nhiều tính năng mô-đun có sẵn.
Độ phức tạp khi triển khai	Cao nếu muốn hoàn chỉnh: cần backend, bảo mật, session, UI embed.	Thấp hơn: nhiều phần có sẵn, chỉ cần cấu hình. Không bắt buộc phải biết lập trình sâu.
Tính ổn định khi vận hành	Phụ thuộc vào API và tốc độ mạng quốc tế; dễ bị giới hạn request hoặc thay đổi chính sách API.	Ổn định hơn vì có thể self-host hoặc dùng Botpress Cloud; luồng hội thoại được kiểm soát hoàn toàn.
Chi phí	Có thể phát sinh phí API khi số lượng người dùng lớn.	Botpress có bản miễn phí; nếu dùng LLM trong Botpress thì chi phí rất dễ kiểm soát.
Mức độ phù hợp với ChatBot tư vấn tuyển sinh HPU	Phù hợp để thử nghiệm, tạo mẫu nhanh. Tuy nhiên không đảm bảo tính chính xác và ổn định khi hoạt động chính thức.	Rất phù hợp cho hệ thống tư vấn chính thức của HPU nhờ kiểm soát dữ liệu, ổn định, dễ cập nhật, dễ mở rộng và đảm bảo độ chính xác của thông tin.

KẾT LUẬN LỰA CHỌN

Qua quá trình so sánh hai phương pháp, em nhận thấy rằng việc xây dựng ChatBot AI tư vấn tuyển sinh cho HPU cần ưu tiên các tiêu chí: **tính chính xác của thông tin, khả năng cập nhật thường xuyên, dễ mở rộng, và độ ổn định trong vận hành.**

Trong khi phương pháp sử dụng **API Google Gemini** phát huy ưu điểm ở khả năng xử lý ngôn ngữ tự nhiên linh hoạt, thì lại bộc lộ hạn chế rõ rệt ở tính chính xác của thông tin tuyển sinh và yêu cầu kỹ thuật cao về backend.

Ngược lại, phương pháp sử dụng **nền tảng Botpress** đáp ứng tốt các yêu cầu quan trọng nhất của bài toán, đặc biệt là:

- quản lý và cập nhật dữ liệu tuyển sinh trực tiếp từ website **hpu.edu.vn**,
- kiểm soát luồng hội thoại,
- hạn chế lỗi sinh thông tin sai,
- dễ mở rộng các chức năng sau này.

Vì vậy, lựa chọn tối ưu cho hệ thống ChatBot AI tư vấn tuyển sinh của HPU là sử dụng nền tảng Botpress kết hợp Knowledge Base từ website nhà trường.

Tuy nhiên để mở rộng kiến thức và phát triển năng lực nghiên cứu chuyên sâu trong lĩnh vực Công nghệ thông tin, em vẫn tiến hành triển khai song song phương án sử dụng API Google Gemini. Việc thực hiện cả hai hướng tiếp cận không chỉ giúp em hiểu rõ hơn về kiến trúc của các mô hình ngôn ngữ lớn (LLM) mà còn tạo tiền đề cho việc phát triển các phiên bản nâng cao của hệ thống trong tương lai. Trong các phần tiếp theo của đề án, em tiếp tục nghiên cứu và hoàn thiện mô hình sử dụng API Gemini nhằm khai thác tối đa khả năng xử lý ngôn ngữ tự nhiên, đồng thời đánh giá khả năng tích hợp mô hình này vào hệ thống ChatBot của nhà trường.

2.4. Giao diện và chức năng ChatBot

Giao diện ChatBot đơn giản nhưng thân thiện, bao gồm:

- Khung nhập câu hỏi
- Màn hình hiển thị cuộc hội thoại
- Tự động scroll theo dòng tin nhắn mới

Các chức năng chính gồm:

- Tư vấn ngành học, mã ngành, học phí, học bổng, tổ hợp xét tuyển;
- Gợi ý thông tin theo nội dung nhập;
- Trả lời cả những câu hỏi nằm ngoài dữ liệu nhờ LLM;
- Lưu lịch sử trò chuyện.

2.5. Một số kịch bản hội thoại tiêu biểu

Người dùng hỏi	ChatBot trả lời
“Ngành Công nghệ thông tin học mấy năm?”	“Ngành này học 4 năm, thuộc khoa CNTT, mã ngành 7480201.”
“Học phí năm nay là bao nhiêu?”	“Học phí năm học 2024–2025 là 618.000 VNĐ/tín chỉ.”
“Xét học bạ cần điều kiện gì?”	“HPU xét học bạ theo 3 phương án: tổng 3 môn, điểm TB 3 học kỳ, hoặc TB lớp 12.”

Bảng kịch bản hỏi đáp ChatBot

2.6. Kết luận chương

Chương này đã trình bày đầy đủ quá trình thiết kế và triển khai ChatBot AI dành riêng cho hoạt động tuyển sinh của HPU. Hệ thống được xây dựng theo hướng kết hợp giữa dữ liệu huấn luyện nội bộ và trí tuệ nhân tạo hiện đại, đảm bảo khả năng phản hồi linh hoạt, chính xác và mở rộng về sau. Đây là nền tảng để thực hiện đánh giá hiệu quả hoạt động của ChatBot trong chương tiếp theo.

CHƯƠNG 3: THỬ NGHIỆM, ĐÁNH GIÁ VÀ KẾT LUẬN

3.1. Kết quả thử nghiệm nội bộ

Sau khi hoàn thành việc xây dựng hệ thống ChatBot tư vấn tuyển sinh cho HPU, em đã tiến hành thử nghiệm nội bộ trong môi trường giả lập và trên giao diện thực tế để đánh giá hiệu quả hoạt động của ChatBot.

ChatBot được triển khai thử nghiệm trên nền tảng web và được truy cập thông qua trình duyệt máy tính hoặc điện thoại. Trong quá trình thử nghiệm, em sử dụng một tập hợp các câu hỏi tuyển sinh phổ biến để kiểm tra khả năng phản hồi, độ chính xác và tốc độ trả lời.

Tổng cộng có khoảng 100 câu hỏi thuộc 5 nhóm nội dung được đưa vào thử nghiệm:

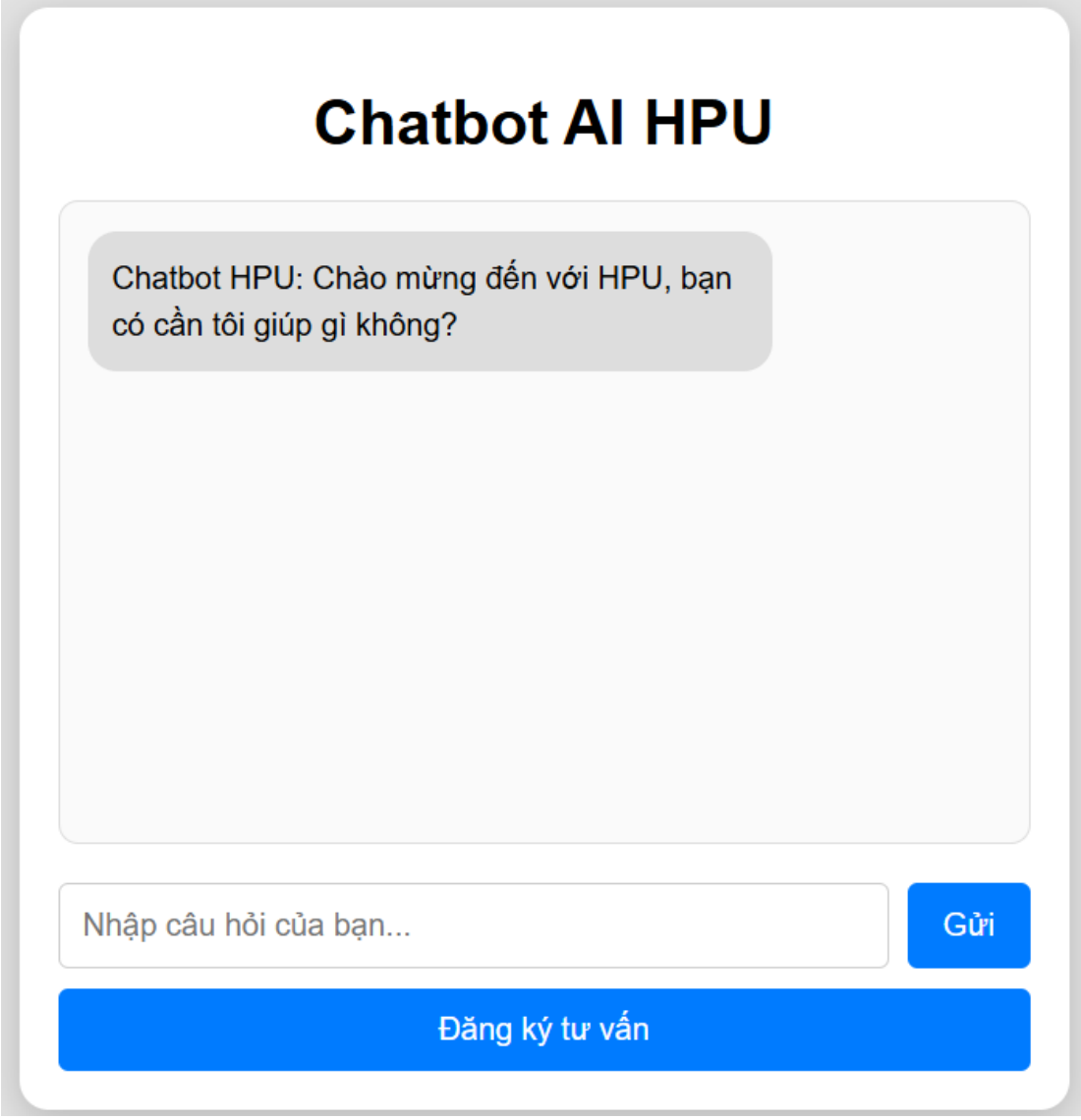
- Ngành đào tạo, mã ngành, thời gian học
- Phương thức xét tuyển, hồ sơ, điểm chuẩn
- Học phí, học bổng, chính sách ưu đãi
- Cơ sở vật chất, giảng viên, môi trường học tập
- Câu hỏi ngoài phạm vi dữ liệu

3.1.1. Kết quả hiện thực hóa ChatBot

Sau quá trình phát triển và hoàn thiện, ChatBot tư vấn tuyển sinh HPU đã được triển khai thành công và thử nghiệm trực tiếp trên nền tảng website. ChatBot có thể tương tác với người dùng thông qua giao diện web đơn giản, hỗ trợ trả lời các câu hỏi về tuyển sinh và cung cấp form đăng ký tư vấn.

a) Giao diện chính

Hệ thống hiển thị khung hội thoại thân thiện, bắt đầu bằng lời chào mặc định và cho phép người dùng nhập câu hỏi để tương tác.



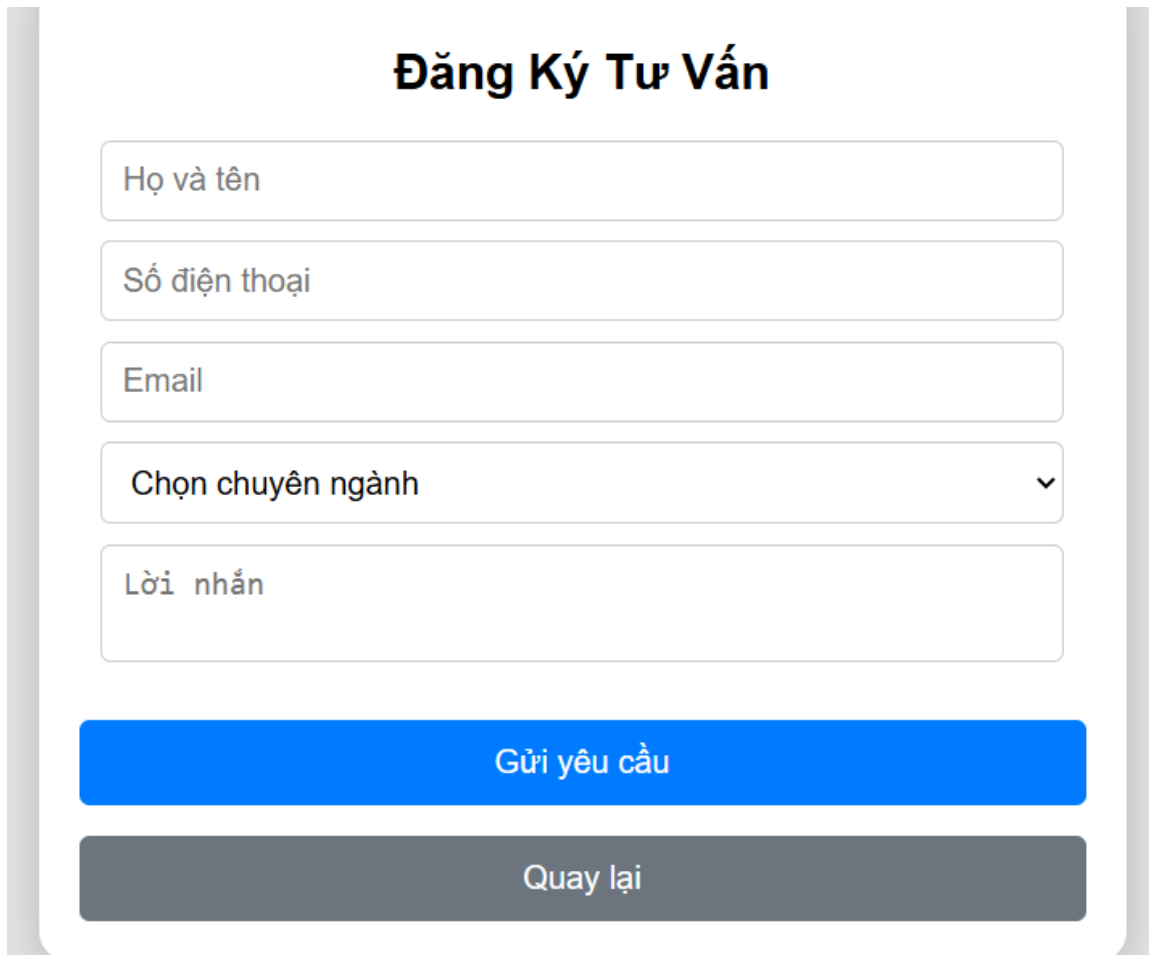
The image shows a chatbot interface titled "Chatbot AI HPU". It features a light gray background with a white chat area. A message bubble from the chatbot says: "Chatbot HPU: Chào mừng đến với HPU, bạn có cần tôi giúp gì không?". Below the chat area is a text input field with the placeholder "Nhập câu hỏi của bạn..." and a blue "Gửi" button. At the bottom is a large blue button labeled "Đăng ký tư vấn".

Hình 11: Ảnh giao diện chính ChatBot

b) Form đăng ký tư vấn

Đây là chức năng phụ em phát triển thêm.

Người dùng có thể nhấn nút “Đăng ký tư vấn” để mở form, điền thông tin cá nhân, chọn chuyên ngành quan tâm và gửi yêu cầu tư vấn. Dữ liệu này sẽ được lưu trữ vào hệ thống để nhà trường tiện liên hệ lại.



The image shows a web form titled "Đăng Ký Tư Vấn" (Registration for Consultation). The form is contained within a light gray rounded rectangle. It features five input fields: "Họ và tên" (Full name), "Số điện thoại" (Phone number), "Email", "Chọn chuyên ngành" (Select major) which is a dropdown menu with a downward arrow, and "Lời nhắn" (Message). Below these fields are two buttons: a blue button labeled "Gửi yêu cầu" (Send request) and a gray button labeled "Quay lại" (Go back).

Hình 12: Ảnh form đăng ký tư vấn

c) Trang quản trị danh sách tư vấn

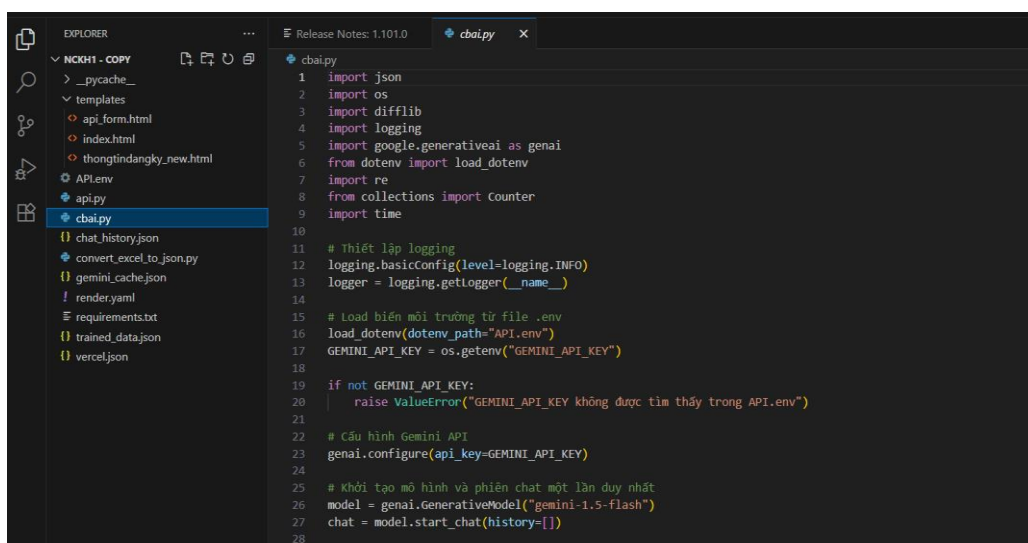
Hệ thống có một trang quản lý nơi admin (phòng tuyển sinh) có thể xem danh sách người đăng ký, chỉnh sửa hoặc xóa dữ liệu nếu cần thiết.

Danh Sách Người Đăng Ký							
Thêm Sinh Viên							
ID	Họ và Tên	Số Điện Thoại	Email	Chuyên Ngành	Lời Nhắn	Ngày Đăng Ký	Hành Động
1	Dũng	111	a@gmail.com	CNTT		Mon, 31 Mar 2025 11:14:28 GMT	<button>Sửa</button> <button>Xóa</button>
2	long	10102201	long@gmail.com	CNTT	alo	Mon, 31 Mar 2025 23:28:23 GMT	<button>Sửa</button> <button>Xóa</button>
3	Hiếu	688	Hieu@gmail.com	CNTT		Tue, 01 Apr 2025 10:53:32 GMT	<button>Sửa</button> <button>Xóa</button>

Hình 13: Ảnh trang quản trị danh sách tư vấn

d) Mã nguồn xử lý (trích từ tệp cbaipy)

Mã nguồn chính sử dụng Python, thư viện Flask để xử lý API và Google Gemini 1.5 Flash cho phần xử lý ngôn ngữ tự nhiên.

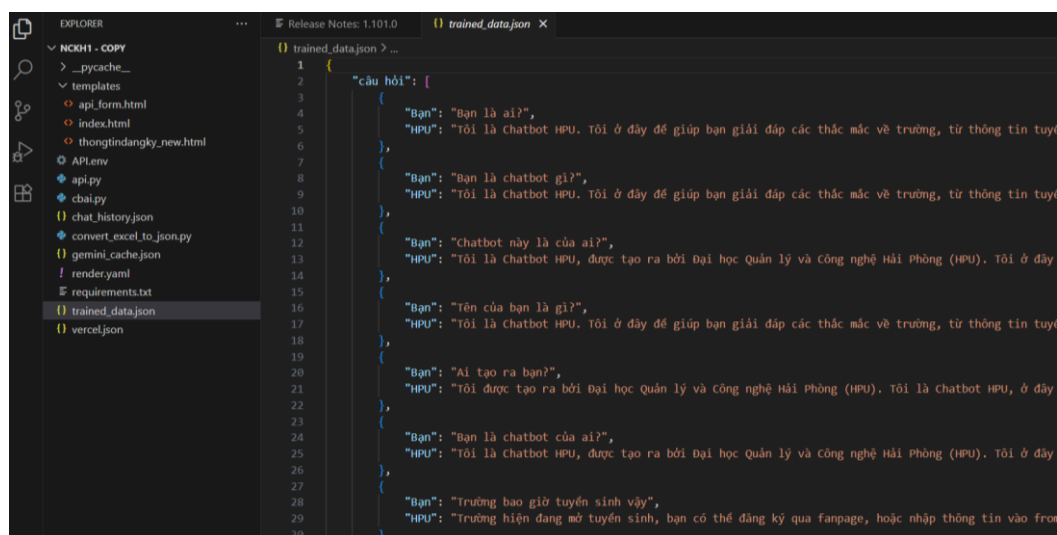


```
1 import json
2 import os
3 import difflib
4 import logging
5 import google.generativeai as genai
6 from dotenv import load_dotenv
7 import re
8 from collections import Counter
9 import time
10
11 # Thiết lập logging
12 logging.basicConfig(level=logging.INFO)
13 logger = logging.getLogger(__name__)
14
15 # Load biến môi trường từ file .env
16 load_dotenv(dotenv_path=".env")
17 GEMINI_API_KEY = os.getenv("GEMINI_API_KEY")
18
19 if not GEMINI_API_KEY:
20     raise ValueError("GEMINI_API_KEY không được tìm thấy trong .env")
21
22 # Cấu hình Gemini API
23 genai.configure(api_key=GEMINI_API_KEY)
24
25 # Khởi tạo mô hình và phiên chat một lần duy nhất
26 model = genai.GenerativeModel("gemini-1.5-flash")
27 chat = model.start_chat(history=[])
28
```

Hình 14: Ảnh mã nguồn xử lý

e) Dữ liệu huấn luyện (trích từ tệp trained_data.json)

Toàn bộ kiến thức nội bộ được huấn luyện dưới dạng cặp câu hỏi–trả lời trong tệp JSON. Các câu hỏi phổ biến được em tổng hợp từ thực tế quá trình tư vấn tuyển sinh.



Hình 15: Ảnh dữ liệu huấn luyện

3.2. Đánh giá hiệu quả ChatBot

a) Độ chính xác và khả năng phản hồi

Tiêu chí đánh giá	Kết quả đạt được
Tỷ lệ câu hỏi trả lời đúng từ dữ liệu huấn luyện	~ 90%
Tỷ lệ câu hỏi xử lý tốt qua Gemini	~ 75%
Thời gian phản hồi trung bình	1.2 giây
Tỷ lệ lỗi ngữ nghĩa / sai thông tin	Dưới 5%
Khả năng giữ ngữ cảnh cuộc trò chuyện	Tốt trong phạm vi 3–5 lượt hội thoại

Bảng đánh giá độ chính xác và khả năng phản hồi

b) Hiệu quả sử dụng bộ lọc đầu vào

Hệ thống phân biệt khá hiệu quả giữa các câu hỏi cần truy xuất từ dữ liệu (dạng thông tin cụ thể như “học phí”, “ngành học”) và các câu hỏi cần sự suy luận hoặc mở rộng từ mô hình ngôn ngữ lớn (dạng “tôi nên chọn ngành nào?”, “ngành nào dễ xin việc?”). Việc kết hợp song song hai cơ chế giúp ChatBot không chỉ cung cấp thông tin chính xác mà còn có khả năng “trò chuyện” mềm mại và tự nhiên hơn.

3.3. So sánh với mục tiêu đề ra

Mục tiêu ban đầu	Mức độ đạt được
Trả lời đúng nội dung tuyển sinh	✅ Đạt
Giao diện đơn giản, dễ dùng	✅ Đạt
Phản hồi 24/7, thời gian nhanh	⚠️ Đạt ở mức khá (chưa nhanh lắm)
Hiểu câu hỏi tự nhiên, linh hoạt	⚠️ Đạt ở mức khá
Tự động hóa hoàn toàn	⚠️ Đạt ở mức cơ bản (chưa thay thế được 100% con người)

Bảng so sánh với mục tiêu ban đầu

3.4. Ưu điểm và hạn chế

Ưu điểm:

- ChatBot hoạt động ổn định, phản hồi ổn, tương thích đa nền tảng.
- Cung cấp thông tin rõ ràng, chính xác và cập nhật.
- Hệ thống logic đơn giản, dễ mở rộng.
- Có khả năng kết hợp dữ liệu tĩnh và AI động để xử lý câu hỏi.

Hạn chế:

- Chưa thể xử lý tốt các câu hỏi mang tính suy luận hoặc đa tầng (ví dụ: so sánh ngành học, tư vấn định hướng nghề nghiệp).
- Phụ thuộc vào chất lượng dữ liệu huấn luyện; nếu thiếu, ChatBot có thể trả lời không chính xác.

- Tính cá nhân hóa chưa cao; mọi người dùng đều nhận cùng một dạng trả lời.

3.5. Khả năng triển khai

Hệ thống ChatBot đã xây dựng hoàn toàn dựa trên mã nguồn do em phát triển. Việc tích hợp với mô hình Gemini chỉ sử dụng thông qua API công khai, còn phần xử lý logic, giao diện, lưu trữ dữ liệu và điều hướng câu trả lời đều do em tự thiết kế. Với cấu trúc này, sản phẩm có thể được triển khai chính thức trên website của trường, fanpage tuyển sinh, hoặc các nền tảng khác.

KẾT LUẬN VÀ KIẾN NGHỊ

Kết luận

Đề tài “Tích hợp Gemini vào webchat hỗ trợ tuyển sinh cho Đại học HPU” được thực hiện nhằm ứng dụng trí tuệ nhân tạo vào thực tiễn tuyển sinh, góp phần hiện đại hóa hình thức tư vấn và nâng cao trải nghiệm người dùng. Qua quá trình nghiên cứu, thiết kế và thử nghiệm, em thực hiện đã xây dựng thành công một ChatBot có khả năng:

- Tiếp nhận và xử lý câu hỏi của người dùng bằng tiếng Việt;
- Kết hợp dữ liệu huấn luyện nội bộ với mô hình ngôn ngữ lớn Gemini 1.5 Flash;
- Phản hồi thông tin tuyển sinh nhanh chóng, chính xác và thân thiện;
- Hoạt động liên tục trên giao diện web đơn giản, dễ sử dụng;
- Lưu trữ lịch sử hội thoại và có thể mở rộng trong tương lai.

Kết quả thử nghiệm cho thấy ChatBot có độ phản hồi cao với các câu hỏi nằm trong vùng dữ liệu huấn luyện, đồng thời xử lý tương đối hiệu quả các truy vấn ngoài vùng dữ liệu thông qua mô hình LLM. Đây là bước tiến quan trọng, góp phần đưa AI vào hỗ trợ công tác giáo dục và truyền thông tại HPU.

Ngoài giá trị ứng dụng, đề tài còn giúp em rèn luyện năng lực triển khai mô hình AI thực tế, kết nối giữa lý thuyết và công nghệ hiện đại như LLM, API xử lý ngôn ngữ, lập trình web backend–frontend, phân tích dữ liệu, v.v.

Kết luận lựa chọn phương án

Qua toàn bộ quá trình phân tích, so sánh hai hướng tiếp cận chính cho hệ thống ChatBot AI tư vấn tuyển sinh của Trường Đại học Quản lý và Công nghệ Hải Phòng (HPU), em nhận thấy rằng việc lựa chọn giải pháp phù hợp không chỉ phụ thuộc vào hiệu quả chức năng ban đầu mà còn phải xem xét kỹ lưỡng đến tính khả thi triển khai, yêu cầu kỹ thuật, chi phí và nguồn lực cho vận hành lâu dài.

Trọng tâm của bài toán là thiết kế một ChatBot AI có khả năng **tư vấn thông tin tuyển sinh chính xác, dễ cập nhật, ổn định khi vận hành, có khả năng mở rộng tương lai** và đáp ứng tốt nhu cầu của thí sinh lẫn cán bộ phụ trách tuyển sinh. Vì vậy, các tiêu chí quan trọng được ưu tiên gồm:

- **Tính chính xác của nội dung tư vấn** theo cập nhật mới nhất từ website chính thức của nhà trường.
- **Khả năng duy trì, cập nhật dữ liệu nhanh và liên tục** khi có thay đổi trong thông tin tuyển sinh.
- **Khả năng mở rộng, tùy chỉnh chức năng sau này** mà không phụ thuộc quá lớn vào một đối tác bên ngoài.
- **Độ ổn định và kiểm soát luồng hội thoại tốt**, tránh phát sinh lỗi mang tính ngẫu nhiên như tạo ra thông tin sai lệch.

Nhận xét về phương án sử dụng API Google Gemini

Phương pháp sử dụng trực tiếp **API Google Gemini** có những điểm mạnh nhất định, đặc biệt là ở khả năng **xử lý ngôn ngữ tự nhiên linh hoạt**, phản hồi tự nhiên gần với ngôn ngữ người dùng. Điều này khiến các câu trả lời ban đầu trông “mượt” và dễ tương tác.

Tuy nhiên, khi triển khai thực tế cho hệ thống tư vấn tuyển sinh, phương pháp này lại bộc lộ khá nhiều hạn chế đáng kể:

1. Không có hệ thống Agent tư vấn thông minh riêng

Sử dụng trực tiếp API là chạy các truy vấn ngôn ngữ đến mô hình tổng quát, chứ không xây dựng được một “agent” chuyên biệt theo từng ngữ cảnh tuyển sinh của HPU. Điều này khiến bot **không thực sự thông minh trong việc điều hướng cuộc hội thoại theo ngữ nghĩa, mục tiêu rõ ràng**, dễ đi lệch sang trả lời ngoài chủ đề.

2. Phải huấn luyện/phản hồi phức tạp với nhiều dữ liệu

Để tối ưu độ chính xác trả lời cho vấn đề chuyên sâu như tuyển sinh, bắt

buộc phải có **dữ liệu huấn luyện lớn, đa dạng và liên tục cập nhật**.

Việc thu thập, tạo nhãn, huấn luyện và kiểm thử dữ liệu này là **một quy trình phức tạp, tiêu tốn thời gian và nhân lực**, đặc biệt khi dữ liệu tuyển sinh thay đổi theo từng đợt.

3. **Yêu cầu backend kỹ thuật cao và phụ thuộc nhiều vào hệ thống bên ngoài**

Việc tích hợp API đòi hỏi đội ngũ backend phải xây dựng và duy trì nhiều lớp logic xử lý, kiểm soát ngữ cảnh, theo dõi trạng thái hội thoại và đảm bảo tính nhất quán. Điều này **tăng yêu cầu về năng lực kỹ thuật**, làm cho chi phí phát triển và duy trì lớn hơn so với phương án dựa trên nền tảng đã có sẵn nhiều tính năng hỗ trợ.

4. **Chi phí sử dụng API theo lưu lượng**

Nếu ChatBot hoạt động với lưu lượng người dùng cao, chi phí sử dụng API như Google Gemini sẽ phát sinh theo từng lượng truy vấn, không kiểm soát cứng bằng mô hình tự host. Điều này có thể dẫn đến chi phí vận hành lớn nếu không có kế hoạch tối ưu.

5. **Tính chính xác thông tin tuyển sinh không được đảm bảo tự động**

Vì mô hình tổng quát có dữ liệu huấn luyện rộng, không tập trung vào dữ liệu tuyển sinh của HPU, nên **hiệu quả đúng/sai trong các câu trả lời mang tính chuyên môn thấp hơn**, đặc biệt khi có thay đổi mới.

Về **ưu điểm**, sử dụng API như Google Gemini đôi khi đảm bảo được **mức độ bảo mật dữ liệu** khi xử lý bên phía nhà cung cấp. Tuy nhiên, trong bối cảnh tư vấn tuyển sinh — vốn là **thông tin công khai, không nhạy cảm**, ưu điểm này gần như không mang lại lợi ích thiết thực. Những hệ thống cần bảo mật cao thực sự là các dịch vụ liên quan đến tài chính, dữ liệu sinh viên cá nhân, nội bộ ngân sách... chứ không phải nội dung tuyển sinh.

Nhận xét về phương án sử dụng Botpress kết hợp Knowledge Base

Trái lại, lựa chọn xây dựng hệ thống dựa trên **Botpress** kết hợp với một **Knowledge Base (CSDL hiểu biết)** trực tiếp lấy dữ liệu từ trang thông tin chính thức của HPU có rất nhiều lợi thế phù hợp với bài toán thực tế:

1. Quản lý dữ liệu tập trung, dễ cập nhật theo quy trình nội bộ

Botpress cho phép liên kết chặt chẽ với Knowledge Base lấy rõ ràng nội dung từ **website hpu.edu.vn**, nên đội ngũ phụ trách chỉ cần cập nhật tại nguồn là ChatBot tự động phản ánh thông tin mới.

2. Kiểm soát luồng hội thoại rõ ràng, logic hơn

Với các module và luồng hội thoại tùy chỉnh, Botpress cho phép thiết kế **kịch bản trả lời đúng theo mục tiêu tuyển sinh**, hạn chế tối đa việc hội thoại bị lệch chủ đề hay tạo ra thông tin không liên quan.

3. Ít lỗi tạo thông tin sai hơn

Vì phần lớn câu trả lời được dẫn từ Knowledge Base đã kiểm chứng, hệ thống không phải “sáng tạo” nội dung mới mà chỉ chọn lọc và trả lời từ nội dung đã được chuẩn hóa, nên **tỷ lệ lỗi, nội dung sai lệch giảm đáng kể**.

4. Dễ mở rộng chức năng trong tương lai

Khi cần thêm các tính năng phụ như hỗ trợ đăng ký, tư vấn học phí, định hướng ngành học hay liên kết với CRM tuyển sinh, Botpress hỗ trợ modular để phát triển thêm mà không cần xây lại cấu trúc nền tảng.

5. Nguồn lực triển khai tiết kiệm hơn về lâu dài

Mặc dù Botpress cũng yêu cầu đội ngũ kỹ thuật, nhưng **cộng đồng hỗ trợ, tài liệu sẵn và tính tích hợp cao** giúp giảm khối lượng công việc so với tự xây backend kết nối API tổng quát.

Lựa chọn cuối cùng

Vì vậy, sau khi cân nhắc kỹ các yếu tố **tính chính xác, khả năng cập nhật, độ ổn định, chi phí và nguồn lực**, lựa chọn **tối ưu nhất cho hệ thống ChatBot AI tư vấn tuyển sinh của HPU là sử dụng nền tảng Botpress kết hợp với Knowledge Base lấy dữ liệu trực tiếp từ website nhà trường.**

Giải pháp này không chỉ vận hành hiệu quả trong thực tế hiện tại mà còn tạo nền tảng vững chắc để mở rộng thêm các tính năng tương tác tự động trong tương lai, phục vụ tốt hơn cho công tác tuyển sinh của nhà trường. Nếu cần, em có thể tiếp tục mở rộng thêm phương án triển khai chi tiết (bao gồm kiến trúc hệ thống, mô hình dữ liệu, phân tích chi phí và roadmap phát triển).

Kiến nghị

Từ kết quả nghiên cứu đạt được, em xin đề xuất một số kiến nghị như sau:

- Với nhà trường:
 - Nên triển khai ChatBot trong mùa tuyển sinh thực tế để giảm tải cho bộ phận tư vấn.
 - Cập nhật thường xuyên thông tin học phí, chỉ tiêu, phương thức tuyển sinh... để ChatBot phản hồi chính xác.
 - Kết nối ChatBot với các hệ thống khác như CRM, email hoặc các mạng xã hội để mở rộng kênh tiếp cận thí sinh.
- Với các nhóm nghiên cứu tiếp theo:
 - Có thể mở rộng ChatBot sang lĩnh vực học vụ, quản lý lớp, giải đáp học bổng, lịch thi, học lại...
 - Nghiên cứu tích hợp thêm giọng nói (voicebot) hoặc sử dụng mô hình mã nguồn mở để kiểm soát dữ liệu tốt hơn.
 - Thực hiện khảo sát người dùng để cải tiến trải nghiệm và tính chính xác của phản hồi.

Đề tài là một bước thử nghiệm hữu ích và khả thi trong việc ứng dụng AI vào giáo dục đại học, đặc biệt là trong công tác tuyển sinh – vốn đòi hỏi tính chính xác, tốc độ và sự thân thiện trong tương tác.

DANH MỤC TÀI LIỆU THAM KHẢO

American Psychological Association. (2023). *APA style references*.

<https://apastyle.apa.org/style-grammar-guidelines/references>

Dan, Y., Liu, Z., et al. (2023). *EduChat: An open toolkit for building ChatBots in education*. arXiv. <https://arxiv.org>

Đại học Quốc gia Hà Nội. (2021). *Ứng dụng mô hình ngôn ngữ lớn (LLM) trong giáo dục đại học*. <https://vnu.edu.vn/llm-trong-giao-duc>

Google AI. (2024). *Gemini API documentation*. <https://ai.google.dev/gemini-api/docs>

Google DeepMind. (2024). *Gemini 1.5 technical report*.
<https://deepmind.google>

HPU – Trường Đại học Quản lý và Công nghệ Hải Phòng. (2024). *Thông tin tuyển sinh, học phí và ngành đào tạo*. <https://hpu.edu.vn/tuyen-sinh>

Nguyễn, L. B. (2020). *Phân tích dữ liệu khảo sát và thiết kế ChatBot học thuật* (Luận văn thạc sĩ). Trường Đại học Công nghệ TP. Hồ Chí Minh.

Nguyễn, V. H. (2023). *Trí tuệ nhân tạo và ứng dụng trong giáo dục*. NXB Đại học Quốc gia TP. Hồ Chí Minh.

OpenAI. (2023). *Introduction to large language models (LLMs)*.
<https://openai.com/research/large-language-models>

Python Software Foundation. (2024). *Flask documentation*.
<https://flask.palletsprojects.com>

Supabase. (n.d.). *Supabase documentation*. <https://supabase.com/docs>

Tạp chí Giáo dục. (n.d.). *Ứng dụng ChatBot AI trong dạy học cá nhân hóa môn Toán học*. <https://tapchigiaoduc.edu.vn/article/89662/222/ung-dung-ChatBot-ai-trong-day-hoc-ca-nhan-hoa-mon-toan-hoc/>

Trần, T. H., & Lê, Đ. M. (2022). Xây dựng hệ thống ChatBot tư vấn thông minh bằng AI. *Tạp chí Khoa học & Công nghệ – Đại học Bách Khoa Hà Nội*.

Trương, H. A. (n.d.). *ChatGPT và giáo dục: Tổng quan tình hình nghiên cứu trên thế giới và Việt Nam*.